TÉCNICO
LISBOA

CDRGen: A Clinical Data Registry Generator

Pedro Miguel da Cunha Alves

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisors: Prof. João Carlos Serrenho Dias Pereira
Prof. Helena Isabel de Jesus Galhardas

Examination Committee

Chairperson: Prof. Alberto Manuel Rodrigues da Silva
Supervisor: Prof. João Carlos Serrenho Dias Pereira
Member of the Committee: Prof. Alexandre Paulo Lourenço Francisco

November 2019

Acknowledgments

First, I would like to thank all of family and especially my parents, José and Rosa, for their unconditional support throughout my school progress.

Second, I thank to my supervisors, Helena, João and Manuel, for the constant support and weekly motivation that they kindly provided during the development of this thesis.

Third, I thank to my friends, in particular, to Ana, Catarina, Diogo, Jorge and Pedro, for their friendship and strength.

Finally, I thank to my university, Instituto Superior Técnico, for providing me with an excellent education and for all the knowledge it has given me.

Abstract

In the health sector, data analysis is typically performed over clinical data to conduct scientific researches that address clinical problems. Clinical data is stored in clinical data registries, which currently tend to be electronic and supported by a set of Excel files or by a software application composed of a user interface, a database and the logic that provides the communication between these two. As far as the author could find, in Portugal there are twelve clinical data registries covering seven medical specialties, which is not enough to satisfy the demand of physicians from other medical specialties. This demand eventually causes the creation of new electronic clinical data registries, but these are usually created from scratch. This results in a waste of resources since clinical data registries have common characteristics: their user interfaces support common functionalities and the data that is collected covers common topics.

In order to satisfy the physicians' demands while avoiding wasting resources when creating new clinical data registries, this thesis focus on the design and development of a Clinical Data Registry Generator (CDRGen) software system. CDRGen is able to generate a clinical data registry for any medical specialty with a minimum effort in terms of software design and development. With CDRGen, the creation of a new clinical data registry becomes more efficient, which enables to satisfy the physicians' requirements in a timely manner. In addition, CDRGen facilitates the developers' work. In simple terms, the developer only needs to specify the data required to be collected and CDRGen generates a clinical data registry based on it. To develop CDRGen, we start by performing an analysis of the existing Portuguese clinical data registries regarding three topics: (i) collected data; (ii) functionalities; and (iii) user interface. Based on the information collected during this analysis, we were able to know what a clinical data registry needs to support.

To develop CDRGen, we use one existing clinical data registry. After its development, we use two other clinical data registries: one to test and another to validate. For each of these two clinical data registries, we measure the time required to develop a new clinical data registry. In addition, we assess the quality of the generated clinical data registry in terms of its database structure and its functionalities.

Keywords

CDRGen, clinical data registry, database, user interface

Resumo

No setor da saúde, a análise de dados é tipicamente realizada sobre dados clínicos por forma a levar a cabo pesquisas científicas que abordam problemas clínicos. Os dados clínicos são armazenados em registos de dados clínicos. Estes atualmente tendem a ser eletrónicos e são suportados por um conjunto de ficheiros Excel ou por uma aplicação de software composta por uma base de dados, uma interface de utilizador e pela lógica que fornece a comunicação entre estes dois. Tanto quanto o autor foi capaz de encontrar, em Portugal existem doze registos de dados clínicos que abrangem sete especialidades médicas, o que não é suficiente para satisfazer a demanda dos médicos de outras especialidades médicas. Eventualmente, esta demanda acaba por causar a criação de novos registos de dados clínicos eletrónicos, mas estes normalmente são criados de raiz. Isto resulta num desperdício de recursos, uma vez que os registos de dados clínicos têm características em comum: as suas interfaces de utilizador suportam funcionalidades comuns e os dados que são recolhidos abrangem tópicos comuns.

Por forma a satisfazer a demanda dos médicos enquanto se evita o desperdício de recursos a criar novos registos de dados clínicos, esta tese foca-se no desenho e desenvolvimento de um Gerador de Registos de Dados Clínicos (CDRGen). O CDRGen é capaz de gerar um registo de dados clínicos para uma qualquer especialidade médica com um esforço mínimo em termos de desenho e desenvolvimento de software. Com o CDRGen, a criação de um registo de dados clínicos torna-se mais eficiente, o que permite satisfazer os requisitos dos médicos de forma atempada. Além disso, o CDRGen facilita o trabalho dos desenvolvedores. Em termos simples, o desenvolvedor apenas precisa de especificar os dados necessários a serem recolhidos e o CDRGen gera um registo de dados clínicos baseado nisso. Para desenvolver o CDRGen, nós começamos por realizar uma análise aos registos de dados clínicos que existem em Portugal em relação a três tópicos: (i) dados recolhidos; (ii) funcionalidades; e (iii) interface de utilizador. Com base nas informações que recolhemos durante esta análise, nós fomos capazes de saber o que um registo de dados clínicos precisa de suportar.

Para desenvolver o CDRGen, nós usamos um registo de dados clínicos que já existe. Após o seu desenvolvimento, nós usamos outros dois registos de dados clínicos: um para testar e outro para validar. Para cada um destes dois registos de dados clínicos, nós medimos o tempo necessário para desenvolver um novo registo de dados clínicos. Além disso, nós avaliámos a qualidade do registo de dados clínicos gerado em termos da estrutura da sua base de dados e das suas funcionalidades.

Palavras Chave

CDRGen, registo de dados clínicos, base de dados, interface de utilizador

Contents

1 Introduction **1**

1.1 Problem 4

1.2 Objectives 5

1.3 Solution 6

1.4 Validation 6

1.5 Document Outline 7

2 Related Work **9**

2.1 Specialty-specific Clinical Data Registries 11

2.1.1 Reuma.pt 12

2.1.2 Derma.pt 18

2.1.3 National Cancer Registry (RON) 19

2.1.4 SI.VIDA 22

2.1.5 Discussion 23

2.2 Cardiological Clinical Data Registries 24

2.2.1 Portuguese Registry of Acute Coronary Syndromes (ProACS) 25

2.2.2 National Registry of Arrhythmogenic Right Ventricular Myocardiopathy (RNMAVD) 26

2.2.3 Portuguese Registry of Hypertrophic Cardiomyopathy (PRo-HCM) 27

2.2.4 Portuguese Myocarditis Registry (PMR) 28

2.2.5 Portuguese Registry of Non-compaction Cardiomyopathy (PRNC) 28

2.2.6 Portuguese Registry on Interventional Cardiology (RNCI) 29

2.2.7 Discussion 30

2.3 Research Clinical Data Registries 31

2.3.1 Umedicine 32

2.3.2 PRECISE Stroke 36

2.3.3 Discussion 38

2.4 Low-Code Development Platforms (LCDPs) 39

2.4.1 OutSystems 40

2.4.2 Discussion . . . 42

3 Requirements Analysis 43

3.1 Data to Store . . . 46

3.2 Functionalities . . . 52

3.3 User Interface . . . 57

3.4 Discussion . . . 58

4 CDRGen: A Clinical Data Registry Generator 63

4.1 CDRGen Architecture . . . 65

4.2 CDRGen Input 1/2: Specification . . . 66

4.2.1 Entity Groups . . . 70

4.2.2 Entities . . . 72

4.2.3 Attributes . . . 73

4.3 CDRGen Input 2/2: User Interface Labels . . . 76

4.4 CDRGen Output: Clinical Data Registry . . . 77

4.5 CDRGen: Parsing and Code Generation . . . 78

4.5.1 Parser . . . 78

4.5.2 Database Generator . . . 79

4.5.3 Web User Interface Client Generator . . . 82

4.5.3.1 Data Entry Forms . . . 82

4.5.3.2 Data Navigation Screens . . . 85

4.5.4 Web Application Generator . . . 90

4.6 Summary . . . 94

5 Validation 97

5.1 Validation Methodology . . . 99

5.2 Metrics . . . 99

5.3 Testing with PRNC . . . 100

5.4 Validation with PRECISE Stroke . . . 104

5.5 Discussion . . . 110

6 Conclusions and Future Work 115

6.1 Conclusions . . . 117

6.2 Limitations and Future Work . . . 117

A Example of a Case Report Form (CRF) 127

B PRNC - JSON file specification 129

C PRECISE Stroke - JSON file specification 144

List of Figures

1.1 Overview of the solution to generate a clinical data registry. 6

2.1 First screen displayed inside the physicians' area. 13
2.2 Visit's screen example in Reuma.pt for RA. 14
2.3 Example of a screen with an excerpt of a questionnaire in Reuma.pt's patient area. Figure from www.reuma.pt/docs/NL201604.html. 15
2.4 The screen where we defined the filters to select the patients who initiated the biological therapy in 2012 and whose first biological was an *anti-TNF*. 16
2.5 The screen that contains the list of the patients who initiated the biological therapy in 2012 and whose first biological was an *anti-TNF*. 16
2.6 Patient's information screen shown to a physician in Umedicine. 34
2.7 Form to edit part of a patient's personal data in Umedicine. Figure from Umedicine's user interface. 34
2.8 Visualization resulting from the application of the biclustering algorithm. Figure from Umedicine's user interface. 35
2.9 Example of the common information displayed regarding a cluster of patients when clicking on a chart. Figure from Umedicine's user interface. 36
2.10 An example of a screen of the PRECISE Stroke test version's user interface. This screen allows the user to visualize and edit the patient's personal characteristics. Figure from PRECISE Stroke test version's user interface. 37
2.11 Screen of OutSystems platform that allows the user to create the data model of an application. 41
2.12 Screen of OutSystems platform where the user can define the application logic. 41
2.13 Screen of OutSystems platform that allows the user to create and customize user interface screens of an application. 42

3.1 UML class diagram of the entity group, entity and attribute concepts. 46

3.2 Relationships between entities in PRNC. 47

3.3 Relationships between entities in RNMAVD. 49

3.4 Relationships between entities in Umedicine. 51

3.5 Relationships between entities in RON. 52

3.6 Umedicine's use case diagram. 53

3.7 Reuma.pt's use case diagram. 54

3.8 RON's use case diagram. 56

3.9 RNMAVD and PRNC's use case diagram. 58

3.10 Use case diagram of the clinical data registry to be generated. 60

4.1 High-level architecture of CDRGen. 65

4.2 UML class diagram of the specification metamodel. 67

4.3 Metadata of the clinical data registry to be generated according to Example (1). 68

4.4 The resulting JSON file specification according to Example (1). 69

4.5 The root object of the JavaScript Object Notation (JSON) file specification. 69

4.6 The value of the `users` key to generate a clinical data registry that supports the `administrator physician` and `non-administrator physician` user types. 70

4.7 The value of the `entity groups` key that includes all the possible keys that are allowed to be specified. 70

4.8 The value of the `entity groups` key to generate a clinical data registry that has the `person` and the `treatment` entity groups, the `patient` user type has read and write permissions in the `person` entity group, but has only read permissions in the `treatment` entity group. 72

4.9 Example of how to specify the historic property to an entity in which the historic date-type attribute's name is "Data". 72

4.10 Specification of the `treatment` and the `complications` attributes according to the example. 75

4.11 Specification of the object that represents an enumerated-type attribute whose name is "gender", that may have "male" and "female" as values, and not null as a property. 76

4.12 Excerpt of a JSON file *user interface labels* that is customized for the Portuguese idiom. . 76

4.13 Architecture of the generated clinical data registry. Figure partially edited from Lages *et al.* [9]. 77

4.14 UML class diagram of the set of classes required to store the data extracted from the specification. 79

4.15 The relational schema of the database that would be generated by CDRGen for an entity (denominated `patient`) with one list-type attribute (denominated `comorbidities`). 80

4.16 The relational schema of the database that would be generated by CDRGen according to the given JSON file specification for Example (1). 81
4.17 The relational schema of the `user` and the `userRole` tables. 81
4.18 Form for editing an instance of the `Patient` entity in Simple Registry. 83
4.19 UML class diagram of the set of classes used to generate the data entry forms. 84
4.20 Screen that displays an overview of all data of a patient in Simple Registry. 85
4.21 Screen that displays an overview of all data of a patient if, in Simple Registry, we had also specified the `First Exam` and `Second Exam` entities in the `medical examination` entity group. 86
4.22 UML class diagram of the set of classes used to generate the screen that displays an overview of all data of a patient. 87
4.23 Screen that displays all data of the `Patient` entity in Simple Registry. 88
4.24 Screen that displays all data of the `Medical Information` historic-type entity that has two instances for the concerned patient. 88
4.25 UML class diagram of the set of classes used to generate the screen that displays all data of an entity. 89
4.26 UML class diagram of the set of classes used to generate the screen that displays one instance (the unique or the most recent chronologically) of an entity. 90
4.27 The web server component architecture. Figure from Lages *et al.* [9]. 91
4.28 UML class diagram of the set of classes required to generate the services, the data acess objects and the Java classes that represent the entities. 92
4.29 UML class diagram of the set of classes required to generate the controllers. 93

5.1 Excerpt of the JSON file specification for PRNC clinical data registry specifying two entities of the `person` entity group. 101
5.2 Screen that displays an overview of all data of a patient in the generated clinical data registry based on PRNC's data. 104
5.3 Form to edit the instance of the entity that represents the patient (labelled as "Identificação") in the generated clinical data registry based on PRNC's data. 104
5.4 Screen that displays all data of the `person` entity group, in this case regarding the entity that represents the patient (labelled as "Identificação") in the generated clinical data registry based on PRNC's data. 105
5.5 Excerpt of the JSON file specification for PRECISE Stroke clinical data registry specifying one entity of the `person` entity group (the entity that represents the patient). 106
5.6 Screen that displays an overview of all data of a patient in the generated clinical data registry based on PRECISE Stroke's data. 110

5.7 Form to edit the instance of the entity that represents the patient (labelled as “Geral”) in the generated clinical data registry based on PRECISE Stroke’s data. 111
5.8 Form to edit the instance of the entity that represents the patient (labelled as “Geral”) in the existing PRECISE Stroke test version. 111
5.9 Screen that displays all data of the `person` entity group, in this case regarding the entity that represents the patient (labelled as “Geral”) in the generated clinical data registry based on PRECISE Stroke’s data. 112

A.1 Example of a CRF. Figure from https://docs.openclinica.com/sites/fileuploads/akaza/cms-docs/3.1/CRFWebView.png. 128

B.1 PRNC - JSON file specification 1/14. 130
B.2 PRNC - JSON file specification 2/14. 131
B.3 PRNC - JSON file specification 3/14. 132
B.4 PRNC - JSON file specification 4/14. 133
B.5 PRNC - JSON file specification 5/14. 134
B.6 PRNC - JSON file specification 6/14. 135
B.7 PRNC - JSON file specification 7/14. 136
B.8 PRNC - JSON file specification 8/14. 137
B.9 PRNC - JSON file specification 9/14. 138
B.10 PRNC - JSON file specification 10/14. 139
B.11 PRNC - JSON file specification 11/14. 140
B.12 PRNC - JSON file specification 12/14. 141
B.13 PRNC - JSON file specification 13/14. 142
B.14 PRNC - JSON file specification 14/14. 143

C.1 PRECISE Stroke - JSON file specification 1/57. 145
C.2 PRECISE Stroke - JSON file specification 2/57. 146
C.3 PRECISE Stroke - JSON file specification 3/57. 147
C.4 PRECISE Stroke - JSON file specification 4/57. 148
C.5 PRECISE Stroke - JSON file specification 5/57. 149
C.6 PRECISE Stroke - JSON file specification 6/57. 150
C.7 PRECISE Stroke - JSON file specification 7/57. 151
C.8 PRECISE Stroke - JSON file specification 8/57. 152
C.9 PRECISE Stroke - JSON file specification 9/57. 153
C.10 PRECISE Stroke - JSON file specification 10/57. 154
C.11 PRECISE Stroke - JSON file specification 11/57. 155

C.12 PRECISE Stroke - JSON file specification 12/57. 156
C.13 PRECISE Stroke - JSON file specification 13/57. 157
C.14 PRECISE Stroke - JSON file specification 14/57. 158
C.15 PRECISE Stroke - JSON file specification 15/57. 159
C.16 PRECISE Stroke - JSON file specification 16/57. 160
C.17 PRECISE Stroke - JSON file specification 17/57. 161
C.18 PRECISE Stroke - JSON file specification 18/57. 162
C.19 PRECISE Stroke - JSON file specification 19/57. 163
C.20 PRECISE Stroke - JSON file specification 20/57. 164
C.21 PRECISE Stroke - JSON file specification 21/57. 165
C.22 PRECISE Stroke - JSON file specification 22/57. 166
C.23 PRECISE Stroke - JSON file specification 23/57. 167
C.24 PRECISE Stroke - JSON file specification 24/57. 168
C.25 PRECISE Stroke - JSON file specification 25/57. 169
C.26 PRECISE Stroke - JSON file specification 26/57. 170
C.27 PRECISE Stroke - JSON file specification 27/57. 171
C.28 PRECISE Stroke - JSON file specification 28/57. 172
C.29 PRECISE Stroke - JSON file specification 29/57. 173
C.30 PRECISE Stroke - JSON file specification 30/57. 174
C.31 PRECISE Stroke - JSON file specification 31/57. 175
C.32 PRECISE Stroke - JSON file specification 32/57. 176
C.33 PRECISE Stroke - JSON file specification 33/57. 177
C.34 PRECISE Stroke - JSON file specification 34/57. 178
C.35 PRECISE Stroke - JSON file specification 35/57. 179
C.36 PRECISE Stroke - JSON file specification 36/57. 180
C.37 PRECISE Stroke - JSON file specification 37/57. 181
C.38 PRECISE Stroke - JSON file specification 38/57. 182
C.39 PRECISE Stroke - JSON file specification 39/57. 183
C.40 PRECISE Stroke - JSON file specification 40/57. 184
C.41 PRECISE Stroke - JSON file specification 41/57. 185
C.42 PRECISE Stroke - JSON file specification 42/57. 186
C.43 PRECISE Stroke - JSON file specification 43/57. 187
C.44 PRECISE Stroke - JSON file specification 44/57. 188
C.45 PRECISE Stroke - JSON file specification 45/57. 189
C.46 PRECISE Stroke - JSON file specification 46/57. 190

C.47 PRECISE Stroke - JSON file specification 47/57. 191
C.48 PRECISE Stroke - JSON file specification 48/57. 192
C.49 PRECISE Stroke - JSON file specification 49/57. 193
C.50 PRECISE Stroke - JSON file specification 50/57. 194
C.51 PRECISE Stroke - JSON file specification 51/57. 195
C.52 PRECISE Stroke - JSON file specification 52/57. 196
C.53 PRECISE Stroke - JSON file specification 53/57. 197
C.54 PRECISE Stroke - JSON file specification 54/57. 198
C.55 PRECISE Stroke - JSON file specification 55/57. 199
C.56 PRECISE Stroke - JSON file specification 56/57. 200
C.57 PRECISE Stroke - JSON file specification 57/57. 201

List of Tables

2.1 Summary of the specialty-specific clinical data registries 24
2.2 Summary of the cardiological clinical data registries 31
2.3 Summary of the research clinical data registries 38

3.1 List of entities with respect to the data collected by PRNC and the corresponding entity groups. 47
3.2 List of entities with respect to the data collected by RNMAVD and the corresponding entity groups. 49
3.3 List of entities with respect to the data collected by Umedicine and the corresponding entity groups. 50
3.4 List of entities with respect to the data collected by RON and the corresponding entity groups. 51
3.5 Existing widgets and the corresponding attribute types available in Reuma.pt and Umedicine user interfaces. 58
3.6 Widget that will be used to enter data according to each type of attribute. 62

5.1 Time spent writing the specification for PRNC. 100
5.2 The ratio of the tables, their attributes and relationships that the generated clinical data registry's database has in relation to what it is expected to have. 102
5.3 Time spent writing the specification for PRECISE Stroke. 105
5.4 The ratio of the tables, their attributes and relationships that the generated clinical data registry's database has in relation to what it is expected to have. 107

Acronyms

ACS	Acute Coronary Syndromes
ACSS	Central Administration of the Health System
Ajax	Asynchronous JavaScript and XML
APIC	Portuguese Association of Interventional Cardiology
ARVM	Arrhythmogenic Right Ventricular Myocardiopathy
BPH	Benign Prostatic Hyperplasia
CARDS	European Data Standards for Clinical Cardiology Practice
CDRGen	Clinical Data Registry Generator
CNCDC	National Center for Data Collection in Cardiology (*Centro Nacional de Coleção de Dados em Cardiologia*, in Portuguese)
CRF	Case Report Form
DMARD	Disease Modifying Anti-rheumatic Drugs
GEDMP	Working Group on Myocardial and Pericardial Diseases (*Grupo de Estudo de Doenças do Miocárdio e do Pericárdio*, in Portuguese)
GHIPOFG	Portuguese Institute Oncology Francisco Gentil Hospital Group (*Grupo Hospitalar Instituto Português de Oncologia Francisco Gentil*, in Portuguese)
HCM	Hypertrophic Cardiomyopathy
HIV	Human Immunodeficiency Virus
IC	Interventional Cardiology
INSA	National Health Institute Doutor Ricardo Jorge (*Instituto Nacional de Saúde Doutor Ricardo Jorge*, in Portuguese)

IPSS	International Prostate Symptom Score
JIA	Juvenile Idiopathic Arthritis
JSON	JavaScript Object Notation
JSP	Java Server Pages
LCDP	Low-Code Development Platform
NC	Non-compaction Cardiomyopathy
PMR	Portuguese Myocarditis Registry
PRNC	Portuguese Registry of Non-compaction Cardiomyopathy
PRO	Patient Reported Outcomes
ProACS	Portuguese Registry of Acute Coronary Syndromes
PRo-HCM	Portuguese Registry of Hypertrophic Cardiomyopathy
PsA	Psoriatic Arthritis
RA	Rheumatoid Arthritis
RIS	Health Informatics Network (*Rede de Informação da Saúde*, in Portuguese)
RNCI	Portuguese Registry on Interventional Cardiology (*Registo Nacional de Cardiologia de Intervenção*, in Portuguese)
RNMAVD	National Registry of Arrhythmogenic Right Ventricular Myocardiopathy (*Registo Nacional de Miocardiopatia Arritmogénica do Ventrículo Direito*, in Portuguese)
RON	National Cancer Registry (*Registo Oncológico Nacional*, in Portuguese)
SLE	Systemic Lupus Erythematosus
SNS	National Health Service (*Serviço Nacional de Saúde*, in Portuguese)
SPC	Portuguese Society of Cardiology (*Sociedade Portuguesa de Cardiologia*, in Portuguese)
SPDV	Portuguese Society of Dermatology and Venereology (*Sociedade Portuguesa de Dermatologia e Venereologia*, in Portuguese)
SPMS	Shared Services of the Ministry of Health (*Serviços Partilhados do Ministério da Saúde*, in Portuguese)

SPR	Portuguese Society of Rheumatology (*Sociedade Portuguesa de Reumatologia*, in Portuguese)
SQL	Structured Query Language

1

Introduction

Contents

1.1 Problem 4

1.2 Objectives 5

1.3 Solution 6

1.4 Validation 6

1.5 Document Outline 7

With big data the need for data analysis has been increasing as it brings new knowledge that can help people making decisions. The typical data analysis process encompasses several steps. First, we have to know what data we want to acquire. Second, we need to define a way to collect data, which is usually handled by a software application. Third, we need a place to store the collected data. Although a database can store data, it does not have, by itself, the proper tools for an ordinary person to enter data that will be collected. Thus, a software application is required. It provides a user interface to enable entering data and then accessing it. By having an application and a database, we have a way of entering data and a place to store it, that is a *data registry*. Fourth, as the final step, we can use the registry's data to meet the ultimate goal: data analysis. With data analysis we can discover useful information using, for example, clustering algorithms that can be computed over the data stored in the database. Clustering algorithms split data into groups that have similar properties, which can allow us to discover, for example, what are the most common properties.

In the health sector, the concept of *clinical data registry* is crucial [15]. A clinical data registry records data about patients and the health care they receive over time. Typically, a clinical data registry is focused on a given medical specialty or a set of diseases. It is assumed that clinical data registries are of paramount importance for the execution of a clinical practice with quality [11]. In addition, they prove to be remarkable for monitoring the quality of care and for conducting scientific research that addresses clinical problems [3]. This scientific research is based on data analysis performed over data collected and stored in clinical data registries. Clinical data registries can also support clinical trials, i.e., scientific studies that are conducted in order to find better ways to treat diseases or to identify which treatments work best for certain diseases or groups of people.

In the past, clinical data was registered in paper [3,15,18], which could lead, for example, to potential reading errors due to the need to perceive the physicians' handwriting. In addition, searching data registered on paper was very time-consuming. With clinical data registries in paper, clinical trials, for example, took longer to be performed, since a lot of time was spent searching for patients who were suited for a clinical trial. Current clinical data registries increasingly tend to be electronic. Unlike paper-based, electronic clinical data registries do not require large amounts of time to search over the data. They also enable to carry out analyses on a large number of patients and to obtain information, for example, regarding rare clinical events, that formerly was not captured [15–17].

Very often, physicians use Excel files to build their own electronic clinical data registry. However, Excel files scale poorly. For example, to perform data analysis on data from multiple Excel files built by several physicians of the same medical specialty, it is required to integrate them taking into account the semantic and syntax differences that often exist in different files created by different people. Given the limitations of Excel files, over the years there has been an effort to build more robust electronic clinical data registries. In particular, these clinical data registries are composed of three components: (i) a user interface that enables to enter data through forms and then to access data; (ii) a database to store the entered data; and (iii) the logic that provides the communication between the user interface and the database. In some cases, these clinical data registries also have a data analysis component that enables to explore groups of similar patients, for instance. From now on, whenever we mention clinical data registries, we will be referring to electronic clinical data registries.

In Portugal, there have been some initiatives regarding the creation of clinical data registries. However, the clinical data registries that currently exist do not cover all medical specialties. In fact, we were able to find 12 clinical data registries (such as Reuma.pt[1], Derma.pt[2], PRNC[3] and PRECISE Stroke[4]) that cover 7 medical specialties (all described in Chapter 2). All the existing Portuguese clinical data registries have common purposes (i.e., monitoring treatments and diseases). They also collect data and types of data that are common to various medical specialties (for instance, the patient's name as a string and the birth date as a date). In addition, as far as we know, the data that is collected and stored by half of the Portuguese clinical data registries is structured (e.g., it is stored in a relational database). This characteristic provides advantages, since storing structured data facilitates the execution of data analysis. In the majority of Portuguese clinical data registries, data analysis is mainly performed externally to the clinical data registries. Reuma.pt is one of the few examples that supports the execution of some statistical analysis and produces statistical reports.

1.1 Problem

Although the electronic clinical data registries have advantages over the paper version, they still have some shortenings.

First, as already mentioned, the existing electronic clinical data registries in Portugal do not cover all medical specialties. This, by itself, poses a problem, given the demand from physicians to have a clinical data registry for their medical specialty. When physicians want a clinical data registry for their medical specialty or for a given pathology/disease and it does not exist, one of the following two situations occurs: (i) the physician builds his/her own Excel file, either by copying data from the paper-based clinical data registries that already exist or by directly entering data in an Excel file; or (ii) a new software application is built in order to support a clinical data registry. As mentioned above, Excel files have limitations, so situation (i) is not the most appropriate.

Second, if a new software application system is built to support a clinical data registry, it is typically constructed from scratch. This turns to be a waste of resources, since a lot of time is lost on the design and implementation of a clinical data registry that has similar characteristics to other clinical

[1] http://www.reuma.pt/
[2] http://www.derma.pt/
[3] http://registos.spc.pt/RegistoMNC/
[4] https://stroke.precisemed.org/home/

data registries that already exist. Namely, the design of clinical data registries is usually composed of data entry forms and several user interface screens where the user can access and navigate through the data. In addition, the data collected by clinical data registries covers common topics. Specifically, clinical data registries often collect data regarding the patient's characteristics, medical examinations, diagnosis, among others. We are, therefore, systematically repeating work (or very similar work) and wasting resources. The repeated work increases substantially if the way in which the user interacts with the clinical data registries' user interfaces is always the same. Derma.pt (described in Section 2.1.2) is an exceptional example that, we believe, has been built based on Reuma.pt (described in Section 2.1.1). Therefore, some of the results of Reuma.pt may have already been reused. We could use LCDPs to build applications that support clinical data registries and also reuse the work performed between them. However, we would always have to make a lot of changes if we wanted to generate another clinical data registry that collected different specific data and that provided the same standard user interface. Namely, we would not only have to change the database that stores the clinical data registry's data (as expected), but also its user interface to accommodate the different data.

1.2 Objectives

The main objective of this thesis is to design and develop a software system that is able to generate a clinical data registry for any medical specialty based on a high-level specification given as input. Using this software system, we want to generate clinical data registries with a minimum effort in terms of software design and development. This software system is called *Clinical Data Registry Generator* (CDRGen).

To develop CDRGen, we focus on the following three sub-objectives:

- Identification of the collected data, the functionalities and the user interface characteristics taking into account the existing Portuguese clinical data registries. This is required to identify what a typical clinical data registry needs to support.

- Syntax of the specification to provide as input to CDRGen in order to generate a new clinical data registry.

- Generation of the software logic required to create and manage the database, the user interface and the logic that provides the communication between these two components of the generated clinical data registry.

1.3 Solution

The overview of the solution to generate a clinical data registry is illustrated in Figure 1.1. The

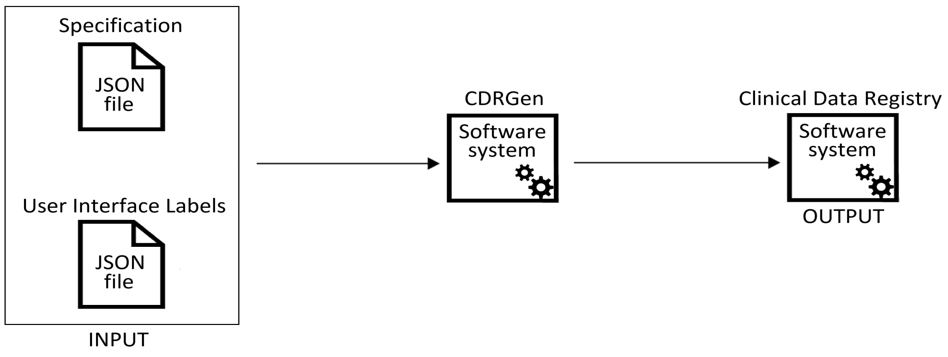


Figure 1.1: Overview of the solution to generate a clinical data registry.

process for generating a clinical data registry is described as follows:

- The input of *CDRGen* is two JSON files: the JSON file *specification* and the JSON file *user interface labels*. The specification is mainly composed of the data that must be collected by the clinical data registry (e.g., patient's name) and the corresponding data types (e.g., string, integer). The user interface labels describe the labels of the user interface that are not related with the data to be collected by the clinical data registry (e.g., labels of buttons).

- *CDRGen* reads these two JSON files, parses them, and generates the code to create and manage the generated clinical data registry.

- A *clinical data registry* is the output of *CDRGen*. In general, this clinical data registry is composed of three components: a database, a user interface and the logic that provides the communication between the database and the user interface.

The development of CDRGen takes into account the requirements analysis that we perform based on what is currently supported by existing Portuguese clinical data registries. In addition, we use one of these clinical data registries to test CDRGen during its development. With this approach, we seek to make CDRGen generic enough to generate a clinical data registry for any medical specialty.

1.4 Validation

After the development of the CDRGen software system, we use two other existing clinical data registries that were not used during its development. In concrete, we test CDRGen against PRNC for

Cardiology (described in Section 2.2.5), in order to verify if our requirements analysis has been well performed. In addition, we validate CDRGen against PRECISE Stroke for Neurology (described in Section 2.3.2).

For each clinical data registry, we measure:

- The *time* required to develop a clinical data registry with CDRGen, which includes the time to write the specification and to generate the clinical data registry.
- The *ratio* of tables, their relationships and attributes that exist in the generated database in relation to the expected database based on the data collected by the existing clinical data registry.
- The *ratio* of functionalities that the generated clinical data registry can perform in relation to the functionalities provided by the existing clinical data registry.

1.5 Document Outline

This thesis is organized as follows.

In Chapter 2, we detail the related work with respect to the clinical data registries that exist in Portugal and with respect to LCDPs in general. In particular, we describe the twelve Portuguese clinical data registries that we were able to find and the LCDPs.

In Chapter 3, we perform a requirements analysis based on what is currently supported by the Portuguese clinical data registries that we describe in Chapter 2. This analysis aims at deciding what the clinical data registry to be generated will have to support regarding three topics: data to store, functionalities and user interface.

In Chapter 4, we present CDRGen. In concrete, we describe the CDRGen's input, output and the software system modules that process the input in order to generate the output.

In Chapter 5, we validate CDRGen against two existing clinical data registries. In concrete, for each one, we measure the time required to develop the clinical data registry using CDRGen and we evaluate the quality of the generated clinical data registry in terms of its database structure and its functionalities.

Finally, Chapter 6 presents the conclusions of this thesis, an approach to its limitations and a vision of future work.

2

Related Work

Contents

2.1 Specialty-specific Clinical Data Registries 11
2.2 Cardiological Clinical Data Registries 24
2.3 Research Clinical Data Registries 31
2.4 Low-Code Development Platforms (LCDPs) 39

In this chapter, we describe twelve electronic clinical data registries that exist in Portugal and Low-Code Development Platforms (LCDPs). As far as we know, there are no more than twelve clinical data registries in Portugal. Some of the additional clinical data registries that we believe to exist do not have publicly available information describing them, as it happens in the case of some cardiological clinical data registries.

This chapter is organized in four sections. In Section 2.1, we describe 4 clinical data registries, each covering a different medical specialty. In Section 2.2, we present 6 clinical data registries that exclusively address different diseases of the Cardiology specialty. In Section 2.3, we describe 2 clinical data registries built in the context of research projects regarding two other medical specialties: Urology and Neurology. Finally, in Section 2.4, we address the LCDPs in general and we detail one of them: OutSystems.

2.1 Specialty-specific Clinical Data Registries

In Portugal there are some electronic clinical data registries that individually cover one (or part) of a medical specialty. These data registries enable to store data regarding patients and the corresponding clinical data related to the medical specialty in question. The main purpose of clinical data registries is to monitor patients and the clinical aspects around them. The ultimate goal of these clinical data registries is to support data analysis aiming at improving the care provided.

In this section, we address four Portuguese electronic clinical data registries, namely:

- Reuma.pt (Section 2.1.1)
- Derma.pt (Section 2.1.2)
- RON (Section 2.1.3)
- SI.VIDA (Section 2.1.4).

For each of these clinical data registries, we mention the medical specialty to which each one refers and its purposes. We also present some considerations concerning its history and/or current state. Particularly in relation to Reuma.pt, we refer some aspects regarding the user interface offered, describing in detail the most relevant screens and the corresponding functionalities. In addition, regarding Reuma.pt and RON, we list the data that each one collects or intends to collect. Finally, we also refer examples of data analysis that are or should be provided by Reuma.pt, RON and SI.VIDA.

2.1.1 Reuma.pt

The Rheumatic Diseases Portuguese Registry, Reuma.pt[1], is an online software platform that contains a database of patients with rheumatic diseases and their clinical records. One of the key aspects of Reuma.pt is that it is designed for real time utilization during patient observation. Another key aspect is that Reuma.pt enables the link with the electronic health record systems of, for instance, a hospital [3]. This connection avoids double-typing and record duplications since a replica of the data inserted through Reuma.pt can be added to the patient's clinical process that belongs to a given hospital. The development of Reuma.pt was conducted by the Portuguese Society of Rheumatology (SPR) and was launched in June 2008. Currently, Reuma.pt contains more than 19.000 patients and 158.000 visits registered among more than 80 clinical centers[2]. These centers are spread mainly in Portugal, and a few in two other countries: Brazil and United Kingdom. Reuma.pt covers about 13 diseases, in which Rheumatoid Arthritis (RA), Spondyloarthritis, Psoriatic Arthritis (PsA), Juvenile Idiopathic Arthritis (JIA) and Systemic Lupus Erythematosus (SLE) are the main ones. Initially, the goal of Reuma.pt was to register all the patients in Portugal with rheumatic diseases that are treated with biological treatments, where biotechnological drugs are administrated. This was intended to monitor treatment efficacy, safety and long-term comorbidities [3]. However, its initial goal was extended to include treatments with synthetic Disease Modifying Anti-rheumatic Drugs (DMARD) and other therapeutic strategies, and also to determine the associated outcomes of all treatments [15].

Reuma.pt is built on Microsoft Silverlight[3], a software platform that enables to develop web-based applications. Although all the data that will be shown through the user interface is in Portuguese, Reuma.pt is also available in English since 2014. Reuma.pt is accessed through a web browser and provides two different areas: the patient's area and the physician's area. Each area has its own login page and its access is protected by a username and a password.

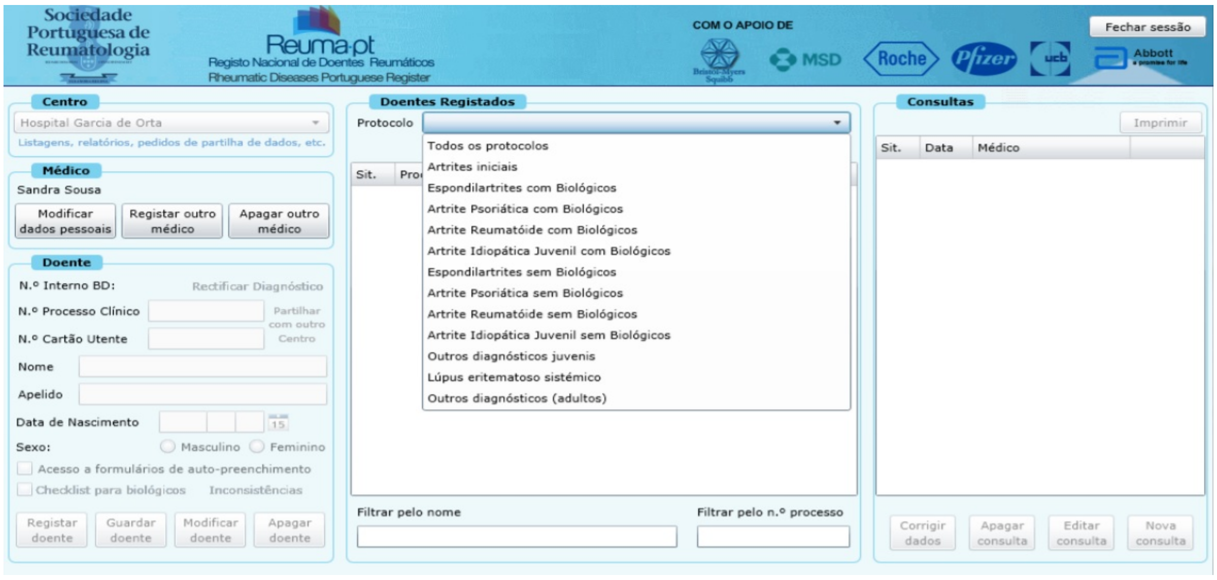
Inside the physician's area, the left side of the first screen (see Figure 2.1) displays sections concerning physicians and patients. Regarding the physicians' section, it is possible to:

- Edit the personal data of the physician that is logged ("Modificar dados pessoais" button);
- Add a new physician ("Registar outro médico" button);
- Delete a physician ("Apagar outro médico" button).

In the patients' section, the physicians can:

- Add a new patient ("Registar doente" button);
- Edit the patient's data ("Modificar doente" button);

[1] http://www.reuma.pt/
[2] http://www.reuma.pt/docs/NL201806.html
[3] https://www.microsoft.com/silverlight/


Centro: Hospital Garcia de Orta

Médico: Sandra Sousa
Sociedade Portuguesa de Reumatologia
Reuma.pt
Registo Nacional de Doentes Reumáticos
Rheumatic Diseases Portuguese Register
COM O APOIO DE
MSD
Roche
Pfizer
ucb
Abbott
Fechar sessão
Centro
Listagens, relatórios, pedidos de partilha de dados, etc.
Médico
Modificar dados pessoais
Registar outro médico
Apagar outro médico
Doente
N.º Interno BD:
Rectificar Diagnóstico
N.º Processo Clínico
Partilhar com outro Centro
N.º Cartão Utente
Nome
Apelido
Data de Nascimento
Sexo:
Masculino
Feminino
Acesso a formulários de auto-preenchimento
Checklist para biológicos
Inconsistências
Registar doente
Guardar doente
Modificar doente
Apagar doente
Doentes Registados
Protocolo
Todos os protocolos
Artrites iniciais
Espondilartrites com Biológicos
Artrite Psoriática com Biológicos
Artrite Reumatóide com Biológicos
Artrite Idiopática Juvenil com Biológicos
Espondilartrites sem Biológicos
Artrite Psoriática sem Biológicos
Artrite Reumatóide sem Biológicos
Artrite Idiopática Juvenil sem Biológicos
Outros diagnósticos juvenis
Lúpus eritematoso sistémico
Outros diagnósticos (adultos)
Sit.
Filtrar pelo nome
Filtrar pelo n.º processo
Consultas
Imprimir
Sit.
Data
Médico
Corrigir dados
Apagar consulta
Editar consulta
Nova consulta

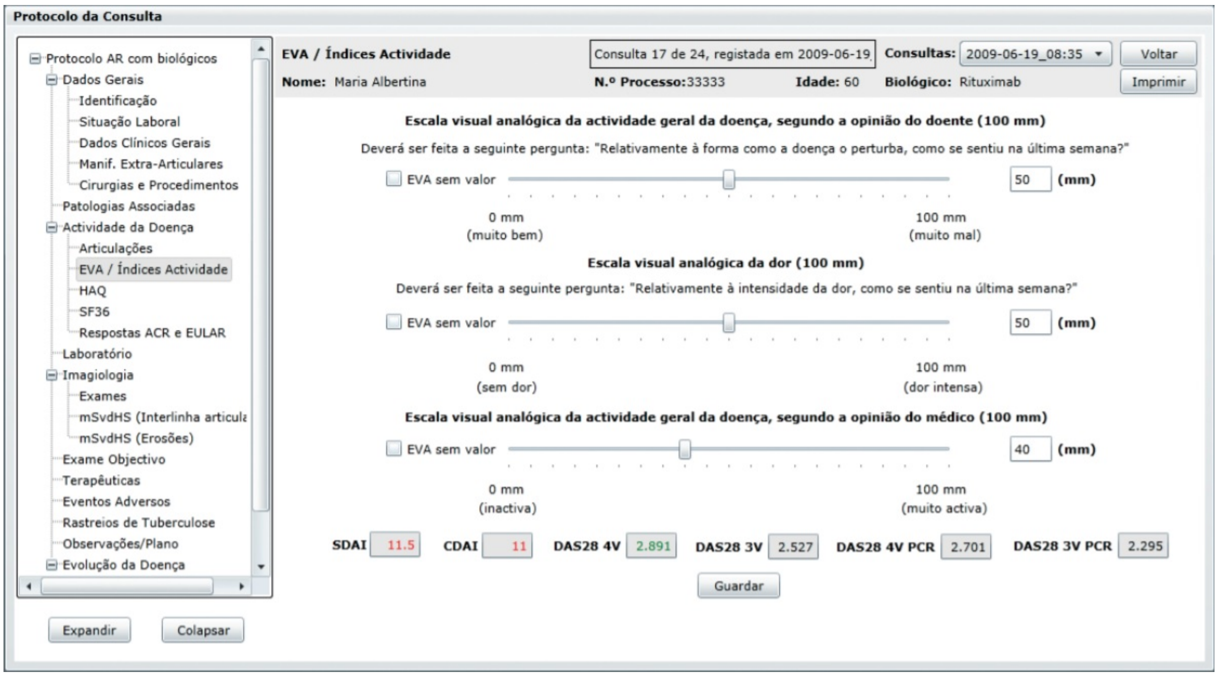
Figure 2.1: First screen displayed inside the physicians' area. Figure from www.reuma.pt/docs/Reumapt_EULAR_-2013.pdf, page 1.

- Save the updates made to the patient's data ("Guardar doente" button);
- Delete a patient ("Apagar doente" button).

In the middle of the first screen, a list of patients according to a protocol (i.e., the disease and its type of treatment) is displayed. One of them can be chosen. Finally, the right side of the first screen displays the visit's section. This section shows a list of visits and allows the physicians to:

- Edit a visit ("Editar consulta" and "Corrigir dados" buttons);
- Delete a visit ("Apagar consulta" button);
- Add a visit ("Nova consulta" button).

By clicking on the "Nova consulta" button, the visit screen opens (see Figure 2.2). The visit screen displays, on the left side, a menu list with submenus. For all the protocols, the menu list displayed on the left side incorporates common menus like identification data, demographic data, work status, life styles, body mass index, among others. In addition, there are menus that are specific to each disease. For instance, on the left side of Figure 2.2, we can see that, in relation to the biological treatment of RA, it is collected data regarding the patient's characteristics, treatments, disease and medical examinations. The right side of the figure shows the part of the screen regarding the "EVA / Índices Actividade" (the submenu that is selected). In this screen we can see, on the top side, the generic data about the patient and the visit. In the middle, are displayed several numeric variables that are entered through a horizontal scale or through a text box. For example, in the first horizontal scale, the user chooses the value (between 0 and 100) that corresponds to the patient's response to the following question: "regarding how the disease disturbs you, how did you feel in the last week?" (question translated into English). The bottom side displays several automatic computed fields that are calculated based on formulas that use the data entered in the horizontal scales and text boxes. One of those automatic computed fields is DAS28, which describes the severity of RA using clinical and laboratory data.


Protocolo da Consulta

- Protocolo AR com biológicos
  - Dados Gerais
    - Identificação
    - Situação Laboral
    - Dados Clínicos Gerais
    - Manif. Extra-Articulares
    - Cirurgias e Procedimentos
  - Patologias Associadas
  - Actividade da Doença
    - Articulações
    - EVA / Índices Actividade
    - HAQ
    - SF36
    - Respostas ACR e EULAR
  - Laboratório
  - Imagiologia
    - Exames
    - mSvdHS (Interlinha articula
    - mSvdHS (Erosões)
  - Exame Objectivo
  - Terapêuticas
  - Eventos Adversos
  - Rastreios de Tuberculose
  - Observações/Plano
  - Evolução da Doença

Expandir | Colapsar

EVA / Índices Actividade | Consulta 17 de 24, registada em 2009-06-19 | Consultas: 2009-06-19_08:35 | Voltar

Nome: Maria Albertina | N.º Processo: 33333 | Idade: 60 | Biológico: Rituximab | Imprimir

Escala visual analógica da actividade geral da doença, segundo a opinião do doente (100 mm)

Deverá ser feita a seguinte pergunta: "Relativamente à forma como a doença o perturba, como se sentiu na última semana?"

EVA sem valor | 50 (mm)

0 mm (muito bem) | 100 mm (muito mal)

Escala visual analógica da dor (100 mm)

Deverá ser feita a seguinte pergunta: "Relativamente à intensidade da dor, como se sentiu na última semana?"

EVA sem valor | 50 (mm)

0 mm (sem dor) | 100 mm (dor intensa)

Escala visual analógica da actividade geral da doença, segundo a opinião do médico (100 mm)

EVA sem valor | 40 (mm)

0 mm (inactiva) | 100 mm (muito activa)

SDAI 11.5 | CDAI 11 | DAS28 4V 2.891 | DAS28 3V 2.527 | DAS28 4V PCR 2.701 | DAS28 3V PCR 2.295

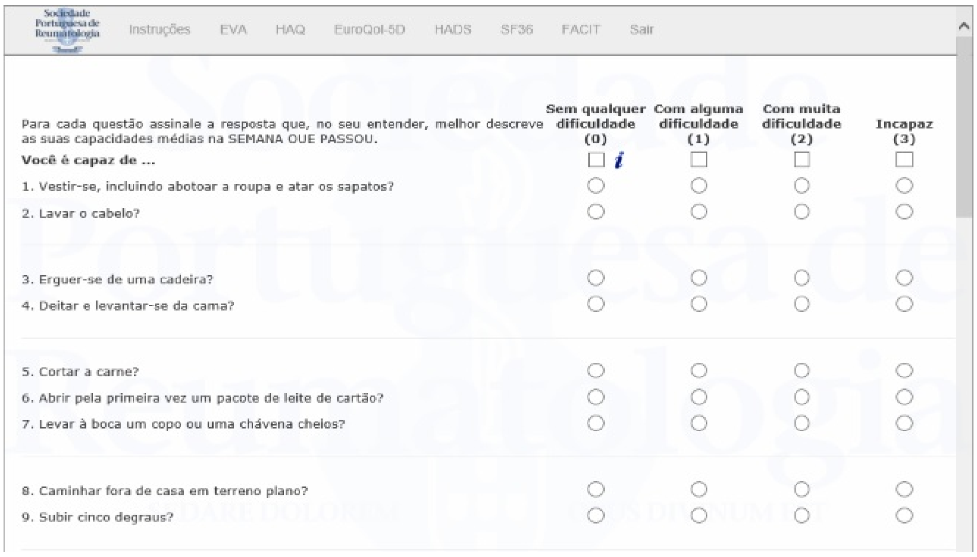
Guardar

Figure 2.2: Visit's screen example in Reuma.pt for RA. Figure from www.reuma.pt/pt_PT/docs/ONDOR_-reumapt.pdf, page 9.

In the patient's area, the patients can complete the Patient Reported Outcomes (PRO) before their next medical visit [15, 17]. PROs are questionnaires concerning their disease. Then, during the visit, the answers to questionnaires can be seen by the physician. This procedure enables to speed up and facilitate the assessment process of the patient's status [16]. Figure 2.3 shows an example of a screen with an excerpt of a questionnaire.

Patients can also visualize data related with their clinical process, namely:

- Personal identification data;
- Employment status and education level;
- Alcohol and tobacco consumption;
- Therapies and non-pharmacological treatments;
- Comorbidities and drug allergies;



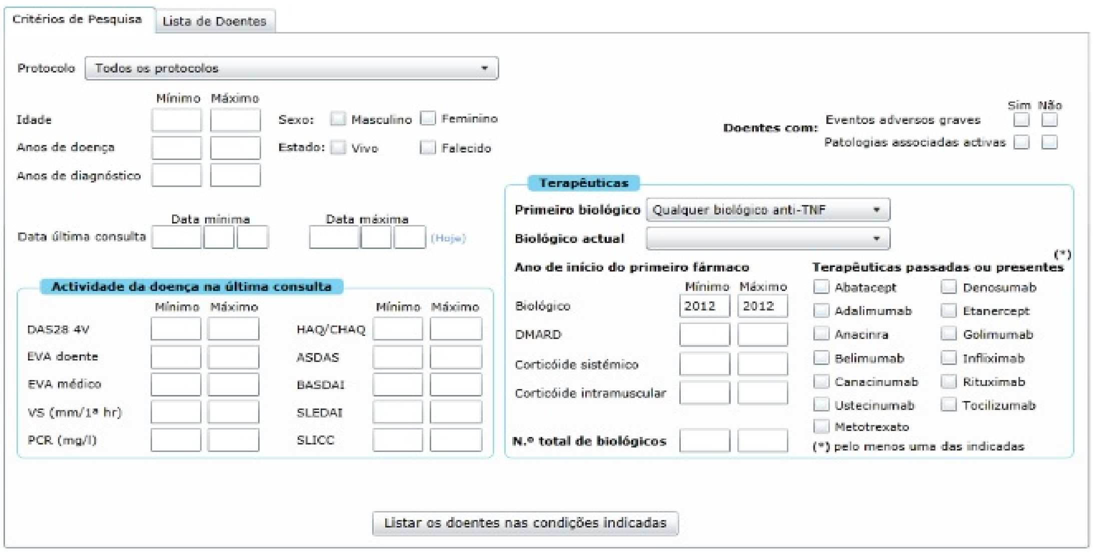
Para cada questão assinale a resposta que, no seu entender, melhor descreve as suas capacidades médias na SEMANA OUE PASSOU.	Sem qualquer dificuldade (0)	Com alguma dificuldade (1)	Com muita dificuldade (2)	Incapaz (3)
Você é capaz de ...	☐	☐	☐	☐
1. Vestir-se, incluindo abotoar a roupa e atar os sapatos?	○	○	○	○
2. Lavar o cabelo?	○	○	○	○
3. Erguer-se de uma cadeira?	○	○	○	○
4. Deitar e levantar-se da cama?	○	○	○	○
5. Cortar a carne?	○	○	○	○
6. Abrir pela primeira vez um pacote de leite de cartão?	○	○	○	○
7. Levar à boca um copo ou uma chávena chelos?	○	○	○	○
8. Caminhar fora de casa em terreno plano?	○	○	○	○
9. Subir cinco degraus?	○	○	○	○

Figure 2.3: Example of a screen with an excerpt of a questionnaire in Reuma.pt's patient area. Figure from www.reuma.pt/docs/NL201604.html.

- Laboratory analysis;
- Vaccines;
- Tuberculosis screenings;
- Evolution charts.

In Reuma.pt's platform, all the existing screens have the possibility to be printed, regardless of whether their fields are filled or not. Besides this, after entering data, it is possible to generate a pre-formatted report that contains all the inserted data and it can be printed or copied into an electronic clinical process. There is also the possibility to export the data stored so that it can be further analysed in specific statistical programs or data analysis packages such as SAS, State, R and SPSS.

Another functionality supported by Reuma.pt is the selection of patients through filters. Filters cover many variables that can change according to the chosen protocol (i.e., the disease and its type of treatment). We can insert range values for numerical variables and validations for other types of variables (for example the gender, where we can check the option "male"). Once the filters have been defined and applied, a list is returned in a table format with the patients that verify the selection criteria, and it can be exported to an Excel file in an CSV format. As an example, Figure 2.4 illustrates the screen where the user can define the filters to select the patients who initiated the biological therapy in 2012 and whose first biological was an *anti-TNF*. Both of these filters were defined in the therapeutics' section displayed on the right side of the screen displayed in Figure 2.4. In the "Primeiro biológico" combo box we defined the first biological. Additionally, in the "Biológico" text boxes (respectively "Mínimo" and "Máximo") we defined the interval of time (in years) during which the therapy was initiated. By clicking on the "Listar os doentes nas condições indicadas" button, the screen presented in the Figure 2.5 opens. Figure 2.5 shows the screen that displays the list of patients who satisfy the defined filters.



Critérios de Pesquisa | Lista de Doentes

Protocolo: Todos os protocolos

	Mínimo	Máximo
Idade		
Anos de doença		
Anos de diagnóstico		

Sexo: ☐ Masculino ☐ Feminino

Estado: ☐ Vivo ☐ Falecido

Doentes com:

	Sim	Não
Eventos adversos graves	☐	☐
Patologias associadas activas	☐	☐

Data última consulta: Data mínima / Data máxima (Hoje)

Actividade da doença na última consulta

	Mínimo	Máximo		Mínimo	Máximo
DAS28 4V			HAQ/CHAQ		
EVA doente			ASDAS		
EVA médico			BASDAI		
VS (mm/1ª hr)			SLEDAI		
PCR (mg/l)			SLICC		

Terapêuticas

Primeiro biológico: Qualquer biológico anti-TNF

Biológico actual:

Ano de início do primeiro fármaco

	Mínimo	Máximo
Biológico	2012	2012
DMARD		
Corticóide sistémico		
Corticóide intramuscular		
N.º total de biológicos		

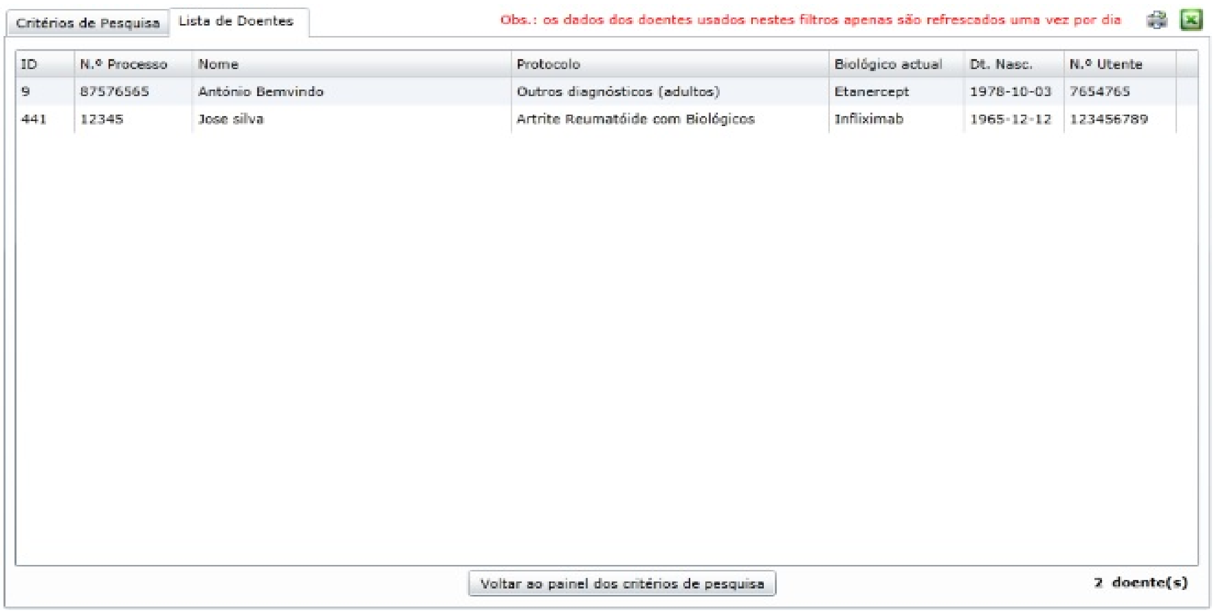
Terapêuticas passadas ou presentes (*)

☐ Abatacept ☐ Denosumab
☐ Adalimumab ☐ Etanercept
☐ Anacinra ☐ Golimumab
☐ Belimumab ☐ Infliximab
☐ Canacinumab ☐ Rituximab
☐ Ustecinumab ☐ Tocilizumab
☐ Metotrexato

(*) pelo menos uma das indicadas

Listar os doentes nas condições indicadas

Figure 2.4: The screen where we defined the filters to select the patients who initiated the biological therapy in 2012 and whose first biological was an *anti-TNF*. Figure from www.reuma.pt/docs/NL201407.html.



Critérios de Pesquisa | Lista de Doentes

Obs.: os dados dos doentes usados nestes filtros apenas são refrescados uma vez por dia

ID	N.º Processo	Nome	Protocolo	Biológico actual	Dt. Nasc.	N.º Utente
9	87576565	António Bemvindo	Outros diagnósticos (adultos)	Etanercept	1978-10-03	7654765
441	12345	Jose silva	Artrite Reumatóide com Biológicos	Infliximab	1965-12-12	123456789

Voltar ao painel dos critérios de pesquisa

2 doente(s)

Figure 2.5: The screen that contains the list of the patients who initiated the biological therapy in 2012 and whose first biological was an *anti-TNF*. Figure from www.reuma.pt/docs/NL201407.html.

Finally, there are some other functionalities that also facilitate the patient monitoring and follow-up, and that support clinical decision such as:

- Checklists of procedures to be performed before initiating the biological therapy;

- Pre-formatted letters for the family's doctor;
- Graphs showing disease evolution through follow-up;
- Summary tables showing the patient's clinical situation;
- Clinical data and metrics derived from the entered data in relation to treatments that are provided to the user, acting as a support mechanism for making clinical decisions.

The database of Reuma.pt was developed taking into account the suggestions of all center representatives in order to reach a general agreement. The center representatives are individuals that represent each rheumatology center where the patients have their visits. This agreement was an important aspect to consider due to the fact that Reuma.pt needed to standardize the data required to be collected in rheumatologic clinical records. Therefore, efforts were made to create protocols for each disease according to its treatment, which described, for example, what data was mandatory to be introduced and, consequently, what was optional. This effort enabled to structure the data to be collected and that is appropriate to perform data analysis afterwards. To support the storage of structured data, in the platform the insertion of data is not done through free text fields. Instead, the insertion of data is performed in fields like combo boxes, check boxes and numeric that contain a predefined set of values or that undergo a validation process (for example, dates). However, there are free type fields too, but they are specifically assigned to notes [3]. Additionally, there are automatic computed fields that are calculated based on formulas that use the entered data.

Based on the analysis of the data stored regarding patients and their visits, an execution report containing essential data about the effectiveness and safety of therapies is publicly released every year [16]. For example, the report of 2017[4] is mainly composed by tables and charts that show statistical data such as:

- The total number of patients per disease and the number of patients that are treated with biological agents according to each disease;
- The total number of visits per center and per disease;
- The total number of patients with and without an active biological, by center and by disease;
- The evolution of the number of patients by disease.

This report also contains statistical data regarding the patients' characterization and detailed information about the comorbidities, the safety and efficacy of therapeutics (as stated before) and its associated adverse effects.

[4] http://www.reuma.pt/pt_PT/docs/Reumapt_relatorio_execucao_201712.pdf

Each medical center that uses Reuma.pt has the possibility to extract statistical reports regarding data stored. In addition to a report for all diagnostics (a global report), the following three other types of reports can be extracted:

- An efficacy report for each disease. This report mentions the total number of patients with a given diagnosis, their ages, the follow-up time, the disease duration, clinical efficacy measures, among others;
- A safety report for each disease or for all the patients registered at that center. This report mentions the adverse events, both in patients receiving DMARDs and in patients undergoing biological therapy;
- A tuberculosis report for each disease or for all patients registered at that center.

Reuma.pt enables to report to INFARMED[5] the drugs' safety with respect to their side effects in a short, medium and long term. In addition, Reuma.pt stores comparative patient cohorts, i.e., groups of patients with common characteristics that are observed during a predefined or undefined time interval. There are groups composed of patients that are treated with DMARD (a type of drugs) and other therapeutic strategies. These groups enable to compare the outcomes of the standard conventional treatments with the biotechnological therapeutics, for instance.

Although data analysis currently appears to be based only on the data collected from the structured fields, Reuma.pt signed an agreement with INESC-ID[6,7] in order to expand the data analysis to the non-structured fields. This agreement was signed in December of 2017 and aims at applying Artificial Intelligence and Automatic Learning techniques for processing non-structured fields [16]. Its purpose is to infer new and relevant knowledge from these types of fields, such as the fields specifically assigned for notes in Reuma.pt.

Since Reuma.pt was launched, several scientific projects based on Reuma.pt's data were presented in major rheumatology meetings and published in national and international peer-reviewed journals [17].

2.1.2 Derma.pt

Derma.pt[8], from the Portuguese Society of Dermatology and Venereology (SPDV), is an online software platform that contains a database of patients with psoriatic diseases. Currently, Derma.pt contains more than 800 patients and 2800 visits registered among 31 clinical centers spread across Portugal. Its final goal is to register all the patients in Portugal with psoriatic diseases that are treated with biological treatments in order to monitor treatment efficacy and safety.

[5]http://www.infarmed.pt/
[6]https://www.inesc-id.pt/
[7]https://tecnico.ulisboa.pt/en/events/signing-ceremony-of-the-cooperation-agreement-between-inesc-id-and-spr/
[8]http://www.derma.pt/

Unlike other clinical data registries, we believe that Derma.pt has been built based on Reuma.pt (described in Section 2.1.1), where it is likely that the way the user interacts with their user interfaces is similar. Derma.pt, such as Reuma.pt, is built on Microsoft Silverlight and it is accessed through a web browser. Contrarily to Reuma.pt, Derma.pt was designed only for physicians. Nevertheless, as in Reuma.pt, the access to Derma.pt's platform is also protected by a username and a password. In addition, analogously to Reuma.pt, it also stores comparative patient cohorts, in this case with psoriasis, that are treated with conventional therapies.

As far as we could find, currently there is only one research work that was developed based on Derma.pt's data. Particularly, it is an observational study that addresses the psoriasis in the elderly [7].

2.1.3 National Cancer Registry (RON)

RON is a centralized registry of oncological patients supported by a single electronic platform. This clinical data registry was developed by the Shared Services of the Ministry of Health (SPMS)[9] and is administrated by the board of directors of the Portuguese Institute Oncology Francisco Gentil Hospital Group (GHIPOFG). RON was created and regulated by the Portuguese *Decreto Lei* n°53/2017 (DL 53/2017) of July 14[10] that states that it started on 1 January 2018. It is important to note that since we have not been able to find concrete evidence of RON to be actually built, what is described about RON is based in what DL 53/2017 states.

RON aims at collecting and analysing data from oncological patients diagnosed and/or treated in Portugal. Taking this into consideration, RON is able to monitor the activity of the institutions and the therapeutic effectiveness. In addition, in partnership with INFARMED, RON is also able to monitor the applicability of drugs and medical devices. This clinical data registry also enables epidemiological surveillance and the development of research work.

As also stated in the DL 53/2017, the registration of all new cases of cancer diagnosis in RON's platform by all health establishments and services located in Portugal is mandatory. Specifically, this registration is mandatory within a maximum period of 9 months from the date of the diagnosis. In addition, the subsequent update, at least annually, of the oncological disease stage, oncological therapies used, and the patient's vital state is required too. It is mentioned that RON must include the data from the Portuguese pediatric cancer registry and from the 4 regional cancer registries that existed before RON, namely the regional cancer registry of north[11], the regional cancer registry of south, the regional cancer registry of center, and the regional cancer registry of Azores.

The DL 53/2017 established that the following data must be collected and stored in the RON's platform:

[9] http://spms.min-saude.pt/
[10] https://dre.pt/application/file/a/107688306/
[11] http://www.roreno.com.pt/

- Basic personal information, such as name, gender, birth date, address, patient's number, profession and patient's natural state;
- The identification of the institution where the patient was followed and the clinical process's number;
- The date and the results of medical examinations performed for diagnosis and cancer staging, that are relevant to the medical history;
- The current identification of the code of the International Classification of Diseases[12] corresponding to the tumor diagnosed;
- The updated pediatric classification for each group of tumors, for pediatric registry;
- The characterization of the tumor, not limited to the primary location, morphology, staging, receptors, molecular and tumor markers;
- The data related to the diagnosis and to the genetic study of the tumor (when is applicable);
- The date of both diagnosis and the initiation of treatment, as well as the various types of treatment, such as surgery, radiotherapy and chemotherapy;
- The characterization of each treatment line;
- The annual register of the general condition of the patient;
- The condition of the tumor and its modifications, including those that depend on the treatments, and the best outcome obtained from the treatments of the tumor at the end of each treatment line;
- The date and the cause of death, if applied.

To help collecting these kinds of data, RON supports automatic interoperability mechanisms that enable to communicate with several external national databases. These databases are related with patients, their death certificates, diagnosis, tumors and oncological screens. These databases are also related with local information systems, such as prescription programs for hospital drugs, for instance.

DL 53/2017 also mentions that RON is accessed through a software platform that is available in the Health Informatics Network (RIS). As in Reuma.pt and in Derma.pt, its access is protected by a username and a password. There are 8 different types of users that can access the platform, whose permissions are limited according to the user's profile as follows:

- The *local recorder profile* is assigned to the health workers of the health institution. This profile enables to create and modify cases of cancer diagnosis. It also enables to extract unidentified aggregated data reports from all the data of the existing cases in a given institution;

[12]http://www.who.int/classifications/icd/en/

- The *regional recorder profile* is assigned to the health workers from each of the oncology institutes. This profile enables to create, modify and delete cases of cancer diagnosis and to solve doubts or inconsistencies. It also enables to extract unidentified aggregated data reports from all the data of the existing cases in the concerned region. In addition, the user has access to statistical information that can be obtained by searching the database;

- The *profile of the user responsible for the regional cancer data* is assigned to the person appointed by each of the chairman of the boards of directors of the oncology institutes. This profile enables to access, modify and extract unidentified aggregate data reports from all the data in the concerned region. It also enables to monitor the quality of such data, its consolidation and resolving data conflicts;

- The *profile of the user responsible for the regional oncological screening programs* is assigned to the person appointed by the respective regional health administration. This profile enables to access the data from the respective region, referring to the nosologic groups (i.e., groups that are dedicated to the study and classification of diseases) targeted by the screening program;

- The *profile of the health manager* is assigned to the regional health delegate and to the director of the public health department of the regional health administrations. This profile has access to statistical information that can be obtained by searching in RON and that facilitates the development of epidemiological studies. These studies refer in particular to cancer screenings, and enable the definition of regional public health policies;

- The *profile of the RON's coordinator* is assigned to the person appointed by the board of directors of the GHIPOFG (a hospital group). This profile enables to query and extract unidentified aggregated data reports, export anonymized data of all RON's data, monitor data quality and its consolidation, and resolve data conflicts;

- The *pediatric coordinator profile* is assigned to the person appointed by the RON's coordinator. This profile enables to access, modify and extract unidentified aggregated data reports from all pediatric cancer cases. It also enables to monitor the quality of such data and its consolidation, and resolve data conflicts;

- The *administrator profile* is assigned to the entity responsible for administering RON and to its designated workers. This profile enables to manage, monitor and develop the computer application in terms of access profiles, reference tables and the administration of RON.

The RON's coordinator need to elaborate, annually, a report reflecting the national situation verified in a period which preferably does not exceed the previous three years. The responsible for the regional cancer data needs to elaborate, annually, a report regarding the situation verified in the penultimate year concerning the following aspects:

- New cases by pathology, gender and age group;
- Incidence rates by location, gender and age group;
- New cases according to the cancer staging;
- New cases by the area of influence of each regional health administration, by each territorial unit and by each Health Centers Grouping;
- Number of deaths by year and by diagnosis;
- Survival rates at 1, 3 and 5 years, by diagnosis and staging, for patients with more than one year of follow-up after the date of diagnosis;
- Data quality. For example, if the data has been revised in order to correct misspellings or other inconsistencies;
- Access of users to the cancer registry.

In addition, both of these reports should be published in the portal of the National Health Service (SNS).

2.1.4 SI.VIDA

The Information System for HIV/AIDS infection, SI.VIDA, is a clinical data registry that covers in part the medical specialty of Infectiology. In particular, SI.VIDA enables to monitor and follow-up patients with HIV/AIDS. It started in 2008 and has the purpose to improve the quality of epidemiological surveillance of HIV infection and the quality and efficiency of the health services provided by the SNS hospitals. Having this goal in mind, SI.VIDA aims primarily at:

- Ensuring the register of the epidemiological, clinical and laboratorial data regarding the people infected with the Human Immunodeficiency Virus (HIV) and the data necessary to monitor each patient;
- Evaluating through laboratory, clinical and therapeutic indicators, the quality and efficiency of the care provided to the people infected with HIV;
- Contributing to ensure that the health outcomes that have been anticipated and proposed in the targets defined for the HIV infection are achieved.

As cited in the *Despacho* n°8379/2018[13] published in the *Diário da República* n°185/2018, in June 2017, SI.VIDA was implemented in more than 90% of all the SNS hospitals that follow the patients with HIV. It is also mentioned that SI.VIDA should be undergoing an integration process with the SClínico Hospitalar system - an evolving information system that is supposed to be a single software application, common to all healthcare providers and patient-centered - in order to prevent the existence of duplicate records in SI.VIDA. In addition, as in RON, the use of SI.VIDA is mandatory, since December 2012, in the hospital units in which it is implemented[14].

SPMS is the entity currently responsible for the implementation of SI.VIDA in the hospital units. The SPMS releases monthly management reports, produced by the SI.VIDA, to the Central Administration of the Health System (ACSS), the Directorate-General for Health (DGS) and the National Health Institute Doutor Ricardo Jorge (INSA). These reports refer to the data and the quality of the data entered in SI.VIDA.

2.1.5 Discussion

In this section we addressed four Portuguese electronic clinical data registries. Each of these clinical data registries covers a different medical specialty and all are summarized in Table 2.1. This table describes, for each clinical data registry, its medical specialty, monitoring purposes, the data collected, data analysis aspects, whether the data is structured, to whom it is accessible, and whether its use is mandatory by law.

All the four clinical data registries have common monitoring purposes with respect to treatment. In the case of Reuma.pt and RON, they also collect data common to various subjects, such as patient's characteristics, medical examinations, disease-specific data and treatments. Apart from Derma.pt, all clinical data registries have data analysis mechanisms, thus being able to extract data aggregated reports, for instance. As far as we know, Reuma.pt is the only clinical data registry where the data is structured, being composed largely by enumerated fields and by a few free text fields. In addition, Reuma.pt is the only clinical data registry that is accessible both to physicians and patients. The remaining clinical data registries can only be accessed by physicians or by other workers from the health sector. However, in the case of SI.VIDA, it is unknown by whom it can be accessed. The use of RON and SI.VIDA is mandatory by the Portuguese law, contrarily to Reuma.pt and Derma.pt that are of optional use.

[13]https://dre.pt/application/file/a/108204783

[14]http://www.sg.min-saude.pt/NR/rdonlyres/B9EBB192-952E-4C97-94FD-6B54A9F75A58/29226/1739517396.pdf

Table 2.1: Summary of the specialty-specific clinical data registries

Property / Clinical data registry	Reuma.pt	Derma.pt	RON	SI.VIDA
Medical specialty	Rheumatology	Dermatology (psoriatic)	Oncology	Infectiology (HIV/AIDS)
Monitoring purposes	- Treatment - Long-term comorbidities - Disease	- Treatment	- Treatment - Disease - Institutions - Medical devices	- Treatment - Patients
Data collected	- Patient's characteristics - Examinations - Disease-specific - Treatment - Questionnaires	Unknown	- Patient's characteristics - Diagnosis - Examinations - Disease-specific - Treatment	Unknown
Data analysis	- Extraction of aggregated data reports - Summary tables of the patient's clinical situation - Disease evolution graphs	Unknown	- Extraction of aggregated data reports - Access to statistical information	- Extraction of aggregated data reports
Structured data	Yes	Unknown	Unknown	Unknown
Accessible to	- Physicians - Patients	- Physicians	- Health workers	Unknown
Mandatory by law	No	No	Yes	Yes

2.2 Cardiological Clinical Data Registries

The Portuguese Society of Cardiology (SPC)[15] sponsors several Portuguese cardiological clinical data registries[16] that were mainly created by the respective working groups. These clinical data registries are centralized in the National Center for Data Collection in Cardiology (CNCDC)[17], which is the entity responsible for processing and analysing the data maintained by clinical data registries. In this section, we address 6 of the 12 existing clinical data registries from the SPC, since we could not find publicly available information describing all the clinical data registries. In particular, we address the following cardiological clinical data registries:

- The Portuguese Registry of Acute Coronary Syndromes (ProACS) (Section 2.2.1)
- The National Registry of Arrhythmogenic Right Ventricular Myocardiopathy (RNMAVD) (Section 2.2.2)

[15]https://spc.pt/
[16]https://spc.pt/registos/
[17]https://spc.pt/cncdc/

- The Portuguese Registry of Hypertrophic Cardiomyopathy (PRo-HCM) (Section 2.2.3)
- The Portuguese Myocarditis Registry (PMR) (Section 2.2.4)
- The Portuguese Registry of Non-compaction Cardiomyopathy (PRNC) (Section 2.2.5)
- The Portuguese Registry on Interventional Cardiology (RNCI) (Section 2.2.6).

As in Section 2.1, we mention the purposes and some considerations about the history and/or current state of each cardiological clinical data registry.

2.2.1 Portuguese Registry of Acute Coronary Syndromes (ProACS)

ProACS[18], that is promoted by the SPC and coordinated by the CNCDC, is a continuous observational clinical data registry. Regarding the Acute Coronary Syndromes (ACS), ProACS aims at assessing, in Portugal, the clinical profile of patients, the diagnostic and therapeutic management, and the medium-term prognosis. It started in 2002 and currently contains more than 45.000 records covering 49 centers (i.e., university hospitals, hospitals with cardiac surgery and hospitals with a hemodynamic laboratory) that are distributed across the country. Although these seems to be good figures, it would be desirable to increase the participation of the centers to ensure a greater representativeness of the national population. This is an aspect to emphasize since over time some centers stopped participating and others never participated at all[19]. However, the use of ProACS by the centers is voluntary, contrarily to the use of other clinical data registries, such as RON, which is mandatory by the Portuguese law.

ProACS collects the following data:

- Demographic data;
- Basic patient information;
- Clinical and laboratory evolution data;
- Therapeutics followed;
- Percutaneous intervention data;
- Patient discharge and follow-up data at six months or at the first year.

Initially, this data was collected in paper and only afterwards it was inserted into a database. Since 2004, data is entered directly by electronic means. All the patients' data is centralized in the database of the CNCDC in Coimbra, in which the patient's identification is anonymous. Periodically, each center receives treatment quality assessment data, which reports the clinical results of the center and the clinical results

[18]https://registos.spc.pt/RegistoSCA/Public/Login.aspx?ReturnUrl=%2fRegistoSCA%2f
[19]https://spc.pt/registo-nacional-de-sindromes-coronarias-agudas/

at the national level. This data enables physicians to compare the treatment approaches used in their centers with the national standards. With this comparison, physicians can identify potential aspects to improve health care.

ProACS has an important role in research work in Portugal regarding the ACS. From 2002 to 2017, 199 works were presented in national scientific meetings and 151 works in international scientific meetings, all based on ProACS's data. In addition, 6 works were published in national journals and 7 in international journals [18].

2.2.2 National Registry of Arrhythmogenic Right Ventricular Myocardiopathy (RNMAVD)

RNMAVD[20] was created by the Working Group on Myocardial and Pericardial Diseases (GEDMP)[21] of the SPC and, analogously to ProACS, it operates within the scope of the CNCDC. RNMAVD is an electronic clinical data registry that aims at assessing how patients with Arrhythmogenic Right Ventricular Myocardiopathy (ARVM) are diagnosed and treated in Portugal. This clinical data registry also intends to monitor the evolution of ARVM during the patients' follow-up. These purposes enable to obtain more knowledge about the ARVM in Portugal and, thereby, to make it possible to issue clinical recommendations for an increasingly better diagnosis and treatment in the future. RNMAVD started on 1 October 2015 as a study, at a time where the knowledge about the ARVM disease was scarce.

To be able to collect the patients' data, RNMAVD requires an informed oral and written consent[22] from each patient. Data that is collected by RNMAVD is entered through an electronic CRF that was specifically created for this clinical data registry, and covers the following topics[23]:

- Demographic characteristics and family history;
- Diagnosis/first evaluation, diagnostic criteria, medical therapy and follow-up;
- Conventional electrocardiogram, high resolution electrocardiogram and echocardiogram;
- Holter (a medical examination that records cardiac electrical activity), cardiac magnetic resonance, pacemaker and cardioversor implantable defibrillator;
- Cardiac stress test (a medical examination);
- Genetic study.

[20] http://registos.spc.pt/RegistoMAVD/
[21] https://tinyurl.com/scp-gedmp/
[22] https://spc.pt/pdf/cncdc/registos/mavd/Consentimento%20MAVD%20-%20revisto.pdf
[23] https://spc.pt/pdf/cncdc/registos/mavd/LISTA%20VARIAVEIS%20MAVD%20revisto.pdf

Each CRF is exported to CNCDC, which is the central entity responsible for the data analysis and processing. CNCDC releases descriptive reports of the usual clinical practice in relation to the ARVM disease. Due to legal privacy reasons, the identification of patients is only known to the physician that is responsible for the RNMAVD at the hospital/center where the respective patient is followed. Due to this, the patient's identification data is protected by encryption and all the results of the data analysis are only published in aggregate form.

2.2.3 Portuguese Registry of Hypertrophic Cardiomyopathy (PRo-HCM)

PRo-HCM[24] was designed and implemented by the GEDMP of the SPC, and is centralized and managed at CNCDC. PRo-HCM aims at collecting data regarding the Hypertrophic Cardiomyopathy (HCM) disease in Portugal. In particular, this clinical data registry aims at collecting epidemiological, sociodemographic and clinical data, and data related to the diagnosis' standards, treatment, follow-up, and outcomes. Additionally, PRo-HCM was created to support and develop a reliable source of information for the physicians, patients and families, on appropriateness, effectiveness and quality of care. This clinical data registry started in June 2013 and, at the end of 2015, it contained 1.042 patients registered among 29 centers across Portugal [4]. In 2011, this represented only 5% of the estimated number of patients with HCM in Portugal [5]. Each of the participating centers has a principal investigator/physician that possesses a unique username and a password to access the PRo-HCM's platform. In this clinical data registry, as in RNMAVD, a written informed consent from each patient is required to be able to collect his/her data. Data to be maintained in PRo-HCM is also entered through an electronic CRF that is composed by the following 7 sections [4]:

- Patient identification and demographic/epidemiological data;
- Past history and baseline clinical data;
- Mortality and risk stratification (i.e., assigning patients to a particular group according to their health status);
- Diagnostic tests, were the researchers enter the medical examinations performed at the time of the first assessment;
- Genetic testing, family screening, and genetic counseling;
- Treatment;
- Clinical course, follow-up, and outcomes regarding the last assessment.

[24]http://registos.spc.pt/RegistosMiocardiopatia/

Since the creation of this clinical data registry, at least 2 studies have been published based on the PRo-HCM's data collected between April 2013 and the end of 2015 [2, 4].

2.2.4 Portuguese Myocarditis Registry (PMR)

The PMR[25], such as PRo-HCM, was created by the GEDMP of the SPC. The PMR was developed because the reality on myocarditis was unknown. In particular, there were no standardized strategies for diagnostics and treatment, nor treatment protocols or published data regarding follow-up[26]. Additionally, a complete panel of tests was usually performed, whose clinical utility was doubtful. Taking this into account, PMR aims at evaluating the Portuguese reality regarding the myocarditis disease in terms of diagnostic and therapeutic management, and its prevalence (i.e., the number of existing cases in a given population and at a given time point). This clinical data registry also intends to know the short-term evolution (precisely, one year) of myocarditis situations. In addition, PMR aims at reaching some conclusions regarding the possible need to standardize behaviors and actions, in order to improve diagnosis, and the effectiveness and quality of the treatment regarding myocarditis. PMR started in April 2013, and in April 2015 it contained 248 patients registered among 18 centers spread across Portugal [1]. This data was part of a prospective survey of these 2 years throughout the country. As in RNMAVD and in PRo-HCM, data is entered in an electronic CRF and is centralized and managed at CNCDC.

2.2.5 Portuguese Registry of Non-compaction Cardiomyopathy (PRNC)

PRNC[27] is an electronic clinical data registry that was created by the GEDMP of the SPC. Such as ProACS and RNMAVD, this clinical data registry operates within the scope of the CNCDC. PRNC aims at assessing how patients with Non-compaction Cardiomyopathy (NC) are diagnosed and treated in Portugal, and to monitor the evolution of the disease during the patients' follow-up. This enables to obtain more knowledge about the NC disease in Portugal and, thereby, to make it possible to issue clinical recommendations for an increasingly better diagnosis and treatment in the future. PRNC started on 1 October 2015 with the purpose of, after 1 year, including all the patients already diagnosed that were followed up in the various Portuguese centers (i.e., hospitals, for instance).

As in the other registries created by the GEDMP, PRNC requires an informed oral and written consent[28] from each patient to be able to collect the corresponding patient's data. As is usual in the SPC's registries, the data is also entered in an electronic CRF that was specifically created for this clinical data registry. The CRF is composed of different sections where all the patient's data will be inserted, namely

[25] http://registos.spc.pt/RegistosMiocardite/
[26] https://www.youtube.com/watch?v=RzBwoyito2s/
[27] http://registos.spc.pt/RegistoMNC/
[28] https://spc.pt/pdf/cncdc/registos/mnc/Consentimento%20MNC%20-%20revisto.pdf

regarding the following subjects[29]:

- Identification;
- Clinical data at the first observation;
- Progression data;
- Outcome of complementary examinations performed;
- Therapy performed;
- Follow-up clinical data.

Each CRF is exported to CNCDC, which is the central entity responsible for data analysis and processing. CNCDC releases descriptive reports of the usual clinical practice in relation to the NC disease. Due to legal privacy reasons, the identification of the patients is only known by the physician that is responsible for the PRNC at the center where he/she is followed. Therefore, the patient's identification data is protected by encryption and all the results of the data analysis are only published in aggregate form. With regard to security issues, each responsible investigator/physician of the centers and their respective appointed co-researchers possess a unique username and a password to access the PRNC's platform.

2.2.6 Portuguese Registry on Interventional Cardiology (RNCI)

The RNCI[30] was created and sponsored by SPC. RNCI aims at collecting, in a continuous way, all the activity in the scope of Interventional Cardiology (IC) that takes place in all laboratories/centers of IC in Portugal. This is intended to:

- Monitor characteristics, evolution, prognostic indicators and management of patients submitted to percutaneous procedures in Portuguese hospitals;
- Identify the appropriateness of clinical and interventional practice recommendations for diagnosis and treatment of the diseases associated with the IC.
- Monitor the evolution of the related diseases.

RNCI is based on the European Data Standards for Clinical Cardiology Practice (CARDS). Hence, the data collected is structured and can be categorized into several general topics: demographics, past history, presenting symptoms, procedure/event, follow-up, among others [8]. RNCI's design, implementation, development, monitoring and support is the responsibility of the Portuguese Association of

[29] A more detailed list of the collected data can be found at https://spc.pt/pdf/cncdc/registos/mnc/LISTA%20VARIAVEIS%20DE%20MNC.pdf
[30] http://registos.spc.pt/RegistosNacionaisSPC/

Interventional Cardiology (APIC). Organically, RNCI relies on a coordinator, a scientific commission, principal investigators, and others investigators[31]. The coordinator, among other functions, submits to the APIC half-yearly reports that contain information about the status of the clinical data registry. In addition, the coordinator publicly presents global results regarding the RNCI. The scientific commission ensures that the RNCI's data is reliable and that it is used correctly. For this, the scientific commission gives its opinion on the study proposals and scientific projects that intend to use the RNCI's data. The principal investigators (which can be physicians, for example), among other tasks, have the duty to:

- Ensure that all data are properly filled for each procedure and safeguard the reliability of the data;
- Ensure the regular and adequate transfer of data from their respective center to the CNCDC;
- Promote and expedite the execution of clinical follow-ups, and export them in a timely manner to the CNCDC.

RNCI started in 2002 as a study. Since 2013 all the Portuguese centers that deal with IC export data to the RNCI, which currently contains more than 150.000 records among all these centers. In addition, some research works were published based on the RNCI's data [12–14], which, for instance, report the trends of the procedures used in the treatment of a particular disease. Concretely, Pereira *et al.* [13] report trends in coronary angioplasty for the treatment of ST-elevation myocardial infarction in Portugal between 2002 and 2013.

2.2.7 Discussion

In this section we addressed six cardiological clinical data registries. These clinical data registries are summarized in Table 2.2. This table describes, for each clinical data registry, its medical specialty/disease, monitoring purposes, the data collected, data analysis aspects, whether the data is structured, and by whom it is accessible.

These clinical data registries only cover the medical specialty of Cardiology. Regarding the specialty of Cardiology, each of the clinical data registries focus on a disease or a set of diseases. These clinical data registries have common monitoring purposes mainly in relation to treatment, diagnosis and disease. Apart from PMR, they also collect data common to various topics, such as patient's characteristics, diagnosis, medical examinations, treatment and patient's follow-up. In the case of PMR, we were not able to find information about the data it is collected. All these cardiological clinical data registries are primarily focused on data entry, and the analysis and processing of the data is done by the CNCDC, an external entity to the clinical data registries. In three of these cardiological clinical data registries the data is structured, being composed largely by enumerated fields and by a few free text fields. We believe

[31] https://spc.pt/documents/20143/109427/Regulamento+RNCI+APIC+2017+Proposta+Final.pdf/f5c555ed-9a56-7c13-8ca8-c5a0622ef295

Table 2.2: Summary of the cardiological clinical data registries

Property \ Clinical data registry	ProACS	RNMAVD	PRo-HCM	PMR	PRNC	RNCI
Medical specialty/disease	Cardiology (ACS)	Cardiology (ARVM)	Cardiology (HCM)	Cardiology (myocarditis)	Cardiology (NC)	Cardiology (IC)
Monitoring purposes	- Treatment - Diagnosis - Prognosis - Patients	- Treatment - Diagnosis - Disease	- Treatment	- Treatment - Diagnosis - Disease	- Treatment - Diagnosis - Disease	- Disease - Prognosis - Patients
Data collected	- Patient's characteristics - Examinations - Treatment - Patient's follow-up	- Patient's characteristics - Family history - Diagnosis - Examinations - Treatment - Patient's follow-up	- Patient's characteristics - Diagnosis - Examinations - Treatment - Patient's follow-up	Unknown	- Patient's characteristics - Diagnosis - Examinations - Treatment - Patient's follow-up	- Patient's characteristics - Patient's history - Diagnosis - Treatment - Patient's follow-up
Data analysis	None	None	None	None	None	None
Structured data	Unknown	Yes	Unknown	Unknown	Yes	Yes
Accessible to	Unknown	Physicians	Physicians	Unknown	Physicians	Physicians

that the two other clinical data registries, PRo-HCM and PMR, also have their data structured, since the data is entered in forms (specifically, CRFs) too. However, this is not known with certainty. Finally, as far as we know, four of the six cardiological clinical data registries are accessed only by physicians. In the remaining two clinical data registries (ProACS and PMR) this information is unknown.

2.3 Research Clinical Data Registries

There are clinical data registries that have been created in the context of research projects. In this section, we address two Portuguese clinical data registries developed in research projects that the author is aware of. Namely, we describe the Umedicine system in Section 2.3.1 and the PRECISE Stroke system in Section 2.3.2, thus covering two additional medical specialties. Umedicine system is a clinical data registry for Urology. It has internal mechanisms to perform data analysis, contrarily to most clinical data registries described in Sections 2.1 and 2.2, in which this task is performed externally. The PRECISE Stroke system is a clinical data registry for Neurology. In particular, it maintains data related to stroke patients.

2.3.1 Umedicine

Umedicine is a clinical data registry for Urology. It was developed in the context of a research project with the advise of a Urology physician. Umedicine has not been used by physicians yet, contrarily to other clinical data registries that were described in Sections 2.1 and 2.2.

The current version of Umedicine was developed using the Spring framework[32]. Umedicine has a relational database and provides a user interface. This database stores clinical data about the following topics:

- *Patients*: stores patient's characteristics, personal history and other observations about the patient
- *Treatments*: stores Urology treatments, such as surgical and drug treatments
- *Medical examinations*: stores the name of the examination, the date it was performed, and the names and values of the parameters associated with each one
- *Symptoms*: stores the name of the symptom, the name of the disease and the dates on which both were detected
- *Questionnaires*: stores data about internationally accepted questionnaires where the patient condition is assessed, in particular the name of the questionnaire, its questions and answers, and the date it was filled in.

Additionally, this database also contains data about users who have permissions to access the Umedicine's user interface.

As we have come across in other clinical data registries, the access to the Umedicine's system is also protected by a username and a password. It can be accessed by four types of users [10] who have the following permissions:

- *Patients* can view and add a subset of the data about themselves
- *Non-administrator physicians* can add patient-type users and edit/view patient data
- *Administrator physicians* can add users from any type and view/edit patient data
- *Clerks (administrative personnel)* can add patient-type users and enter several types of patient's personal data.

The Umedicine user interface was designed and implemented to minimize the manipulation of textual data. In concrete, the interface presents predefined values to the user. An additional characteristic of the user interface is the organization of the screens shown so that the physician gets an overview of all

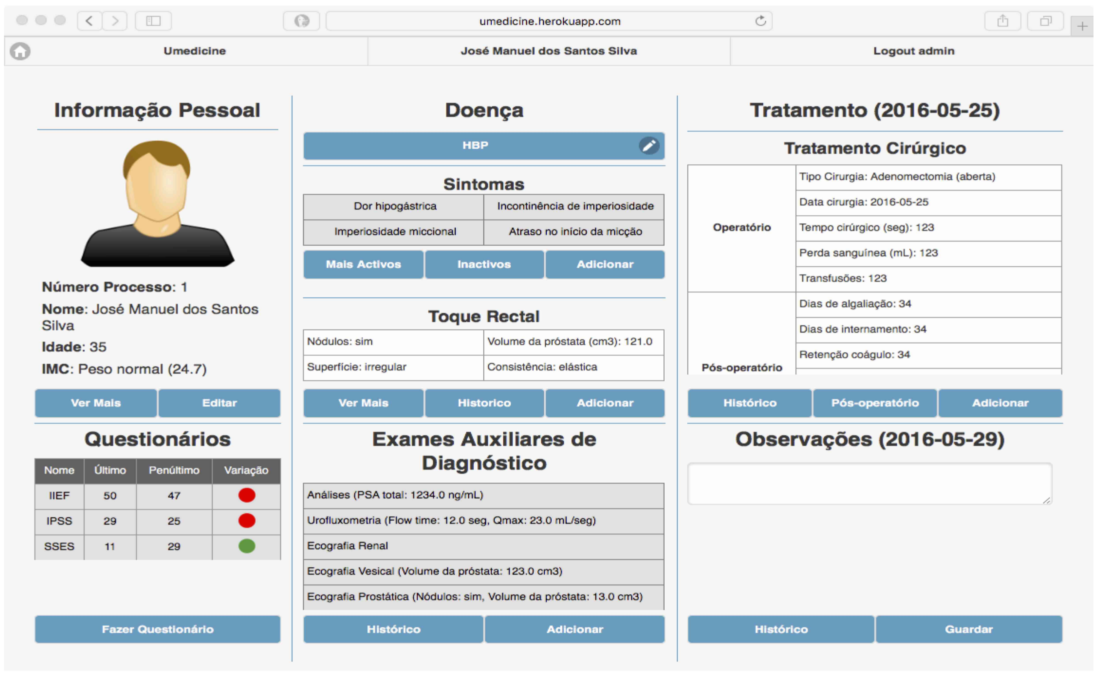
[32]https://spring.io/

the data from each patient. Both characteristics were required by the physician who co-supervised the project.

When a physician accesses the Umedicine user interface and selects a patient, the patient's information screen is shown. This screen is illustrated in Figure 2.6 and is organized in six sections [9]:

- *Personal Information* ("Informação Pessoal") is displayed on the left upper side. This section enables to view and modify personal data of the patient such as patient's name and his/her birth date;
- *Disease* ("Doença") is displayed on the middle upper side. This section enables to select the medical condition of the patient, and view or modify data about symptoms. This section also includes rectal examination data on the bottom side, where the physician can add new symptoms and examination results, and view symptom and rectal examination histories;
- *Treatment* ("Tratamento") is displayed on the right upper side. This section shows data regarding the treatments that are being performed. In addition, the physician can also access the treatment history and add new treatments;
- *Questionnaires* ("Questionários") is displayed on the left lower side. In this section the physician can access to questionnaires which are answered by the patient;
- *Medical examinations* ("Exames Auxiliares de Diagnóstico") is displayed on the middle lower side. This section displays the most recent results of several medical examinations and laboratory tests. The physician can access the patient's examination and test histories, and can also add new results;
- *Notes* ("Observações") is displayed on the right lower side. This section enables to write a textual note regarding the patient's condition on the day of the medical visit and to access notes from previous appointments.

For the insertion of data, the user interface provides several forms. An example of a form is shown in Figure 2.7. The form presented enables to edit part of a patient's personal data. The left side of the form is composed of several labels followed by the respective text box below. Specifically, on the left side it is possible to enter the patient's name, birth date, telephone, email, profession, and weight in kilograms. Besides that, at the bottom the "Cancelar" button allows the user to return to the previous screen without making any changes to the patient's personal data. The right side of the form is also composed of several labels but mainly followed by the respective radio buttons below. Specifically, on the right side of the form it is possible to select the radio button regarding the status of the patient's alcoholism, smoking, and drug problems. Additionally, closely to the bottom is also displayed a label ("Altura cm") and a text box that allows the user to input the patient's height in centimeters. Besides that, at the bottom is the


umedicine.herokuapp.com

Umedicine | José Manuel dos Santos Silva | Logout admin

Informação Pessoal

Número Processo: 1
Nome: José Manuel dos Santos Silva
Idade: 35
IMC: Peso normal (24.7)

Ver Mais | Editar

Questionários

Nome	Último	Penúltimo	Variação
IIEF	50	47	🔴
IPSS	29	25	🔴
SSES	11	29	🟢

Fazer Questionário

Doença

HBP

Sintomas

Dor hipogástrica	Incontinência de imperiosidade
Imperiosidade miccional	Atraso no inicio da micção

Mais Activos | Inactivos | Adicionar

Toque Rectal

Nódulos: sim	Volume da próstata (cm3): 121.0
Superfície: irregular	Consistência: elástica

Ver Mais | Historico | Adicionar

Exames Auxiliares de Diagnóstico

Análises (PSA total: 1234.0 ng/mL)
Urofluxometria (Flow time: 12.0 seg, Qmax: 23.0 mL/seg)
Ecografia Renal
Ecografia Vesical (Volume da próstata: 123.0 cm3)
Ecografia Prostática (Nódulos: sim, Volume da próstata: 13.0 cm3)

Histórico | Adicionar

Tratamento (2016-05-25)

Tratamento Cirúrgico

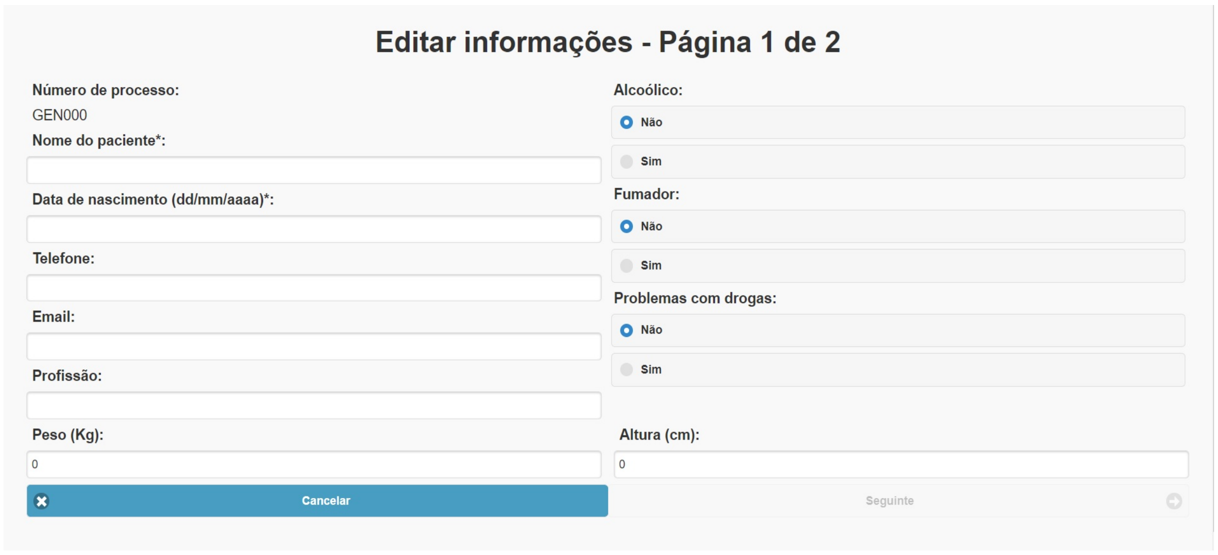
Operatório	Tipo Cirurgia: Adenomectomia (aberta)
	Data cirurgia: 2016-05-25
	Tempo cirúrgico (seg): 123
	Perda sanguínea (mL): 123
	Transfusões: 123
Pós-operatório	Dias de algaliação: 34
	Dias de internamento: 34
	Retenção coágulo: 34

Histórico | Pós-operatório | Adicionar

Observações (2016-05-29)

Histórico | Guardar

Figure 2.6: Patient's information screen shown to a physician in Umedicine. Figure from Lages *et al.* [9].


Editar informações - Página 1 de 2

Número de processo: GEN000	Alcoólico: [✓ Não] Sim
Nome do paciente*:	Fumador: [✓ Não] Sim
Data de nascimento (dd/mm/aaaa)*:	Problemas com drogas: [✓ Não] Sim
Telefone:	
Email:	
Profissão:	
Peso (Kg): 0	Altura (cm): 0

Cancelar | Seguinte

Figure 2.7: Form to edit part of a patient's personal data in Umedicine. Figure from Umedicine's user interface.

"Seguinte" button that allows the user to proceed to the next form, which enables to edit the remaining patient's personal data. The "Seguinte" button is displayed as disabled and it is only enabled when the mandatory form fields (in this case, the patient's name and his/hers data of birth) are filled.

The storage of structured data in a database enables the execution of data analysis algorithms. The current version of Umedicine supports the application of a biclustering algorithm whose goal is to find groups of patients that have a similar International Prostate Symptom Score (IPSS) over time.

The IPSS is a score calculated from a questionnaire that is filled by Urology patients that suffer from Benign Prostatic Hyperplasia (BPH), for instance [9]. A higher score represents a worse condition of the patient. This questionnaire is filled several times by BPH patients during their treatment, generating several IPSSs that are used as data in the biclustering algorithm. In the Umedicine user interface, the output of this algorithm provides several clusters of patients that the user can explore through an interactive visualization. The Figure 2.8 illustrates the visualization that results from the application of this algorithm.

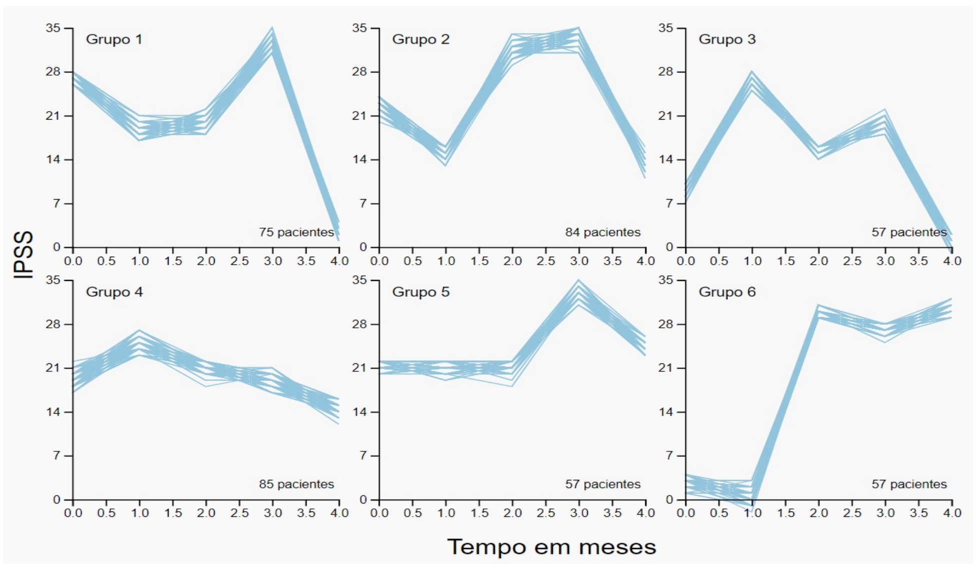


Figure 2.8: Visualization resulting from the application of the biclustering algorithm. Figure from Umedicine's user interface.

The visualization is composed of a matrix of line charts, where each chart corresponds to a cluster of patients [9]. In each chart is represented the evolution of IPSS over time (in months) of a particular cluster of patients. On the right lower side of each chart is mention the number of patients that compose the respective cluster. The user can click on each chart to obtain more information regarding the respective cluster. In concrete, it is displayed the most common characteristics ("Características mais comuns") of the patients belonging to that cluster and the most common treatment among them ("Tratamento mais comum") (see Figure 2.9). In addition, even for more information, the user can click on the "Ver mais" button, which will display a table that shows several characteristics and a short information for each one (mainly a percentage). For example, 54% (information) of the patients in that cluster consume drugs (characteristic).

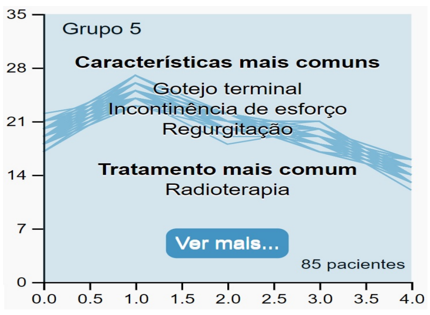


Figure 2.9: Example of the common information displayed regarding a cluster of patients when clicking on a chart. Figure from Umedicine's user interface.

2.3.2 PRECISE Stroke

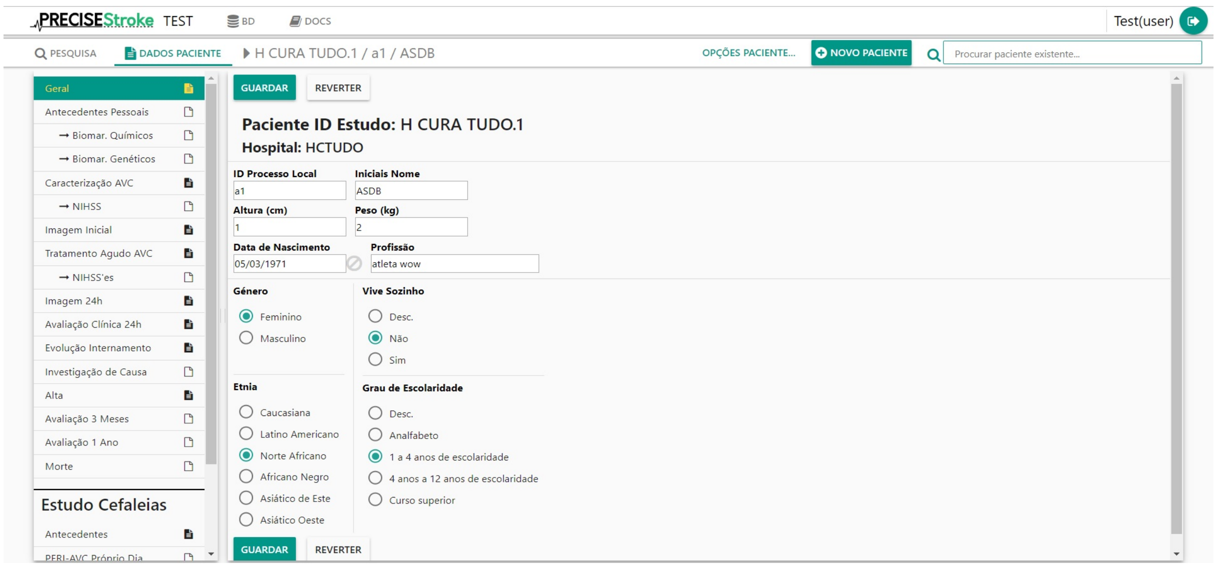
PRECISE Stroke[33] is a clinical data registry for Neurology, in particular to maintain data about stroke patients. Currently, PRECISE Stroke is used in the hospitals that are collaborating in the project in which PRECISE Stroke was developed: the PAC-PRECISE project. This project aims at accelerating the progress of the Portuguese healthcare for the new era of precision medicine. Within PAC-PRECISE, the main aim of PRECISE Stroke is to discover the stroke biomarkers (clinical, imaging, biochemistry, genetic) that are able to determine a better prognosis of stroke in order to prevent it afterwards. To achieve this goal, PRECISE Stroke must provide the means to: (i) collect and gather all stroke data (clinical, imaging, blood samples) from several hospitals in a central database; (ii) store, share and analyse the collected data.

In generic terms, PRECISE Stroke is composed of a database and a user interface (as Umedicine described in 2.3.1). The database is a key-value store that stores data regarding the patient's characteristics and historial data, symptoms, medical examinations, treatments, patient's follow-up and questionaries. The access to the user interface is protected by a username and a password. The user interface is only accessible to the physicians of the hospitals that are collaborating with PRECISE. This interface consists of several forms that are composed mostly of closed response fields (i.e., fields where it is only possible to select the options that are shown). In particular, most of the fields are radio buttons, thus enabling the data collected and stored to be structured. There are also other fields that undergo a validation process, where, for instance, only numbers are accepted.

A test version of the user interface is available online[34]. One of the screens of the test version's user interface is shown in Figure 2.10. In particular, this figure shows the screen that allows the user to navigate through the data regarding a particular patient. In more detail, it shows the screen that enables the user to visualize and edit the patient's personal characteristics.

The top side of the screen displays two tool bars. Both tool bars are mainly composed of several

[33] https://stroke.precisemed.org/home/
[34] http://stroketest.sysresearch.org/login/



Paciente ID Estudo: H CURA TUDO.1
Hospital: HCTUDO

ID Processo Local	a1
Iniciais Nome	ASDB
Altura (cm)	1
Peso (kg)	2
Data de Nascimento	05/03/1971
Profissão	atleta wow

Género: [✓ Feminino] Masculino

Vive Sozinho: Desc. [✓ Não] Sim

Etnia: Caucasiana, Latino Americano, [✓ Norte Africano], Africano Negro, Asiático de Este, Asiático Oeste

Grau de Escolaridade: Desc., Analfabeto, [✓ 1 a 4 anos de escolaridade], 4 anos a 12 anos de escolaridade, Curso superior

Figure 2.10: An example of a screen of the PRECISE Stroke test version's user interface. This screen allows the user to visualize and edit the patient's personal characteristics. Figure from PRECISE Stroke test version's user interface.

buttons that redirect the user to other screens in which the user can:

- View the list of patients ("BD" button)
- Search patients through filters ("PESQUISA" button)
- Access to PDF documents ("DOCS" button)
- Logout (button next to the "Test(user)" label)
- Create a new patient ("NOVO PACIENTE" button).

The "OPÇÕES PACIENTE" button gives access to other buttons that enables to extract the current patient's data and delete the patient. Finally, the text box displayed on the right side of the screen allows the user to search for patients.

Similarly to the visit screen of Reuma.pt (described in Section 2.1.1), the left side of the screen illustrated in Figure 2.10 displays a menu list that, when clicked, opens the respective screen on the right side. This menu list incorporates several menus regarding the topics of the data that is collected, whose topics were already mentioned above.

Finally, the right side of the screen illustrated in Figure 2.10 displays the personal characteristics of one patient. Specifically, it shows the patient's process id, the initials letters of his/her name, height and weight, birth date and profession in text boxes. Besides that, according to the selected radio button, it shows the gender, ethnicity, level of education, and if he/she lives alone or not. All the personal characteristics displayed can be edited and then saved when pressing the "GUARDAR" button that is on the top left side. The personal characteristics that were edited can be reverted to the original data when pressing the “REVERTER” button that is also on the top left side.

2.3.3 Discussion

In this section we described two clinical data registries developed in the context of research projects: Umedicine and PRECISE Stroke. The information about these clinical data registries is summarized in Table 2.3. This table describes, for each clinical data registry, its medical specialty, monitoring purposes,

Table 2.3: Summary of the research clinical data registries

Clinical data registry / Property	Umedicine	PRECISE Stroke
Medical specialty	Urology	Neurology (stroke)
Monitoring purposes	- Treatment - Diagnosis - Disease	- Prognosis
Data collected	- Patient’s characteristics - Patient’s history - Examinations - Treatment - Patient’s follow-up - Symptoms - Questionnaires	- Patient’s characteristics - Patient’s history - Examinations - Treatment - Patient’s follow-up - Symptoms - Questionnaires
Data analysis	Explore groups of similar patients	None
Structured data	Yes	Yes
Accessible to	- Physicians - Patients - Clerks	- Physicians

the data collected, data analysis aspects, whether the data is structured, and by whom it is accessible.

Umedicine covers the medical specialty of Urology and PRECISE Stroke covers the medical specialty of Neurology (for stroke). Umedicine has monitoring purposes regarding treatment, diagnosis and disease, and PRECISE Stroke just has in relation to prognosis. Both these clinical data registries collect data common to various topics, in particular: patient’s characteristics, patient’s history, medical examinations, treatments, patient’s follow-up, symptoms and questionnaires. In addition, in both clinical data registries, data stored is structured, being composed largely by enumerated fields and by very few free text fields. Specifically in the case of Umedicine, we know that the data is even stored in a relational database and, in PRECISE Stroke, it is stored in a key-value database. Contrarily to PRECISE Stroke, Umedicine provides data analysis mechanisms that enable to explore groups of similar patients, for instance. Finally, Umedicine can be accessed by several types of users (patients included), while PRECISE Stroke can only be accessed by physicians.

2.4 Low-Code Development Platforms (LCDPs)

LCDPs are software platforms composed of tools that allow developers to create applications efficiently and with a minimum amount of code that needs to be written by them. Although programmers can usually write code when creating applications using these platforms, LCDPs are built to be used by people that have little programming knowledge. In fact, business-related people can use these platforms to create applications in order to optimize a business process, for instance.

Besides the fact that LCDPs can be used by a generic person (i.e., that do not need to have explicit programming skills), these platforms are designed in a way that enables to accelerate the software delivery process. This aspect can be useful for software developers that usually have to develop a lot of applications in a short period of time. Therefore, instead of developing an application from scratch, which use to be the standard way, the developers can use the LCDPs to help them.

In order to develop applications, the LCDPs provide an environment that usually has common characteristics that differ from the environment that is provided in the traditional way of programming. These platforms typically provide a high-level environment that is composed of graphical user interfaces that can be used and configurations that only need to be specified. This way, the LCDPs reduce the amount of code that needs to be written by the people that uses them. In some cases, there are platforms that are capable of producing applications without the concerned user even having to write a single line of code. These platforms are called no-code development platforms.

No-code development platforms do not allow users to write code when developing applications. However, this characteristic has the disadvantage that users are limited to what is only possible to do through the platform when developing an application. That is, users are not allowed to extend their application beyond the functionalities that the platform provides. On the other hand, LCDPs allow users to extend their applications by enabling writing code.

OutSystems[35], Mendix[36], Appian[37] and Nintex[38] are examples of LCDPs. OutSystems and Mendix are LCDPs for generic purposes and are similar in terms of their user interface usability and the functionalities that they provide. On the other hand, Appian and Nintex are LCDPs focused on applications that run business processes. In concrete, Apian and Nintex target applications that demand coordination and collaboration between various employees.

In Section 2.4.1, we describe OutSystems in more detail and give some examples of how applications are developed using this LCDP. Finally, in Section 2.4.2, we briefly discuss the possibility of using LCDPs for generating clinical data registries.

[35]https://www.outsystems.com/
[36]https://www.mendix.com
[37]https://www.appian.com
[38]https://www.nintex.com

2.4.1 OutSystems

OutSystems is a LCDP that enables to develop an application in a visually manner, integrate it with existing systems and customize it with our own code. OutSystems was created in 2001, and currently more than 100.000 applications were developed using the OutSystems platform. This platform is defined as a platform as a service (PaaS) and, basically, is the product that OutSystems provides. OutSystems platform is used for developing and delivering enterprise web and mobile applications, which can run on-premises (i.e., applications that are stored and executed locally), in the cloud or even in a hybrid environment. The platform has two versions: the free and the paid. In the free version there is only the development environment, the application's capacity is limited, the end-user's capacity can go up to 100 and it is the community that provides the support. On the other hand, in the paid version, there are at least three environments (to develop, test and produce), the application capacity scales according to its needs, the end-user's capacity can be more than 100 and it is OutSystems that provides the support.

The fact that the OutSystems platform is based on low code means that the developer needs little programming knowledge to be able to develop an application (i.e., the definition of a low-code platform, as we stated above). In fact, the applications are built and changed in a visual environment where the developer is able to define the application's data model, its business logic, its workflow processes and its user interfaces both for the web and mobile devices. In the platform, the development is performed in a drag-and-drop fashion way and the developer can make customizations through code in C#, HTML, SQL, CSS and JavaScript. After the development of an application, the developer only needs to press a button to deploy it. Then, the testers and end-users can use the application and provide feedback on it. As we stated, the applications can run on-premises, in the cloud or in a hybrid environment. The applications are also portable, which enables to change the environment where it is running easily. In addition, the applications developed using OutSystems platform can also be integrated with existing applications that may have been developed without OutSystems.

Figure 2.11 shows the screen of OutSystems platform where the user can create the data model of an application. Typically, this screen consists of a canvas on the left side where the user can drag-and-drop the tables that compose the database of the application that are listed in the menu displayed on the right side. With this, the user can create relationships between the tables as shown in the Figure 2.11. In particular, it is shown a data model that is composed of two tables (`Patient` and `Radiotherephy`) and they have a relationship between them.

Figure 2.12 shows the screen of OutSystems platform where the user can define the application logic. The application logic is defined through actions. The user can create new actions and assign them to buttons that exist in screens of the user interface. In overall, as shown in Figure 2.12, the screen that allows the user to define new actions is composed of a tool bar that is displayed on the left side, a canvas on the middle and a menu that lists the existing actions on the right side. The user can drag

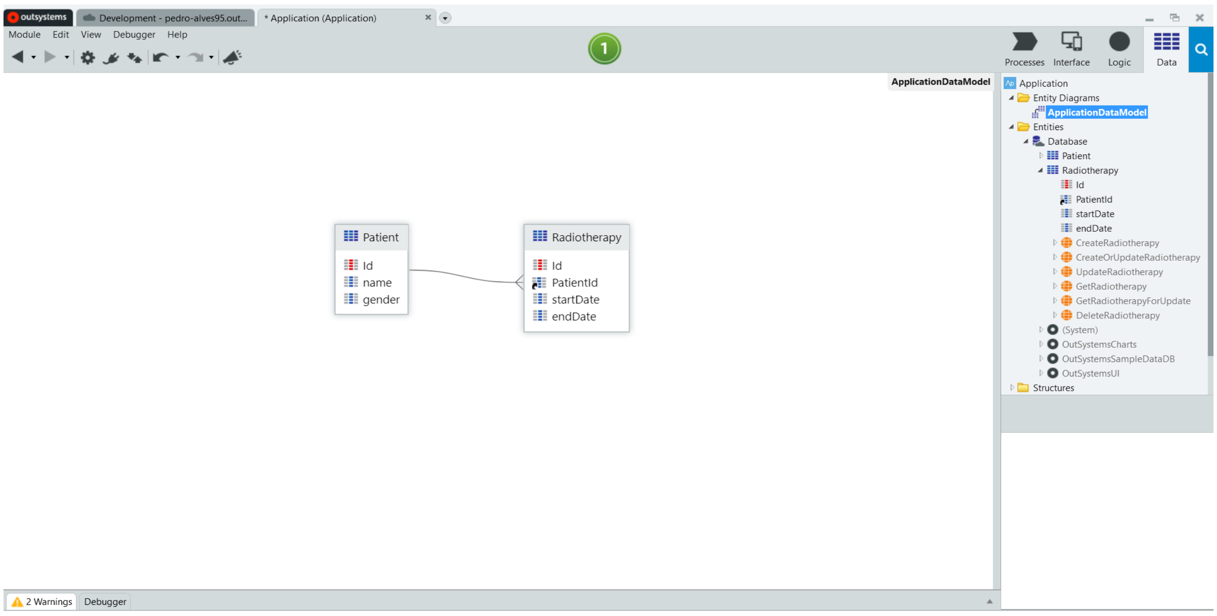
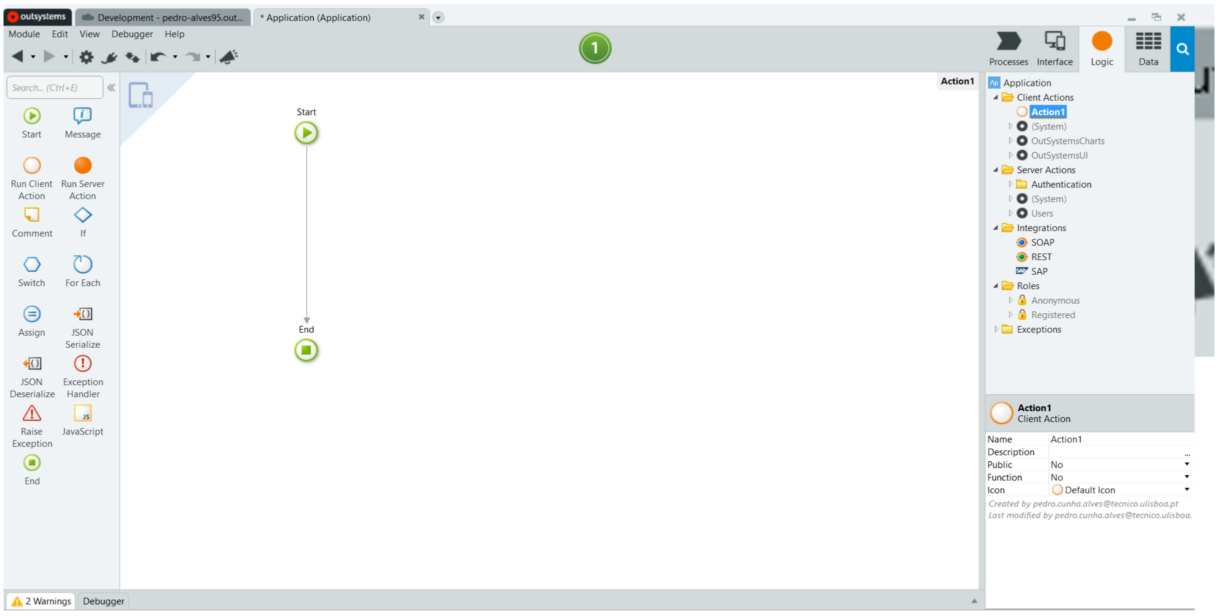


Figure 2.11: Screen of OutSystems platform that allows the user to create the data model of an application.

items from the tool bar and drop in the canvas, as happened in the screen that enables the user to create the data model of an application.

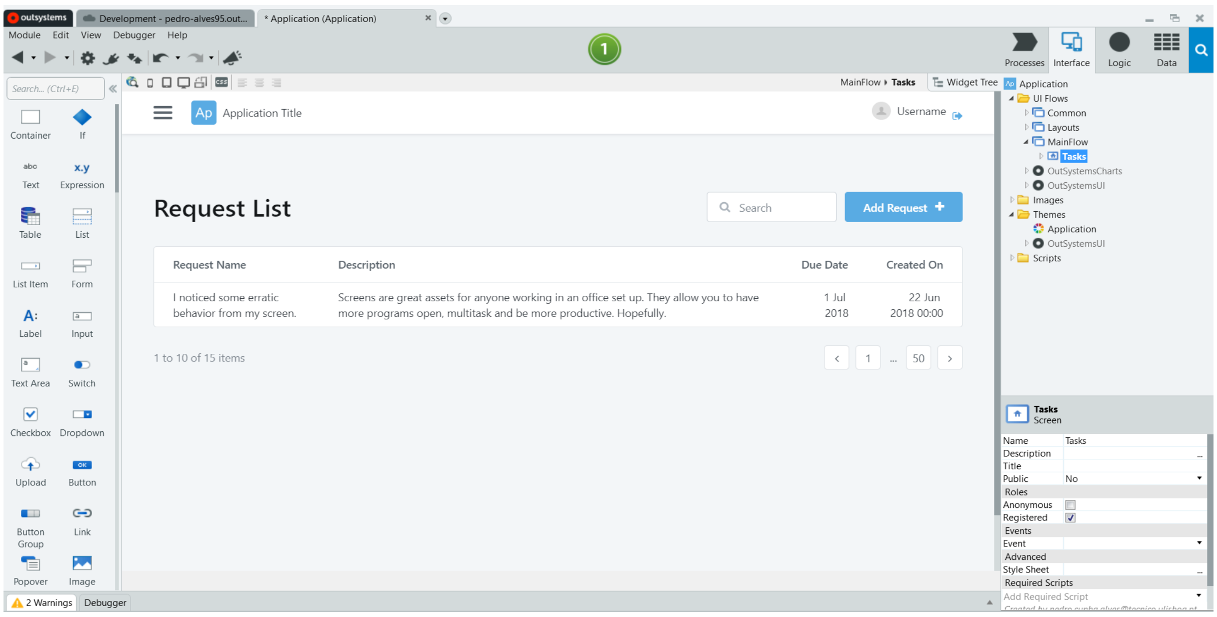


Name	Action1
Description	
Public	No
Function	No
Icon	Default Icon

Created by pedro.cunha.alves@tecnico.ulisboa.pt
Last modified by pedro.cunha.alves@tecnico.ulisboa.

Figure 2.12: Screen of OutSystems platform where the user can define the application logic.

Figure 2.13 shows the screen of OutSystems platform that allows the user to create and customize user interface screens of an application. Regarding aspect and interaction, this screen follow the same reasoning of the screen shown in Figure 2.12. In concrete, this screen is composed of a tool bar on the left side, a canvas on the middle and a menu that lists the existing user interface screens on the right side.



Request Name	Description	Due Date	Created On
I noticed some erratic behavior from my screen.	Screens are great assets for anyone working in an office set up. They allow you to have more programs open, multitask and be more productive. Hopefully.	1 Jul 2018	22 Jun 2018 00:00

1 to 10 of 15 items ‹ 1 ... 50 ›

Name	Tasks
Description	
Title	
Public	No
Roles	
Anonymous	☐
Registered	✓
Events	
Event	
Advanced	
Style Sheet	
Required Scripts	
Add Required Script	

Figure 2.13: Screen of OutSystems platform that allows the user to create and customize user interface screens of an application.

2.4.2 Discussion

In this section we addressed LCDPs in general and detailed one of them: OutSystems. The fact that LCDPs allow developers to create applications more efficiently thus provides a lower cost compared to the standard process of creating applications. Still, suppose we wanted to develop several applications that provide the same functionality but whose data to store is different. We would have two options to do this when using LCDPs: (i) always create a new application and develop everything from scratch; or (ii) reuse the first application developed and change certain aspects in order to develop the other applications. If we always had to create a new application and develop everything from scratch, we would be wasting resources since we would be repeating much work due to the common aspects of the applications. On the other hand, if we reused the first application developed and changed certain aspects, we would not only have to change the application database (which was expected), but also its user interface in order to accommodate the different data. This way, we always had to spend time (and therefore resources) making these changes when we wanted to provide the same functionality but only different data. This would be a problem that fits into the creation of new clinical data registries, since they usually provide common functionalities (such as adding and navigating through the data) and collect different specific data.

3
Requirements Analysis

Contents

3.1 Data to Store 46
3.2 Functionalities 52
3.3 User Interface 57
3.4 Discussion 58

In Section 1.1 we identified the problem that we want to solve: the waste of resources in terms of software design and development when creating a new clinical data registry. In fact, part of the tasks required to create this kind of software application are repeated since the clinical data registries have common characteristics, namely: (i) the user interface of clinical data registries supports common **functionalities**; and (ii) the **data** that is collected by the clinical data registries covers common topics.

In this chapter we present the requirements analysis that we collected in order to decide what we will support in any clinical data registry to be generated. The requirements gathering is performed with respect to three topics: (i) data to store (Section 3.1); (ii) functionalities (Section 3.2); and (iii) user interface (Section 3.3). In Section 3.4, we conclude the requirements analysis, mentioning what we will support in the clinical data registry to be generated with respect to each of the three topics.

The requirements analysis is performed based on the clinical data registries that currently exist in Portugal, that we described in Chapter 2. Since we do not have full access to all these clinical data registries, we perform the requirements analysis based on those that we have information on at least one of the topics. In concrete, we use the following clinical data registries for each topic:

i. Data to store:

    (a) RON (described in Section 2.1.3)

    (b) RNMAVD (described in Section 2.2.2)

    (c) PRNC (described in Section 2.2.5)

    (d) Umedicine (described in Section 2.3.1).

ii. Functionalities:

    (a) Reuma.pt (described in Section 2.1.1)

    (b) RON (described in Section 2.1.3)

    (c) RNMAVD (described in Section 2.2.2)

    (d) PRNC (described in Section 2.2.5)

    (e) Umedicine (described in Section 2.3.1).

iii. User interface:

    (a) Reuma.pt (described in Section 2.1.1)

    (b) Umedicine (described in Section 2.3.1).

We do not perform the requirements analysis for PRECISE Stroke (described in Section 2.3.2) because it is the clinical data registry that we use to validate CDRGen.

3.1 Data to Store

In terms of the data to store, we first identified the common entity groups with respect to the data collected by the 12 clinical data registries described in Chapter 2. An *entity group* is identified by a name and is composed of a set of entities. An *entity* is also identified by a name and is composed of a set of attributes. An *attribute* is identified by a name and has a value. The values of all attributes represent the data that is collected by a clinical data registry. Figure 3.1 represents the UML class diagram of these three concepts.

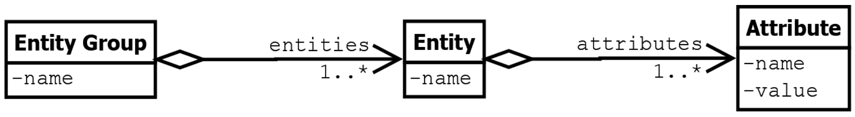


Figure 3.1: UML class diagram of the entity group, entity and attribute concepts.

We identified the following entity groups:

- `Person`, representing the patient, namely her characteristics
- `Diagnosis`, representing the symptoms and diseases
- `Treatment`, representing the various treatments that a patient can perform
- `Questionnaire`, representing the questionnaires that a patient fills in
- `Medical examination`, representing medical examinations, which can include laboratory tests
- `Medical appointment`, representing the patient's follow-up in medical visits.

With these entity groups identified, we focus on four clinical data registries that are the ones for which we have access to a large quantity of information about the data collected. In particular, we identify the entities of the data collected in each of the clinical data registries and we assign each entity to one of the entity groups identified. For example, the entity that represents the patient is assigned to the `person` entity group. We also identify the relationships between entities. For instance, there usually is a many-to-many relationship between symptom and disease since one disease may present several symptoms and one symptom may manifest itself in several diseases. Lastly, we identify the attribute types (e.g., string, integer) and constraints that must be satisfied by attributes. We start by detailing each of these aspects for the PRNC clinical data registry, followed by RNMAVD, Reuma.pt and RON.

PRNC collects data through Case Report Forms (CRFs)[1]. Each CRF contains all the data about a patient. The list of entities with respect to the data collected by PRNC is represented in Table 3.1. This list is based on the PDF file that describes all the data that is collected by PRNC[2]. In this table, each

[1] In Annex A we present an example of a CRF.

[2] https://spc.pt/pdf/cncdc/registos/mnc/LISTA%20VARIAVEIS%20DE%20MNC.pdf

entity is associated to an entity group. For example, the `holter` entity is associated to the `medical examination` entity group since holter is a medical examination. Regarding the relationships between

Table 3.1: List of entities with respect to the data collected by PRNC and the corresponding entity groups.

Entity group	Entity
Person	Patient Comorbidities Family History
Diagnosis	Diagnosis
Treatment	Treatment Complications
Medical examination	Echocardiography Cardiac MRI Electrocardiogram Holter Genetic study
Medical appointment	Follow-up

entities, since all patient's data is contained in a single CRF, all entities need to have a relationship with the `patient` entity in order to identify who the data belongs to. Apart from this relationship with the `patient` identity, as far as we know, it does not seem necessary to have any other relationship between entities. Figure 3.2 represents the relationships between the entities along with the corresponding multiplicities[3].

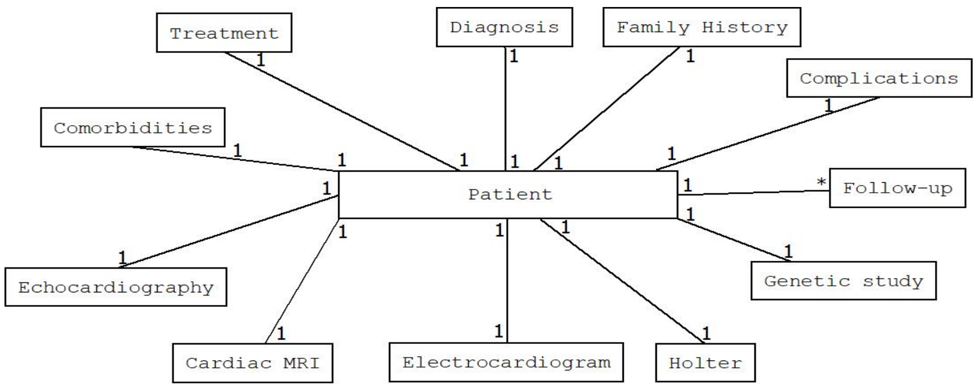


Figure 3.2: Relationships between entities in PRNC.

Concerning the attribute types, we identified attributes of the following types: string, enumerated, list (of one of the previous types), numeric, date and boolean. In the particular case of an attribute of the enumerated type, there are cases where it is possible to assign another value than the predefined ones. For example, suppose that we have the enumerated-type attribute `ethnicity` whose values are "Caucasian", "African", "Asian" and "other". The "other" value can be "Indian" (for instance) or any other

[3]Although in reality makes sense to have a one-to-many relationship between various pairs of the entities that are displayed (for example, the `patient` entity and the `electrocardiogram` entity), the multiplicities shown in the figure are related to the scope of PRNC.

ethnicity that was not predefined.

Finally, we identified the following constraints between attributes:

- The value of an attribute can determine the assignment of values to other attributes. For example, if a patient does not have any symptoms, it does not make sense to describe a symptom (i.e., give values to other attributes that characterize a symptom). In concrete, let us suppose that we have two attributes: (i) the `has symptom` attribute indicates if a patient has symptoms; (ii) and the `symptom name` attribute indicates the name of the symptom. If the value of the `has symptom` attribute is "no", then it does not make sense to give a value to the `symptom name` attribute.

- The value of an attribute must be one of the values that were already given to another attribute (as it happens with a foreign key in the relational model). For example, a requirement may be to store the identifier (that identifies a patient in a clinical data registry) of the patient's father. Suppose that we have only three patients inserted in a clinical data registry whose identifiers are "id1", "id2" and "id3". We can only assign one of these three identifiers to the value of the attribute that represents the patient's father identifier. Otherwise, we would be entering an invalid value since we would be entering an identifier of a patient that did not even exist in the clinical data registry.

RNMAVD collects data through a Case Report Form (CRF), as in PRNC. The list of entities with respect to the data collected by RNMAVD is represented in Table 3.2. This list is based on the PDF file that describes all the data that is collected by RNMAVD[4]. Again, in this table, each entity is associated to an entity group. Regarding the relationships between entities, the same happens as in PRNC, where all entities need to have a relationship with the `patient` entity. Apart from this relationship with the `patient` identity, as far as we know, it does not seem necessary to have any other relationship between entities. Figure 3.3 represents the relationships between the entities along with the corresponding multiplicities[5]. Concerning the attribute types, we identified attributes of the following types: string, enumerated, list (of one of the previous types), numeric, date and boolean. In the particular case of an attribute of the enumerated type, there are cases where it is possible to assign another value than the predefined ones. Finally, we found the same two constraints between attributes as in PRNC: (i) the value of an attribute can determine the assignment of values to other attributes; and (ii) the value of an attribute must be one of the values that were already given to another attribute.

[4]https://spc.pt/pdf/cncdc/registos/mavd/LISTA%20VARIAVEIS%20MAVD%20revisto.pdf

[5]Although in reality makes sense to have a one-to-many relationship between various pairs of the entities that are displayed (for example, the `patient` entity and the `electrocardiogram` entity), the multiplicities shown in the figure are related to the scope of RNMAVD.

Table 3.2: List of entities with respect to the data collected by RNMAVD and the corresponding entity groups.

Entity group	Entity
Person	Patient Comorbidities Family history
Diagnosis	Diagnosis Diagnostic criteria
Treatment	Medical therapy Pacemaker CDI Heart transplant
Medical examination	Echocardiography Cardiac MRI Electrocardiogram Holter Cardiac stress test Genetic study Histological characterization of the wall
Medical appointment	Follow-up

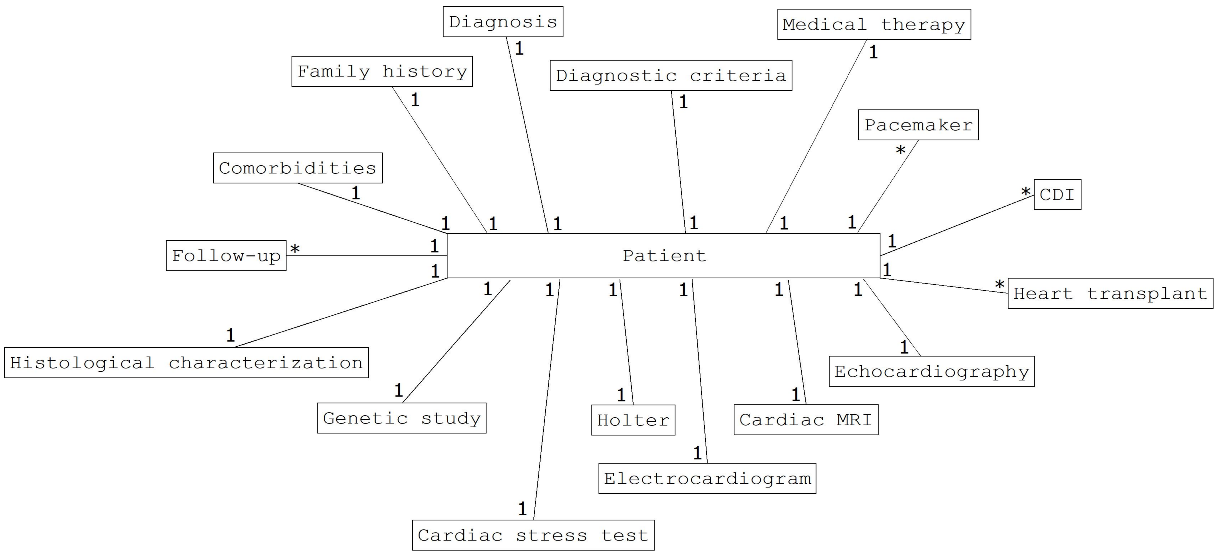


Figure 3.3: Relationships between entities in RNMAVD.

Umedicine's list of entities with respect to the data collected is represented in Table 3.3. This list is based on the relational schema of the Umedicine database. In this table, each entity is associated to an entity group. Regarding the relationships between entities, we identified relationships between the `patient` entity and the other entities. In addition, we identified relationships between the following entities:

- `Symptom` and `disease`, since one disease may present several symptoms and one symptom may manifest itself in several diseases (many-to-many relationship)

Table 3.3: List of entities with respect to the data collected by Umedicine and the corresponding entity groups.

Entity group	**Entity**
Person	Patient Personal history Medical information
Diagnosis	Disease Symptom
Treatment	Treatment Hormonal block Surgery Radical prostatectomy Intraoperative complication Early postoperative complication Medicine Medication Prosthesis Radiotherapy
Medical examination	Medical examination
Medical appointment	Observations Questionnaire

- `Treatment` and `medical examination`, because, for example, in the case of a radiotherapy treatment it may be necessary to measure the Prostate-Specific Antigen (PSA)
- `Treatment` and `disease`, in which several diseases can be treated with one or more treatments and one treatment can treat one or more diseases (many-to-many relationship)
- `Medicine` and `disease`, because a medicine may be prescribed for several diseases and a disease can be treated with several medicines (many-to-many relationship)
- `Medicine` and `medication`, because several medicines can compose a single medication treatment and a medicine can appear in several medication treatments (many-to-many relationship).

Figure 3.4 represents the relationships between the entities along with the corresponding multiplicities. Concerning the attribute types, we identified attributes of the following types: string, numeric (integer and float), date, boolean, enumerated and image. Numeric and string attributes had digit/character limits (e.g., a string could be a maximum of 50 characters). Finally, we found the same two constraints between attributes as in the other two clinical data registries mentioned above: (i) the value of an attribute can determine the assignment of values to other attributes; and (ii) the value of an attribute must be one of the values that were already given to another attribute.

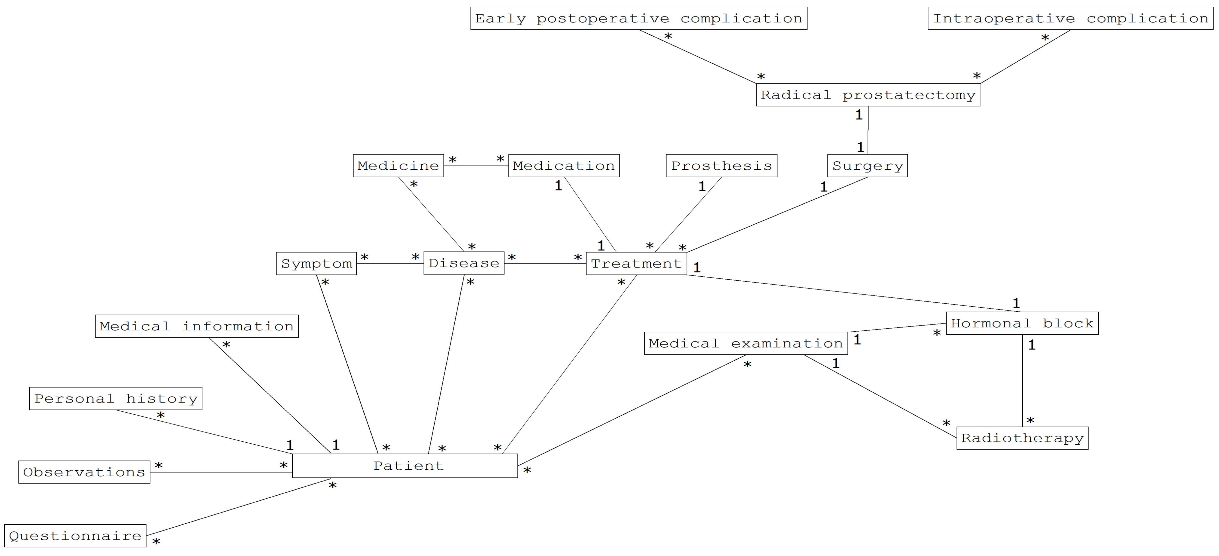


Figure 3.4: Relationships between entities in Umedicine.

RON's list of entities with respect to the data collected is represented in Table 3.4. This list was built based on the *Decreto Lei* n°53/2017 of July 14[6]. Once more, in this table, each entity is associated to an entity group, as in the other tables presented before. Regarding the relationships between entities,

Table 3.4: List of entities with respect to the data collected by RON and the corresponding entity groups.

Entity group	Entity
Person	Patient
Diagnosis	Diagnosis
Treatment	Surgery Radiotherapy Chemotherapy
Medical examination	Medical examination
Medical appointment	Annual follow-up Death

as in PRNC and RNMAVD, as far as we know, it seems that there are only relationships between the `patient` entity and each of the other entities. Figure 3.5 represents the relationships between the entities along with the corresponding multiplicities. Concerning the attribute types, we identified attributes of the following types: string, list (of one the previous type), numeric, date and enumerated. Finally, regarding the constraints between attributes, we only found the constraint stating that the value of an attribute can determine the assignment of values to other attributes (the first constraint mentioned above in PRNC).

[6]https://dre.pt/application/file/a/107688306/

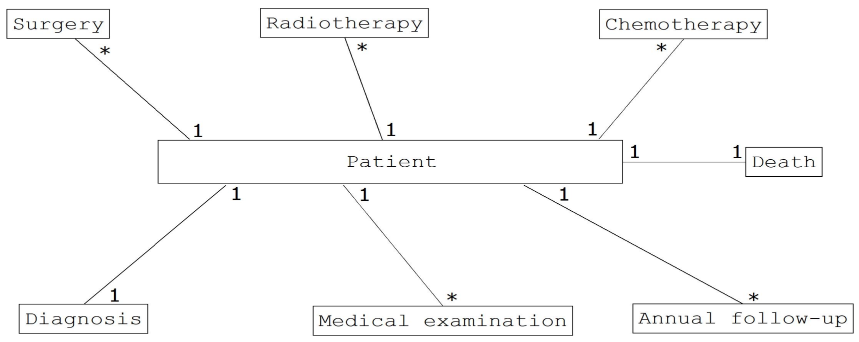


Figure 3.5: Relationships between entities in RON.

3.2 Functionalities

Regarding the functionalities that are supported by clinical data registries, we first describe the Umedicine functionalities, followed by the functionalities supported by Reuma.pt, RON, PRNC and RNMAVD.

Umedicine can be accessed by four types of users: (i) `patient`; (ii) `clerk`; (iii) `non-administrator physician`; and (iv) `administrator physician`. These users can perform different functionalities [10] according to the use case diagram represented in Figure 3.6.

i. The `patient` user can:

- Visualize and edit her personal data
- Add and visualize the results of her medical examinations
- Visualize the medicines that he/she takes regularly and the surgeries that he/she has done previously
- Add the questionnaire responses
- Edit and retrieve password.

ii. The `clerk` user can:

- Add `patient` users
- Add restricted patient personal data (name, birth date, phone number, email and profession)
- Edit and retrieve password.

iii. The `non-administrator` physician user can:

- Add `patient` users
- Add/visualize/edit the data of all patients

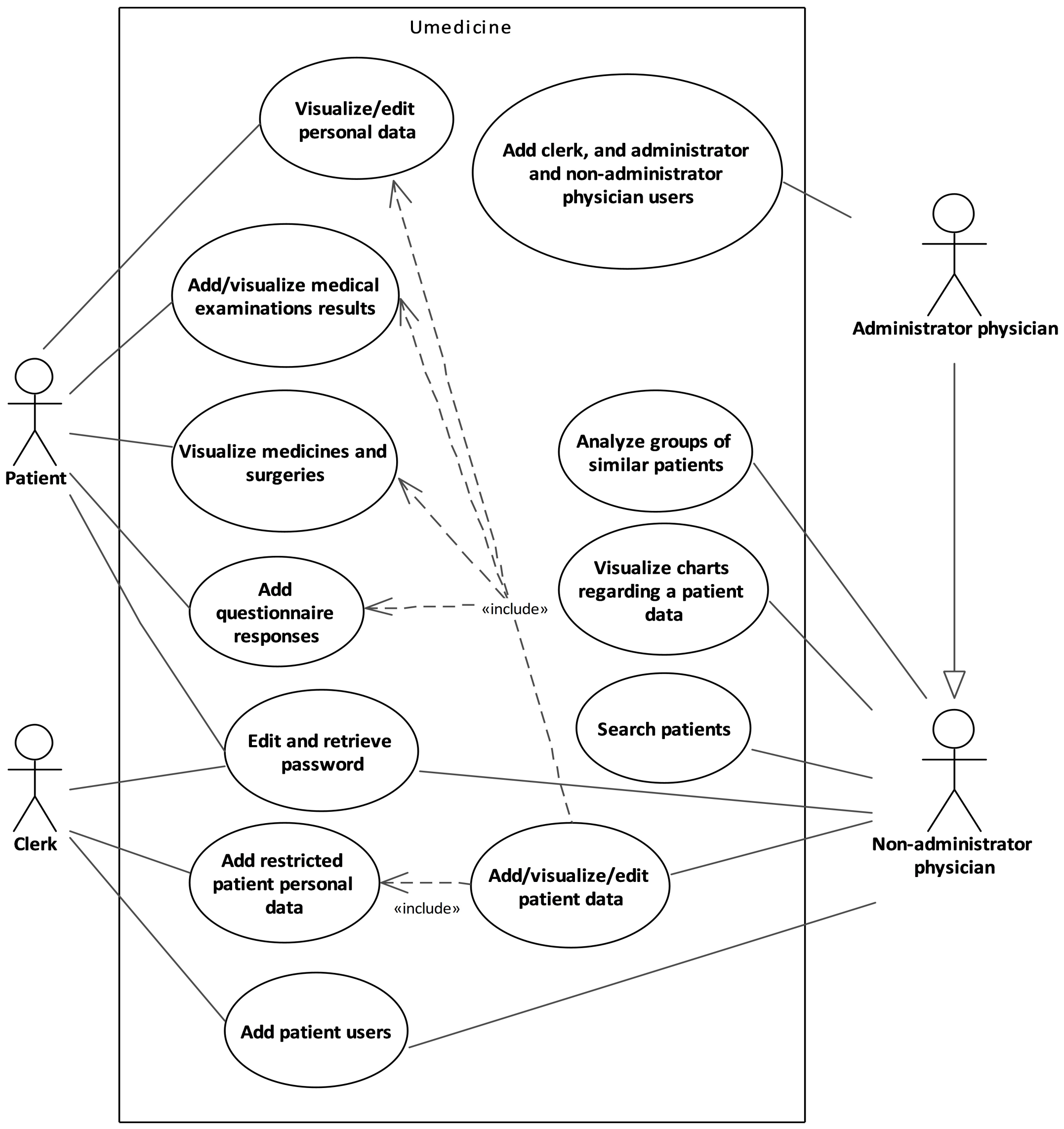

Umedicine
Visualize/edit personal data
Add clerk, and administrator and non-administrator physician users
Add/visualize medical examinations results
Visualize medicines and surgeries
Add questionnaire responses
Edit and retrieve password
Add restricted patient personal data
Add/visualize/edit patient data
Add patient users
Analyze groups of similar patients
Visualize charts regarding a patient data
Search patients
«include»
«include»
Patient
Clerk
Administrator physician
Non-administrator physician

Figure 3.6: Umedicine's use case diagram.

- Search for patients by name or by process number
- Visualize several charts regarding the data of a particular patient (e.g., evolution of prostate volume over time) in a dashboard
- Analyse groups of patients that have a similar IPSS[7]
- Edit and retrieve password.

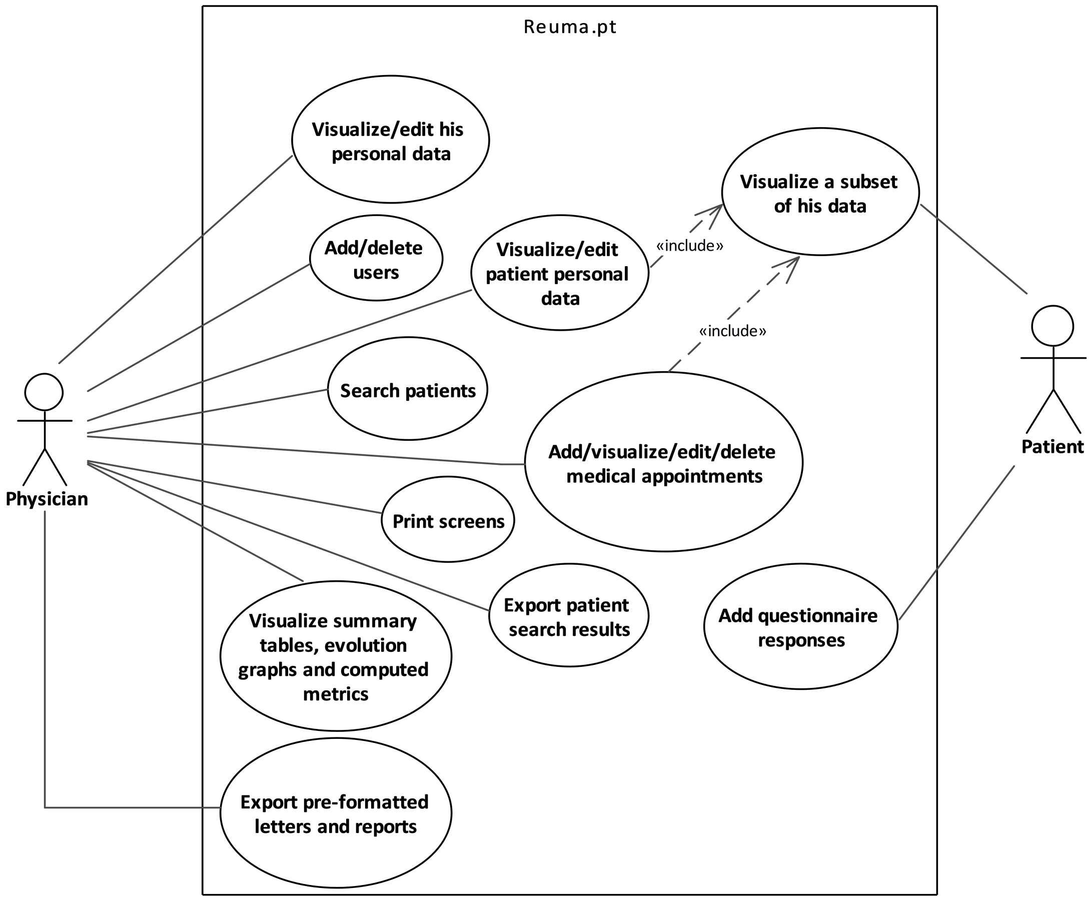
iv. The `administrator physician` user can:

- Perform all the functionalities of the `non-administrator physician` user

[7]The International Prostate Symptom Score (IPSS) is the metric used by physicians to measure the severity of Benign Prostatic Hyperplasia (BPH) symptoms

- Add clerk users, administrator and non-administrator physician users.

Reuma.pt (described in Section 2.1.1) can be accessed by two types of users: (i) physician; and (ii) patient. These users can perform different functionalities according to the use case diagram represented in Figure 3.7.



Reuma.pt
Visualize/edit his personal data
Add/delete users
Visualize/edit patient personal data
«include»
Visualize a subset of his data
«include»
Search patients
Add/visualize/edit/delete medical appointments
Print screens
Export patient search results
Add questionnaire responses
Visualize summary tables, evolution graphs and computed metrics
Export pre-formatted letters and reports
Physician
Patient

Figure 3.7: Reuma.pt's use case diagram.

i. The physician user can:

- Visualize and edit her personal data
- Add/delete physician and patient users
- Visualize and edit patient personal data
- Search for patients by protocol (i.e., the disease and its type of treatment), name, process number and many other filters
- Export the results of a patient search to a CSV file

- Add/visualize/edit/delete medical appointments
- Print screens
- Export pre-formatted letters for the family's doctor
- Export/print a pre-formatted report that contains all the entered data
- Visualize graphs of the evolution of a disease
- Visualize summary tables showing the patient's clinical situation
- Visualize automatic computed metrics from the entered data. For instance, DAS28 is a metric that describes the severity of Rheumatoid Arthritis (RA) and is computed using clinical and laboratory data.

ii. The `patient` user can:

- Visualize a subset of her data (both personal and medical)
- Add the questionnaire responses.

In addition to the functionalities described, each medical center that uses Reuma.pt can export statistical reports based on the data stored.

RON (described in Section 2.1.3) can be accessed by eight types of users: (i) `local recorder`[8]; (ii) `regional recorder`; (iii) `responsible for regional cancer data`; (iv) `responsible for regional oncological screening programs`; (v) `health manager`[9]; (vi) `RON's coordinator`; (vii) `pediatric co-ordinator`; and (viii) `administrator`. These user types were previously described in detail in Section 2.1.3. These user types have different permissions and the first seven types of users can perform different functionalities according to the use case diagram represented in Figure 3.8.

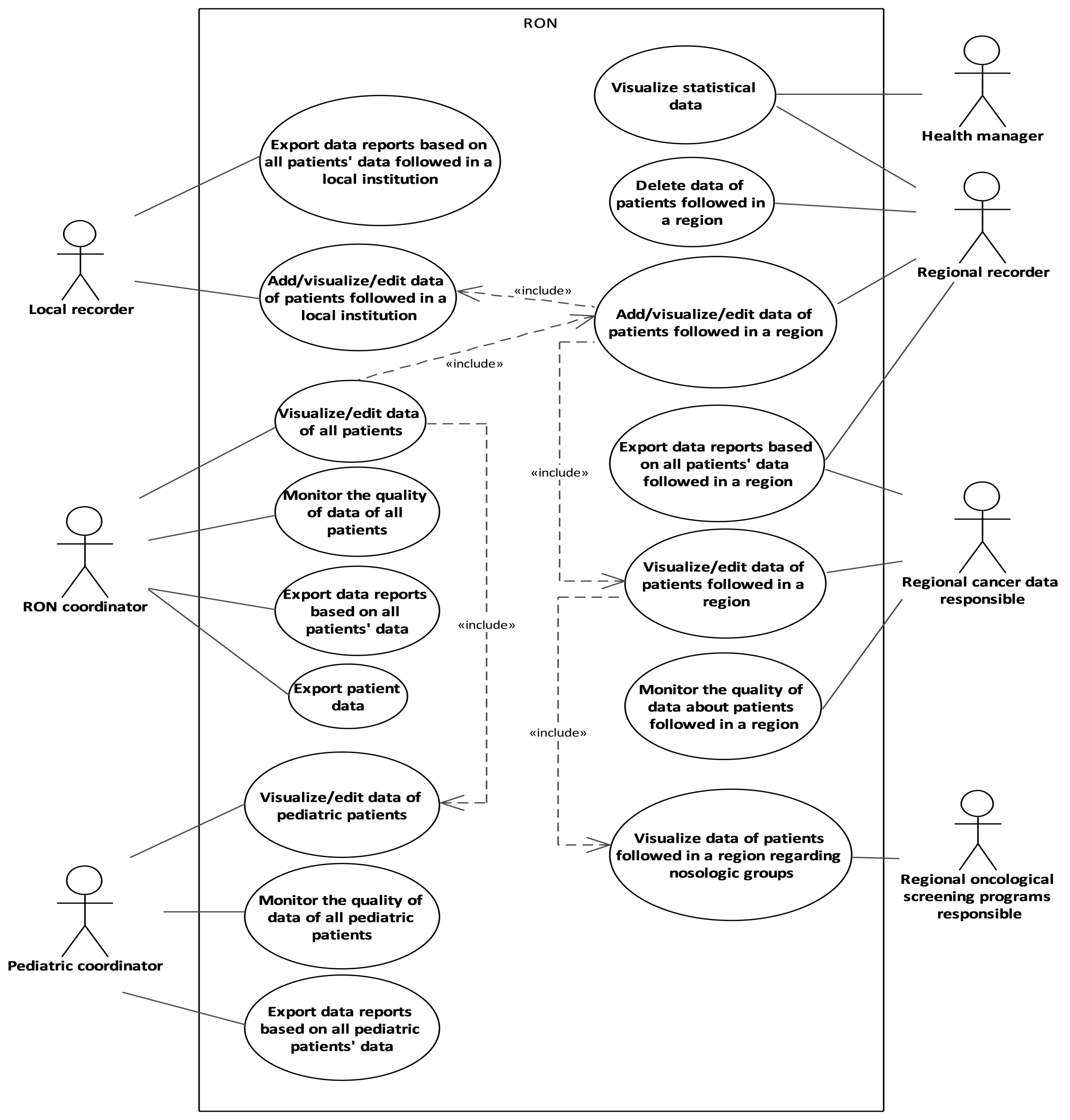
i. The `local recorder` user can:

- Add/visualize/edit data of patients that are followed in a given local institution
- Export unidentified aggregated data reports based on all the patients' data that are followed in a given local institution.

ii. The `regional recorder` user can:

- Add/visualize/edit data of patients that are followed in a given region
- Delete all the data of patients that are followed in a given region
- Export unidentified aggregated data reports based on all the patients' data that are followed in a given region

[8]A health worker of a given institution
[9]This is the regional health delegate or the director of the public health department of the regional health administrations



RON
Visualize statistical data
Health manager
Export data reports based on all patients' data followed in a local institution
Delete data of patients followed in a region
Regional recorder
Local recorder
Add/visualize/edit data of patients followed in a local institution
«include»
Add/visualize/edit data of patients followed in a region
«include»
Visualize/edit data of all patients
Export data reports based on all patients' data followed in a region
«include»
Monitor the quality of data of all patients
Regional cancer data responsible
RON coordinator
Visualize/edit data of patients followed in a region
Export data reports based on all patients' data
«include»
Export patient data
Monitor the quality of data about patients followed in a region
«include»
Visualize/edit data of pediatric patients
Visualize data of patients followed in a region regarding nosologic groups
Regional oncological screening programs responsible
Monitor the quality of data of all pediatric patients
Pediatric coordinator
Export data reports based on all pediatric patients' data

Figure 3.8: RON's use case diagram.

- Visualize statistical about the data stored.

iii. The user `responsible for the regional cancer data` can:

- Visualize and edit data of patients that are followed in a given region
- Monitor the quality of the data about patients that are followed in a given region
- Export unidentified aggregated data reports based on all the patients' data that are followed in a given region.

iv. The user `responsible for the regional oncological screening programs` can:

- Visualize data of patients that are followed in a given region regarding the nosologic groups (i.e., groups that are dedicated to the study and classification of diseases).

v. The `health manager` user can:

- Visualize statistical data.

vi. The `RON's coordinator` user can:

- Visualize and edit data of all patients
- Monitor the quality of the data of all patients
- Export unidentified aggregated data reports based on all the patients' data
- Export unidentified patient data.

vii. The `pediatric coordinator` user can:

- Visualize and edit data of all pediatric patients
- Monitor the quality of the data of all pediatric patients
- Export unidentified aggregated data reports based on all the pediatric patients' data.

RNMAVD and **PRNC** (described in Sections 2.2.2 and 2.2.5, respectively) can be accessed by two types of users: (i) the `principal researcher/physician`; and (ii) the `co-researcher` (also a physician). As far as we can understand, both types of users can add, visualize and edit all the patient data. However, only the `principal researcher` user can visualize the total number of records and the total number of records per clinical center where patients are followed. The corresponding use case diagram is represented in Figure 3.9.

3.3 User Interface

Among all the clinical data registries analysed, with respect to the user interface, we have access to the full Umedicine user interface and we only have access to a few screenshots of Reuma.pt's user interface. However, this is enough to identify which types of widgets are used in the user interfaces to enter data. We went no further than this, since with respect to other aspects of the user interface, we follow the only clinical data registry that we have full access: the Umedicine.

Table 3.5 lists the widgets that are used in each user interface and the corresponding types of attributes (i.e., data to collect). As we can observe, in the two user interfaces, the same widgets are usually used to enter the values of attributes/data of the same types. For example, to enter the value of a string-type attribute, both user interfaces provide a text box.

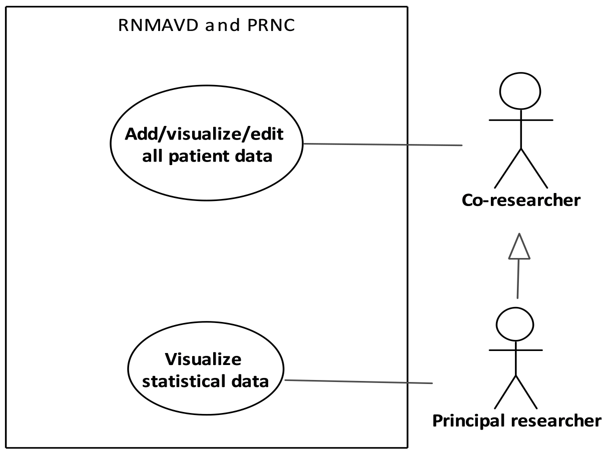

RNMAVD and PRNC
Add/visualize/edit all patient data
Visualize statistical data
Co-researcher
Principal researcher

Figure 3.9: RNMAVD and PRNC's use case diagram.

Table 3.5: Existing widgets and the corresponding attribute types available in Reuma.pt and Umedicine user interfaces.

Widget	Reuma.pt		Umedicine	
	Has?	**Attribute type**	**Has?**	**Attribute type**
Text box	Yes	String	Yes	String Date
Combo box	Yes	Enumerated	Yes	Enumerated
Check box	Yes	Boolean Enumerated list	Yes	Boolean
Radio button	Yes	Enumerated	Yes	Boolean Enumerated
Numeric input	Yes	Integer Date	Yes	Integer Float
Calendar	Yes	Date	Yes	Date
Range slider	Yes	Integer with a defined interval	No	NA (not applicable)

3.4 Discussion

Given the requirements analysis performed, we now discuss what CDRGen must support in relation to each of the three topics that were previously addressed, namely, data to store, functionalities and user interface.

Data to store. We will have the entity groups that were defined earlier at the beginning of this chapter as being the common entity groups of the clinical data registries, i.e., `person`, `diagnosis`, `treatment`, `questionnaire`, `medical examination` and `medical appointment`. Each entity group can be composed of one or more entities. Each entity can have one or more attributes.

Regarding the relationships between entities, we will have relationships between the `patient` entity and other entities as we found in the four clinical data registries that we analysed. As found in Umedicine, we will also support the existence of relationships between the following entities[10]:

- `Symptom` and `disease`
- `Treatment` and `medical examination`
- `Treatment` and `disease`
- `Medicine` and `disease`
- `Medicine` and `medication`.

Concerning the attribute types, we will support attributes of the following types: `string`, `numeric` (integer and float), `date`, `enumerated`, `list` (of one of the previous types), `boolean`, and `image`. This set of attribute types (excluding the `list of dates` and the `list of numbers`) is the result of the union of the sets of attribute types that we found in the four clinical data registries that we detailed in Section 3.1. We decided to also include the `list of dates` and the `list of numbers` in order to extend the attribute types that we will support. In fact, although these attribute types are not included in the clinical data registries that we detail, they may exist in a real clinical data registry. As we also found in two of the four clinical data registries, for an enumerated-type attribute it will be possible to collect another value other than the values that are predefined. Finally, as we found in Umedicine, for `string` and `numeric` attributes we will enable to define a limit of characters/digits.

With respect to the constraints between attributes, we will support the two following ones as detailed in Section 3.1 for PRNC:

- The value of an attribute can determine the assignment of values to other attributes (founded in all four clinical data registries).
- The value of an attribute must be one of the values that were already given to another attribute (founded in three of the four clinical data registries)[11].

Functionalities. The clinical data registry to be generated will be able to be accessed by four different types of users: (i) `patient`; (ii) `clerk`; (iii) `administrator physician`; and (iv) `non-administrator physician`. These types of users are the same as in Umedicine. We chose these user types since we believe that they cover the different user types that may exist. The `administrator physician` user type always exists for any generated clinical data registry, while the other three types of users may not necessarily exist. These types of users will be able to perform different functionalities according to the use case diagram represented in Figure 3.10.

[10]This will not be included in the system implementation
[11]This will not be included in the system implementation

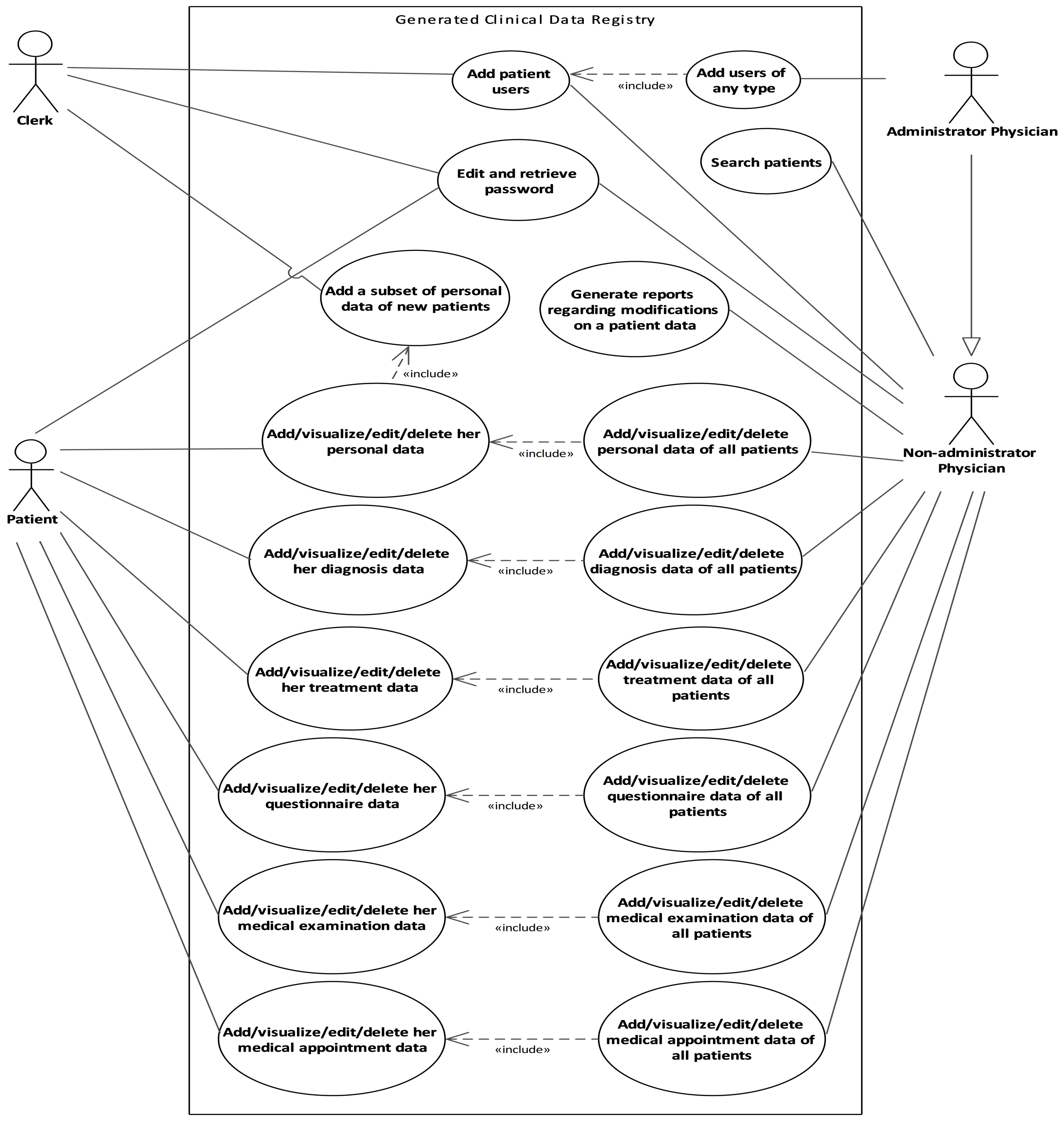

Generated Clinical Data Registry
Clerk
Administrator Physician
Patient
Non-administrator Physician
Add patient users
Add users of any type
«include»
Search patients
Edit and retrieve password
Add a subset of personal data of new patients
Generate reports regarding modifications on a patient data
«include»
Add/visualize/edit/delete her personal data
Add/visualize/edit/delete personal data of all patients
Add/visualize/edit/delete her diagnosis data
Add/visualize/edit/delete diagnosis data of all patients
Add/visualize/edit/delete her treatment data
Add/visualize/edit/delete treatment data of all patients
Add/visualize/edit/delete her questionnaire data
Add/visualize/edit/delete questionnaire data of all patients
Add/visualize/edit/delete her medical examination data
Add/visualize/edit/delete medical examination data of all patients
Add/visualize/edit/delete her medical appointment data
Add/visualize/edit/delete medical appointment data of all patients

Figure 3.10: Use case diagram of the clinical data registry to be generated.

i. The `patient` user will be able to:

- Add/visualize/edit/delete her personal data
- Add/visualize/edit/delete her diagnosis data
- Add/visualize/edit/delete her treatment data
- Add/visualize/edit/delete her questionnaire data
- Add/visualize/edit/delete her medical examination data

- Add/visualize/edit/delete her medical appointment data
- Edit and retrieve password.

ii. The `clerk` user will be able to:

- Add `patient` users
- Add a subset of personal data of new patients
- Edit and retrieve password.

iii. The `administrator physician` user will be able to:

- Add/visualize/edit/delete personal data of all patients
- Add/visualize/edit/delete diagnosis data of all patients
- Add/visualize/edit/delete treatment data of all patients
- Add/visualize/edit/delete questionnaire data of all patients
- Add/visualize/edit/delete medical examination data of all patients
- Add/visualize/edit/delete medical appointment data of all patients
- Search for patients by name or by other kind of identification
- Generate reports regarding modifications (add/edit/delete) on the data of a particular patient
- Add users of any type
- Edit and retrieve password.

iv. The `non-administrator physician` user will be able to perform the same functionalities as the `administrator physician` user, however it will only be able to add `patient` users instead of being able to add users of any type.

While the `administrator physician` user will be allowed to add/visualize/edit/delete data of any patient, the permissions of the other types of users may vary. For instance, the `patient` user may only be able to visualize a subset of her personal data.

These functionalities resulted from the generalization of the union of the functionalities supported by the clinical data registries that we analysed when performing the requirements analysis. In particular, we sought to prioritize the key functionalities that a clinical data registry usually has: add, visualize, edit and delete data. In addition, we enable to change the permissions of the user types over the data in order to give more flexibility. This way, we can always set different permissions over the data even though the system supports a predefined functionality set for each user type.

Table 3.6: Widget that will be used to enter data according to each type of attribute.

Attribute type	Widget to use
String and string list	Text box
Numeric and numeric list	Numeric input
Date and date list	Calendar
Boolean	Combo box
Enumerated	Combo box
Enumerated list	Check box
Image	File input

User Interface. We will predefine the widgets that will be used in the user interface to enter data according to each type of attribute, as shown in Table 3.6. These widgets were chosen taking into account the intersection of the widget sets that are used in the two clinical data registries as described in Section 3.3. For list-type attributes that do not exist in either clinical data registry, we decided that we will use the widget that was assigned to the list elements. For example, since text boxes will be used for `strings`, so text boxes will also be used for `string lists`. In the particular case of the `boolean` type, in order to generalize, we will use the same widget that we will be used for the `enumerated` type (i.e., a combo box), since the `boolean` type can be generalized in an enumerated with the values "Yes" and "No". In the case of the `image` type, although Umedicine supports this attribute type, its current version does not support the upload of images[12]. Therefore, we decided that we will use the file input widget for this type, since it is the only widget that is used in user interfaces to upload files (hence also images).

[12]In the current version of Umedicine, the `image` type only exists to simulate the visualization of the patient's photo

4

CDRGen: A Clinical Data Registry Generator

Contents

4.1 CDRGen Architecture 65

4.2 CDRGen Input 1/2: Specification 66

4.3 CDRGen Input 2/2: User Interface Labels 76

4.4 CDRGen Output: Clinical Data Registry 77

4.5 CDRGen: Parsing and Code Generation 78

4.6 Summary 94

In this chapter, we present a Clinical Data Registry Generator (CDRGen) that aims at addressing the problems identified in Chapter 1. CDRGen is able to generate a clinical data registry for any medical specialty with a minimum effort in terms of software design and development.

In Section 4.1, we describe the architecture of CDRGen. In Sections 4.2 and 4.3, we define the input of CDRGen. In Section 4.4, we describe the CDRGen's output. Finally, in Section 4.5, we detail the CDRGen's software modules.

4.1 CDRGen Architecture

The high-level architecture of CDRGen is illustrated in Figure 4.1. In general terms, the process

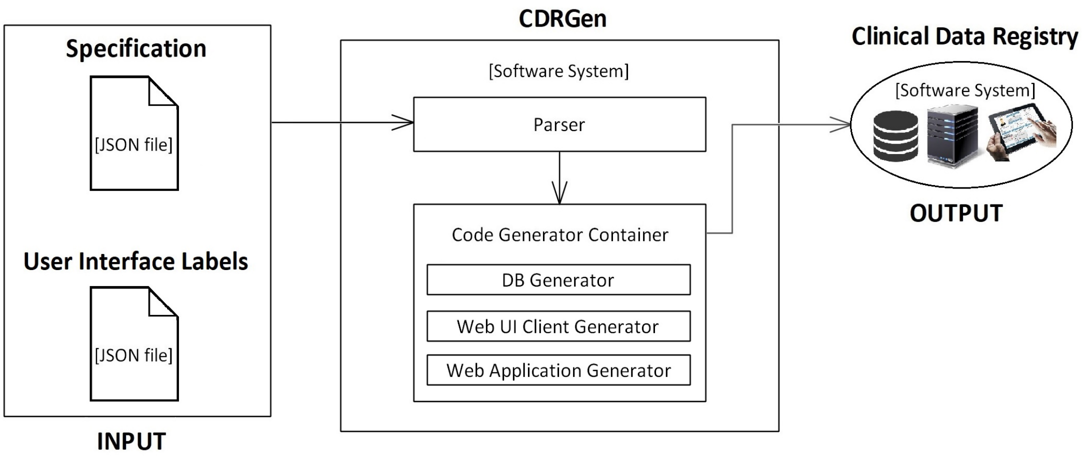


Figure 4.1: High-level architecture of CDRGen.

for generating a clinical data registry starts with two JSON files: (i) the JSON file *specification*; and (ii) the JSON file *user interface labels*. The JSON file *specification* describes the data to be collected by the clinical data registry. The JSON file *user interface labels* describes the labels of the user interface that are not related with the data to be collected by the clinical data registry (for instance, labels of buttons). The CDRGen reads these JSON files, parses them and generates code in order to produce a clinical data registry. The generated clinical data registry is similar to Umedicine in architectural and aesthetic aspects but also takes into consideration the requirements analysis that we performed (described in Chapter 3). We chose Umedicine as our reference for the generated clinical data registry since Umedicine was developed and validated in collaboration with a medical specialist and based on a set of requirements, so we believe that it is a good reference. In addition, during the development process of CDRGen, we also use Umedicine to test it since we know exactly what is the data to be collected and the functionalities that are provided.

4.2 CDRGen Input 1/2: Specification

In order to describe the specification to be given as one of the two parts of the CDRGen's input, we first designed a metamodel. This metamodel is used to describe the metadata that needs to be specified in order to generate a clinical data registry. This metamodel was designed based on the requirements analysis that we performed in Chapter 3, where we were able to know what a clinical data registry needs to support. In the requirements analysis, we identified:

- The four types of users that can be supported by a clinical data registry to be generated: (i) `patient`; (ii) `clerk`; (iii) `administrator physician`; and (iv) `non-administrator physician`.
- The six entity groups that form the basis of the data to be collected by a clinical data registry: (i) `person`; (ii) `diagnosis`; (iii) `treatment`; (iv) `questionnaire`; (v) `medical examination`; and (vi) `medical appointment`.

We must also bear in mind that a clinical data registry to be generated must have a name. Given this, the basis of the metadata of a clinical data registry to be generated consists of the clinical data registry's name, the types of users to be supported and the entity groups required to define the data to be collected.

In the specification, each entity group can optionally have a set of permissions and is composed of one or more entities. Each entity has a name, can optionally have a set of permissions and properties, and has one or more attributes. Each attribute has a name, a type and can optionally have a set of values, permissions and properties. In the particular case of string and numeric attributes, they can optionally have a minimum and a maximum number of characters/digits, respectively. In attributes:

- The **type** specifies the kind of values the attribute can hold (e.g., string, integer)
- The **set of values** is required for enumerated-type attributes. It shows the concrete values that can be hold by that attribute. For example, the gender can be an enumerated-type attribute whose values are male and female.

In entities and attributes, the **set of properties** specifies one or more properties. For example, an attribute can have the `not null` property, which indicates that it cannot hold the null value. An entity can have the `historic` property, indicating that it can have more than one instance, for example. In entity groups, entities and attributes, the **set of permissions** specifies if a user type can read and/or write (i.e., permission's value) a particular entity group/entity/attribute. Figure 4.2 represents the UML class diagram of the specification metamodel.

When instantiating the specification metamodel, we describe the metadata of the clinical data registry to be generated. To do this, we use the JSON format. The JSON format was chosen because it is

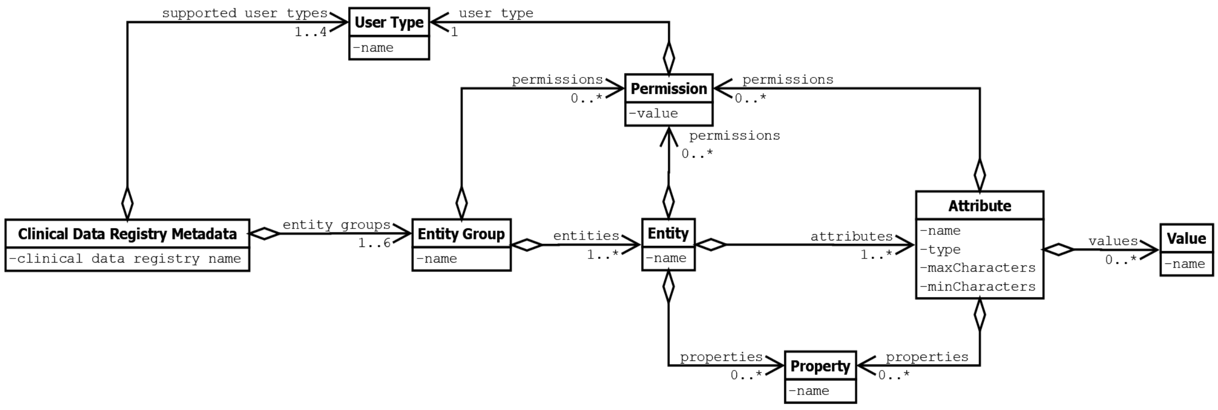


Figure 4.2: UML class diagram of the specification metamodel.

sufficiently expressive to represent the specification in a simple way and because there are many tools capable of manipulating JSON files. In this way, the specification is given as input to CDRGen as a JSON file.

Before describing the specification in detail, let us introduce a simple and concrete example:

(1) **Example.** Suppose that we want to generate a clinical data registry denominated *Simple Registry*. The *Simple Registry* can be accessed by two types of users: `administrator` `physician` and `patient`. In *Simple Registry*, we want to collect:

- The patient's name which, in this case, identifies the patient
- The patient's gender (male or female)
- The date when a patient starts and ends a radiotherapy treatment.

In *Simple Registry*, a `patient` user can visualize and change her name and gender, but he/she can only visualize the start and end dates of radiotherapy treatments.

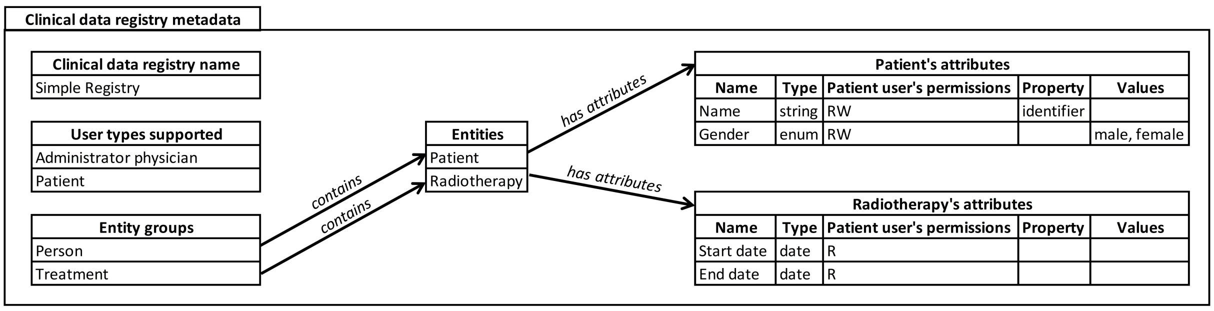
For Example (1), the specification is described, in a structured form, by the following six aspects:

- *Clinical data registry name*: Simple Registry.
- *Types of users supported*: `administrator` `physician` and `patient`.
- *Entities*: `Patient` and `Radiotherapy`.
- *Attributes* of each *entity*: `Patient` entity has a name that holds a string (the patient's identifier) and a gender, which has an enumerated type with "male" and "female" as possible values; `Radiotherapy` entity has a start and end dates, where both hold a date.
- The *permissions* of each *attribute* for the `patient` user: read (R) and write (W) the patient's name and the patient's gender; read (R) the radiotherapy's start date and radiotherapy's end date. We

only need to define the `patient` user's permissions since the `administrator physician` user has full access to any generated clinical data registry, i.e., he/she can read and write in all the entities' attributes.

- The *entities* that compose each *entity group*: `person` entity group is composed of the `Patient` entity; `treatment` entity group is composed of the `Radiotherapy` entity.

According to Example (1), the metadata of the clinical data registry to be generated is represented in Figure 4.3 and results in the JSON file specification illustrated in Figure 4.4[1].



Clinical data registry name
Simple Registry

User types supported
Administrator physician
Patient

Entity groups
Person
Treatment

Entities
Patient
Radiotherapy

Patient's attributes

Name	Type	Patient user's permissions	Property	Values
Name	string	RW	identifier	
Gender	enum	RW		male, female

Radiotherapy's attributes

Name	Type	Patient user's permissions	Property	Values
Start date	date	R		
End date	date	R		

Figure 4.3: Metadata of the clinical data registry to be generated according to Example (1).

In Figure 4.4, line 2 displays the name of the clinical data registry (the value of the `clinical data registry name` key). In line 3 are indicated the user types to be supported (the value of the `users` key). Note that since the `administrator physician` user always exists for any clinical data registry generated, it is not specified. In Lines 5-18 and 19-30 are indicated, respectively, the `person` and `treatment` entity groups. In lines 7-16 and 21-28 are specified, respectively, the `Patient` entity and the `Radiotherapy` entity. In lines 9-11 and 12-14 are specified, respectively, the patient's name and the patient's gender (i.e., the attributes of the `Patient` entity). In lines 23-24 and 25-26 are specified, respectively, the radiotherapy's start date and the radiotherapy's end date (i.e., the attributes of the `Radiotherapy` entity). Each attribute is specified as an object. In this object, each object key represents one attribute characteristic (e.g., permissions, properties), except for its name and type which are represented only by one object key: `attribute`. For instance, the `permissions` object key represents the attribute's permissions regarding each user type.

Let us now describe the JSON file specification in detail. The specification is composed of three keys referring to the root object of the JSON file (see Figure 4.5): (i) `clinical data registry name`; (ii) `users`; and (iii) `entity groups`. These three keys are described as follows:

i. The `clinical data registry name` key is mandatory and its value is a string that represents the

[1] In general, in JSON there are objects, arrays and strings. Each object is composed of key-value pairs, where the key is a string and the value can be a string, an array, or an object. The elements of an array can also be strings, arrays or objects. As an example, `{"keyName":"keyValue"}` represents an object that has a single key (denominated `keyName`) whose value is the `keyValue` string.

```
{
    "clinical data registry name": "Simple Registry",
    "users": ["patient"],
    "entity groups": {
        "person": {
            "entities": [
                {   "entity": [
                        "Patient",
                        {   "attribute": ["Name", "string"],
                            "properties": ["identifier"],
                            "permissions": { "patient": "RW" } },
                        {   "attribute": ["Gender", "enum"],
                            "values": ["male", "female"],
                            "permissions": { "patient": "RW" } }
                    ]
                }
            ]
        },
        "treatment": {
            "entities": [
                {   "entity": [
                        "Radiotherapy",
                        {   "attribute": ["Start date", "date"],
                            "permissions": { "patient": "R" } },
                        {   "attribute": ["End date", "date"],
                            "permissions": { "patient": "R" } }
                    ]
                }
            ]
        }
    }
}
```

Figure 4.4: The resulting JSON file specification according to Example (1).

```
{
    "clinical data registry name": ...,
    "users": [ ... ],
    "entity groups": {
        ...
    }
}
```

Figure 4.5: The root object of the JSON file specification.

name of the clinical data registry.

ii. The `users` key is optional and represents the types of users supported by the generated clinical data registry system. Its value is specified as an array that indicates the user types that the generated system must support beyond the `administrator physician` user type (since this is always supported). In this array, each element is a string that represents a user type and is restricted to the following three strings: `non-administrator physician`, `patient`, and `clerk`. In case one of the user types is not included in the array, it is assumed by default that the generated clinical data registry does not support this user type. In addition, if the `users` key is not included in the JSON file specification, then by default the generated clinical data registry only supports the `administrator physician` user type. As an example, Figure 4.6 illustrates how the value of the `users` key should be specified if we want to generate a clinical data registry that only supports the `administrator physician` and `non-administrator physician` user types (which ends up not supporting the `patient` and `clerk` user types).

```
"users": [ "non-administrator physician" ]
```

Figure 4.6: The value of the `users` key to generate a clinical data registry that supports the `administrator physician` and `non-administrator physician` user types.

iii. The `entity groups` key is mandatory and its value is specified as an object. This object specifies the entity groups that are required to represent the data to be collected by the clinical data registry.

In order to detail the data to be collected by the clinical data registry, in Sections 4.2.1, 4.2.2 and 4.2.3, we explain how the entity groups, entities and attributes are specified, respectively.

4.2.1 Entity Groups

The entity groups are specified in the value of the `entity groups` key. This value is specified as an object in which each key represents an entity group and is restricted to the following six strings: (i) `person`; (ii) `diagnosis`; (iii) `treatment`; (iv) `questionnaire`; (v) `medical examination`; and (vi) `medical appointment`. The `person` key is mandatory and the other five keys are optional. In addition, there may also exist the optional `permissions` key. Figure 4.7 illustrates the `entity groups` key value (an object) that includes all the possible keys that are allowed to be specified inside this object.

```
"entity groups": {
    "person": ...,
    "diagnosis": ...,
    "treatment": ...,
    "questionnaire": ...,
    "medical examination": ...,
    "medical appointment": ...,
    "permissions": ...
}
```

Figure 4.7: The value of the `entity groups` key that includes all the possible keys that are allowed to be specified.

In the `entity groups` key value (an object), the `permissions` key value is an object that specifies the permissions (object values) according to a user type (object key) for all the entity groups. The values and keys of the `permissions` object are restricted to certain values as follows:

- Keys: `non-administrator physician`; `patient`; and `clerk`

- Values: `R` (read); `W` (write); `RW` (read and write); `" "` (i.e., empty string, implying neither write nor read permissions)

As keys of the `permissions` object, it is only possible to insert three of the four possible user types supported by a clinical data registry because the `administrator physician` user type has read and write permissions by default. In case the `permissions` key is not included in the `entity groups` object, the default permissions that were built based on the requirements analysis are applied. By default, the `patient` user type has only read permissions over the `person` entity group. This default permission was built based on the two clinical data registries that support the `patient` user type: Umedicine and Reuma.pt (described in Sections 2.3.1 and 2.1.1, respectively). In Umedicine, the `patient` user type has read and write permissions over the `person` entity group. However, in Reuma.pt, the `patient` user type has only read permissions over the `person` entity group. In common to both clinical data registries, the `patient` user type ends up having read permissions over the `person` entity group. In the remaining four entity groups, by default all user types (excluding the `administrator physician`) do not have read or write permissions. These default permissions were built since, as far as we know, there are no common permissions over the other entity groups. Therefore, we ended up not giving read and write permissions, leaving this as something that has to be specified.

Still in the `entity groups` key value (an object), the value of a key that is specified with a name of an entity group (e.g., `person`) is an object that can contain three keys:

1. `label` (optional). The value of this key is a string that enables to customize the label of the entity group in the user interface.

2. `permissions` (optional). The value of this key is an object with the same syntax of the `permissions` object described for the entity groups. However, a `permissions` object defined inside an entity group specifies the permissions according to a user type only for that entity group.

3. `entities` (mandatory). The value of this key is an array of several objects. Each of these objects represents an entity.

As an example, Figure 4.8 illustrates how the value of the `entity groups` key should be specified if we want to generate a clinical data registry that has the `person` and the `treatment` entity groups, the `patient` user type has read and write permissions in the `person` entity group, but has only read permissions in the `treatment` entity group.

```
"entity groups": {
    "person": {
        ...,
        "permissions": { "patient": "RW" }
    },
    "treatment": {
        ...,
        "permissions": { "patient": "R" }
    }
}
```

Figure 4.8: The value of the `entity groups` key to generate a clinical data registry that has the `person` and the `treatment` entity groups, the `patient` user type has read and write permissions in the `person` entity group, but has only read permissions in the `treatment` entity group.

4.2.2 Entities

An entity is specified as an object that can contain four keys: `label` (mandatory in one case), `permissions` (optional), `properties` (optional) and `entity` (mandatory).

The `label` key value is a string that represents, in the user interface, the label of the entity that characterizes the patient. Therefore, the `label` key is mandatory and can only appear in the entity that characterizes the patient. The label of the other entities corresponds to the name given to each one of them.

The `permissions` key value is an object with the same syntax of the `permissions` objects described above. However, a `permissions` object defined inside an entity specifies the permissions according to a user type only for that entity.

The `properties` key value is an array. This array can optionally contain:

- The `historic` string or an object that contains only the `historic` key whose value is a string. Both these array elements indicate whether an entity is of the historical type. An historic-type entity is an entity that can contain several instances that are identified by a date-type attribute. In the case the `properties` array contains the `historic` string, then the date-type attribute is denominated “Date” by default. Otherwise, if the `properties` array contains the object that contains only the `historic` key, we can specify the date-type attribute name as the `historic` key value. As an example, Figure 4.9 illustrates how to specify the historic property to an entity in which the historic date-type attribute’s name is “Data”.

```
"properties": [ {"historic": "Data"} ]
```

Figure 4.9: Example of how to specify the historic property to an entity in which the historic date-type attribute’s name is “Data”.

- An object that contains only the `date` key whose value is a string. This object can only appear in non-historic entities of the `questionnaire` entity group. It is used to specify the name of the date-type attribute that exists by default in each entity of the `questionnaire` entity group. This default attribute holds the date in which a questionnaire is filled. In a historic-type entity of the `questionnaire` entity group, the object that only contains `historic` key (described above) can be used to achieve the same purpose.

- The `show` string, indicating that the concerning entity should be displayed on the user interface screen where the user has an overview of all the patient data. In simple terms, the entities that have this property are considered the most important to appear on the referred user interface screen.

Finally, the `entity` key value is an array. This array must contain at least two elements: one string as the first element and at least one object or array as the following element. The first element (a string) represents the entity's name. The following elements that are objects represent the entity's attributes. The following elements that are arrays represent sets of entity attributes. Each of these arrays must contain at least two elements: one string as the first element and at least one object or array as the following element. The first element (a string) represents the attribute set's name. Once more, the following elements that are objects represent the entity's attributes and those that are arrays represents sets of entity attributes. The attribute sets exist so that attributes that relate to a particular subject within its entity can be specified separately. For example, suppose we want to specify radiotherapy-related and chemotherapy-related attributes in the same entity. Instead of, for each attribute, we have to indicate in its name where it concerns radiotherapy or chemotherapy, we simple need to specify two sets of attributes: one containing only radiotherapy-related attributes and another containing only chemotherapy-related attributes.

In the specification file, we considered that existence of an entity that represents the patient is mandatory. This entity must be part of the `person` entity group and that is why the existence of this entity group is mandatory. The absence of an entity representing the patient would not make sense in a clinical data registry since it records patient clinical data.

4.2.3 Attributes

The object that represents an entity's attribute can contain eight keys:

- `attribute` (mandatory)
- `properties` (optional)
- `values` (mandatory and allowed only for enumerated-type attributes)
- `condition-yes` (optional and allowed only for boolean attributes)

- `condition-no` (optional and allowed only for boolean attributes)
- `min` (optional and allowed only for string and numeric attributes)
- `max` (optional and allowed only for string and numeric attributes)
- `permissions` (optional).

The `attribute` key value is an array that must contain two strings. The first string represents the attribute's name and the second represents the attribute's type. The attribute's type is restricted to one of the following fourteen types: (i) `string`; (ii) `list[string]`; (iii) `integer`; (iv) `list[integer]`; (v) `float`; (vi) `list[float]`; (vii) `date`; (viii) `list[date]`; (ix) `boolean`; (x) `enum`; (xi) `enum+`; (xii) `list[enum]`; (xiii) `list[enum+]`; and (xiv) `image`. The `enum+` and `list[enum+]` types allow the attribute to hold another value or values, respectively, other than the enumerated values that are specified.

The `properties` key value is a string array that can contain four strings:

- The `identifier` string must appear as a property of one (and only one) attribute of the entity that represents the patient. The attribute that contains the identifier property is the attribute whose value identifies the patient in the clinical data registry. The absence of an identifier for each patient would not make sense, as we have to relate the clinical data to each patient. As such, it is necessary to identify patients in some way, which in this case is done through an attribute of the entity that represents the patient.
- The `not null` string is optional and indicates whether the corresponding attribute must be filled in the clinical data registry (i.e., when filling a form).
- The `searchable` string is optional and indicates that patients can be searched by this attribute. Only attributes of the entity that represents the patient can have this property.
- The `show` string is optional and indicates that the attribute should be displayed on the user interface screen where the user has an overview of all the patient data.

The `values` key value is an array that can contain strings or objects with only one key. The strings and the object keys represent the enumerated values of an enumerated-type attribute (i.e. an attribute whose type is `enum`, `enum+` or a list of these). The key value of the objects is an array where attributes (through an object) or sets of attributes (through an array) can be specified. We call conditional attributes to the elements of this array, since the possibility of filling these attributes is related to the key of the concerned object. For example, suppose we have an enumerated-type attribute denominated `treatment` whose values are `radiotherapy` and `chemotherapy`. Suppose also that there is a string-type attribute denominated `complications` that holds the complications that occur during chemotherapy. It should only be possible to fill in the `complications` attribute if the value of the `treatment` attribute is chemotherapy, otherwise it makes no sense. Therefore, the complications attribute is a conditional attribute for chemotherapy treatments. Figure 4.10 illustrates how the treatment and the complication attributes must be specified.

```
{    "attribute": ["treatment", "enum"],
     "values": [ "radiotherapy",
                 {    "chemotherapy": [
                          { "attribute": ["complications", "string"] }
                      ]
                 }
     ]
}
```

Figure 4.10: Specification of the treatment and the complications attributes according to the example.

The condition-yes and the condition-no key values are arrays where attributes (through an object) or sets of attributes (through an array) can be specified. The elements of these arrays are the conditional attributes of a boolean attribute for the no and yes values, respectively.

The min and max key values are strings. For string-type attributes (i.e., strings and lists of strings), the value of these keys indicates the minimum and maximum length of the string that the attribute holds, respectively. For numeric-type attributes (i.e., integers, floats or lists of these), the value of these keys indicates the minimum and maximum values of the number that the attribute holds, respectively.

Finally, the permissions key value is an object with the same syntax of the permissions objects described above. However, a permissions object defined inside an attribute specifies the permissions according to a user type only for that attribute. This is the case where the permissions are more relevant. For instance, if we have an entity with write permissions for the patient user type, but we have an attribute of that entity with read and write permissions for the same user type, then the permissions defined for that attribute (read and write) are the ones that count. In summary, the permissions relevance follows this order (from low to high relevance): entity groups < entity group < entity < attribute.

As an example of the object that represents an attribute, Figure 4.11 illustrates how an object must be specified for an attribute with the following characteristics:

- Name: gender
- Type: enum
- Values: male; female
- Properties: not null

```
{
    "attribute": ["gender", "enum"],
    "values": ["male", "female"],
    "properties": ["not null"]
}
```

Figure 4.11: Specification of the object that represents an enumerated-type attribute whose name is "gender", that may have "male" and "female" as values, and not null as a property.

4.3 CDRGen Input 2/2: User Interface Labels

Beyond the JSON file *specification* that was described in Section 4.2, the user interface labels are also given as input to CDRGen. As the name implies, the user interface labels describe labels of the user interface that are not related with the data to be collected by the clinical data registry (for instance, labels of buttons, informative sentences). This labels are defined in a JSON file that we call *user interface labels*. This JSON file is only composed of an object that contains several key-value pairs. These key-value pairs enable to customize the labels of the user interface that, as we stated, are not related with the data to be collected by the clinical data registry. Figure 4.12 displays an excerpt of a JSON file *user interface labels* that is customized for the Portuguese idiom. In the JSON file *user interface labels*, as

```
{
    "Add" : "Adicionar",
    "Add administrator physician" : "Adicionar médico administrador",
    "Add clerk" : "Adicionar administrativo",
    "Add patient" : "Adicionar paciente",
    "Add physician" : "Adicionar médico",
    "Administrator" : "Administrador",
    "Authentication" : "Autenticação",
    "Back" : "Voltar",
    "Cancel" : "Cancelar",
    "Cancelling ..." : "A cancelar ...",
    "Change password" : "Mudar palavra-passe",
    "Choose one option." : "Escolha uma opção.",
    "Choose one or more options." : "Escolha uma ou mais opções.",
    "Current password" : "Palavra-passe actual",
    "Delete" : "Eliminar",
    ...
```

Figure 4.12: Excerpt of a JSON file *user interface labels* that is customized for the Portuguese idiom.

exemplified in Figure 4.12, the keys are predefined English strings and the corresponding value is a string that represents the label that is given to an element of the user interface. For example, `Add` is one of the keys, as also exemplified in Figure 4.12. The `Add` key value represents the label that is given to the add buttons that appear in the user interface. As shown in Figure 4.12, the `Add` key value is `Adicionar` and, therefore, the add buttons in the user interface will be labelled as `Adicionar` for the clinical data registries that are generated using the JSON file *user interface labels* shown in this figure. This way, we enable to customize the user interface labels in order to cover any idiom (e.g., Portuguese, English, French).

Since the user interface labels are not related to the data collected by the clinical data registry, we could predefine these labels for each idiom in different JSON files only once. Then, to generate a clinical data registry, we simply needed to indicate the predefined JSON file *user interface labels* that we wanted to use according to the idiom of the data to be collected by the clinical data registry. This way, a JSON file *user interface labels* that describes the user interface labels for one idiom could be used to generate all clinical data registries whose collected data is described in that idiom. For example, the same JSON file *user interface labels* that describes the labels in Portuguese can be used to generate all clinical data registries whose collected data is described in Portuguese.

Given that these files could thus be predefined for each idiom (since they are not related to the data to be collected), we end up not giving much emphasis to this part of the CDRGen's input throughout this thesis.

4.4 CDRGen Output: Clinical Data Registry

The architecture of the generated clinical data registry is the same as Umedicine's architecture and is illustrated in Figure 4.13.

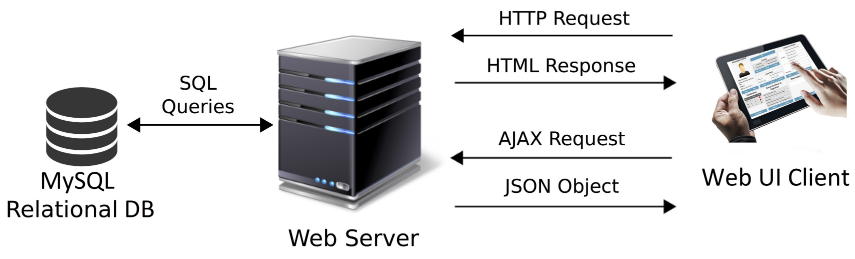


Figure 4.13: Architecture of the generated clinical data registry. Figure partially edited from Lages *et al.* [9].

Basically, the generated clinical data registry has a client-server architecture, which is the typical architecture of a system that queries a database whose access by users is achieved through a web browser. This client-server architecture is composed of three components: (i) a web user interface client, where the user interacts through a web browser; (ii) a web server; and (iii) a MySQL relational database. When a user interacts with the web user interface client, the web user interface client sends HTTP requests to the server that consequently returns HTML pages. In addition, the web user interface client can make Asynchronous JavaScript and XML (Ajax) requests to the server in order to receive new data that is necessary to fill a page. Ajax requests enable to minimize data transfers from the server, which consequently improves the performance, namely the user waiting time. When an Ajax request occurs, the relational database is queried by the server, which then returns the data as JSON objects to the web user interface client.

Analogously to Umedicine, the generated clinical data registry will be accessible to administrator and non-administrator physicians, clerks and patients[2]. These user types will have different permissions. For example, contrarily to physicians, a particular patient will not be able to access data from other patients. Other data read and write permissions regarding a user type (except `administrator physician`) can be defined in the specification, as described in Section 4.2. In addition, these user types will be able to perform different functionalities as we detailed in Section 3.4 and according to the use case diagram shown in Figure 3.10.

4.5 CDRGen: Parsing and Code Generation

CDRGen parses two JSON files (the JSON file *specification* and the JSON file *user interface labels*) and generates the code for creating the clinical data registry's database, web server and web user interface client. As shown in Figure 4.1 of Section 4.1, CDRGen is composed of the following four modules:

- Parser (Section 4.5.1)
- Database generator (Section 4.5.2)
- Web user interface client generator (Section 4.5.3)
- Web application generator (Section 4.5.4).

4.5.1 Parser

The *parser* is responsible for reading and parsing two JSON files: the JSON file *specification* and the JSON file *user interface labels*. Then, it sends the resulting data to the code generator modules (i.e., the database generator, the web user interface client generator and the web application generator). The data that results from the parsing of the JSON file *user interface labels* is sent only to the web user interface client generator module as an object that holds the key-value pairs that were defined in that JSON file. On the other hand, the data that results from the parsing of the JSON file *specification* is sent to the code generator modules in an object of the `CdrData` class. The `CdrData` class is composed of several attributes in conjunction with other classes that enable to store all the data that has been extracted from the parsing of the JSON file specification. Figure 4.14 shows the UML class diagram of

[2]Note that this only happens if all these types of users were specified in the specification, otherwise the clinical data registry can only be accessed by a subset of these user types.

the set of classes that are required to store the data extracted from the specification. The `CdrData` class

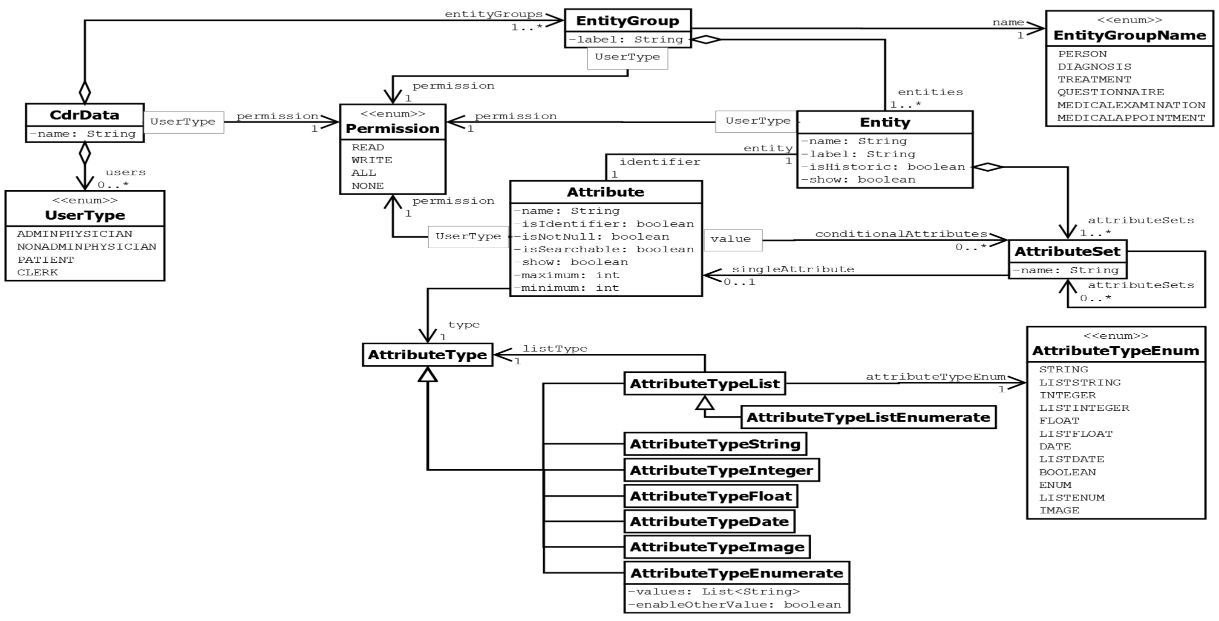


Figure 4.14: UML class diagram of the set of classes required to store the data extracted from the specification.

has the following attributes:

- `name`, a string that holds the name of the clinical data registry
- `permissions`, a map that holds the permissions of the entity groups for each user type
- `users`, an enumerated list that holds the user types specified
- `entityGroups`, a list of objects of the `EntityGroup` class which contains the entity groups that were specified.

The `UserType` is an enumerated-type class that holds the user types supported by a clinical data registry to be generated. The `Permission` is an enumerated-type class that holds the possible permissions values. The `EntityGroup` and `EntityGroupName` classes are used to store the entity groups that are specified. The `Entity` class is used to store the entities specified. The `Attribute` and `AttributeSet` classes are used to store the attributes and the sets of attributes specified. Finally, the `AtributeType` class and the classes that extend it are used to represent the types of attributes.

4.5.2 Database Generator

The *database generator* is responsible for creating a MySQL relational database to store the data that is entered by users through the user interface (i.e. the data collected by the clinical data registry).

The database to be created relies heavily on the specification. Specifically, each entity is represented by a table in the database and each attribute of that entity is represented by an attribute in the same table. However, the list-type attributes of an entity are treated differently. Since a list-type attribute can hold more than one value, for each of these attributes an auxiliary table is created. These auxiliary tables are created given the fact that MySQL does not support attributes with more than one value. The auxiliary table has a foreign key to the table that represents the entity that contains the list-type attribute. As an example, let us suppose that we have an entity (denominated `patient`) with just one attribute (denominated `comorbidities`) whose type is a list of strings. Then, the generated database will be composed of two tables according to the relational schema that is illustrated in Figure 4.15. The `patient` table represents the `patient` entity and the `patient_comorbidities` table represents the

```
patient(id)

patient_comorbidities(id, patient_id, comorbidity)

     patient_id: Foreign Key(patient.id)
```

Figure 4.15: The relational schema of the database that would be generated by CDRGen for an entity (denominated `patient`) with one list-type attribute (denominated `comorbidities`).

auxiliary table of the list-type attribute `comorbidities`. As stated before, the `patient_comorbidities` table contains a foreign key to the `patient` table which enables to assign the `comorbidities` attribute values to a given patient. This way, we can assign multiple values to a given attribute with MySQL since we are creating a one-to-many relationship between these two tables. Therefore, we end up having two types of tables in the database that is generated: the tables that represent an entity and the tables that represent the list-type attributes (i.e., the auxiliary tables).

For each non-auxiliary table that is created in the generated database, there is a relationship with the table that represents the `patient` entity (let us call it `patient` table). In the case that a table represents a non-historic entity, then this table has a one-to-one relationship with the `patient` table. On the other hand, if the table represents an historic-type entity, then this table has a many-to-one relationship with the `patient` table since the historic-type entity can contain several instances.

According to Example (1) introduced in Section 4.2, the relational schema of the generated database is illustrated in Figure 4.16. The database is composed of two tables (`Patient` and `Radiotherapy`) that contain their respective attributes, according to the specification. In addition, these tables have a one-to-one relationship since the `Radiotherapy` entity was specified as a non-historic entity (the default). Note that there is an extra attribute in both tables that was not defined in the specification (i.e., the `cdrGenInternalPatientId` attribute). This attribute represents the internal identifier of a patient that is used to identify and reference a patient in the database. This internal identifier is independent of the clinical data registry's data. In particular, the patient has this internal identifier that is used to identify the patient inside the system and has an external identifier that is used to identify the patient in the clinical data registry. While the attribute that represents the external identifier is defined in the specification, the attribute that represents the internal identifier is automatically created when we generate the database. We chose to create an internal identifier since this gives the possibility for the external identifier of a patient to change, which may happen if a user makes a mistake when entering data, for instance. If the patient's external identifier was changed and there was no internal identifier in the system (i.e., the internal identifier would be the external identifier), then it would be necessary to change all foreign key values that referenced that patient in the database.

```
Patient(cdrGenInternalPatientId, name, gender)

      UNIQUE(name)

Radiotherapy(cdrGenInternalPatientId, start_date, end_date)

      cdrGenInternalPatientId: Foreign Key(Patient.cdrGenInternalPatientId)
```

Figure 4.16: The relational schema of the database that would be generated by CDRGen according to the given JSON file specification for Example (1).

Besides the tables that are created according to the entities in the specification, we always create two predefined tables that store the users that can access to the user interface and their roles. In concrete, we create the `user` and the `userRole` tables. The `user` table stores the users that can access the user interface and the `userRole` table stores the roles (e.g., patient, clerk) of each user. Figure 4.17 illustrates the relational schema of these two tables. Note that the `patient` table represents the patient in the database.

```
patient(cdrGenInternalPatientId)

user(username, cdrGenInternalPatientId, password)

      cdrGenInternalPatientId: Foreign Key(patient.cdrGenInternalPatientId)

userRole(username, role)

      username: Foreign Key(user.username)
```

Figure 4.17: The relational schema of the `user` and the `userRole` tables.

In the `user` table, the `username` and `password` attributes represent the data that is necessary to login in the system. Due to security reasons, the `password` attribute value is encrypted and therefore the value stored is not the real password that is used to login. In order to associate a user to a patient, the `cdrGenInternalPatientId` attribute of the `user` table references the `cdrGenInternalPatientId` attribute of `patient` table (i.e., the internal identifier of a patient, as we stated above).

4.5.3 Web User Interface Client Generator

The *web user interface client generator* is responsible for generating the code that supports the interaction with the user. Specifically, this module creates the screens of the user interface. To that end, it generates Java Server Pages (JSP) files. The design of these screens is aesthetically based on the Umedicine user interface. These screens include the data entry forms and the screens for navigating through the data. The data entry forms (described in Section 4.5.3.1) are the screens where the user enters data. In concrete, they are the screens where the user adds a new instance of an entity or edits an instance. The screens for navigating through the data (described in Section 4.5.3.2) are the screens where the user visualizes and navigates through the data. The screen that displays the data of an entity instance is an example of a data navigation screen.

4.5.3.1 Data Entry Forms

Data entry forms are typically generated for each entity that is defined in the specification. In particular, two forms are generated: one to edit the data of an entity instance and another one to add a new instance. The generation of these data entry forms depends on the functionalities supported for each user type and also on the permissions that were set for each user type in the specification file. In the case that a user type does not have write permissions over an entity, then the data entry forms with respect to that entity are not generated for that user type. The same happens if the add and the edit functionalities are not supported for that user type. For example, if the `clerk` user type has all permissions, its functionalities allow the user only to add new instances of entities that were defined in the `person` entity group. Given this, the data entry forms with respect to an entity that appears in other entity groups never need to be generated for the `clerk` user type.

According to Example (1) introduced in Section 4.2, Figure 4.18 displays the form for editing an instance of the `Patient` entity in Simple Registry. The form is already filled, namely with a patient whose name is John and whose gender is male. To recall, the specified `Patient` entity had two attributes:

- The `Name` attribute, which is a string-type attribute. In the example, this attribute represents the patient identifier and, therefore, is mandatory.
- The `Gender` attribute, which is an enumerated-type attribute with `male` and `female` as values.

In general, as exemplified in Figure 4.18, a data entry form is composed of one or more widgets. Each widget is associated to an attribute of the concerned entity, and only appears if the concerned user type has write permissions over that attribute. The widget is composed of a label (which corresponds to the name of the attribute that was specified) and a field where the user can enter the value of the attribute. The type of this field depends on the type of the attribute, as we presented in Table 3.6 during the requirements analysis performed in Chapter 3. For example, in the `Patient` entity, the patient's

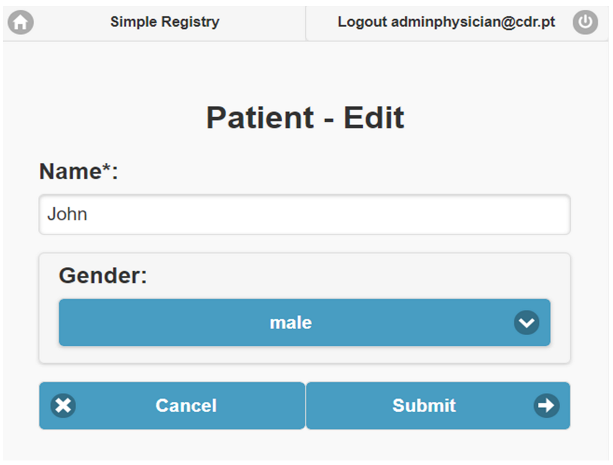


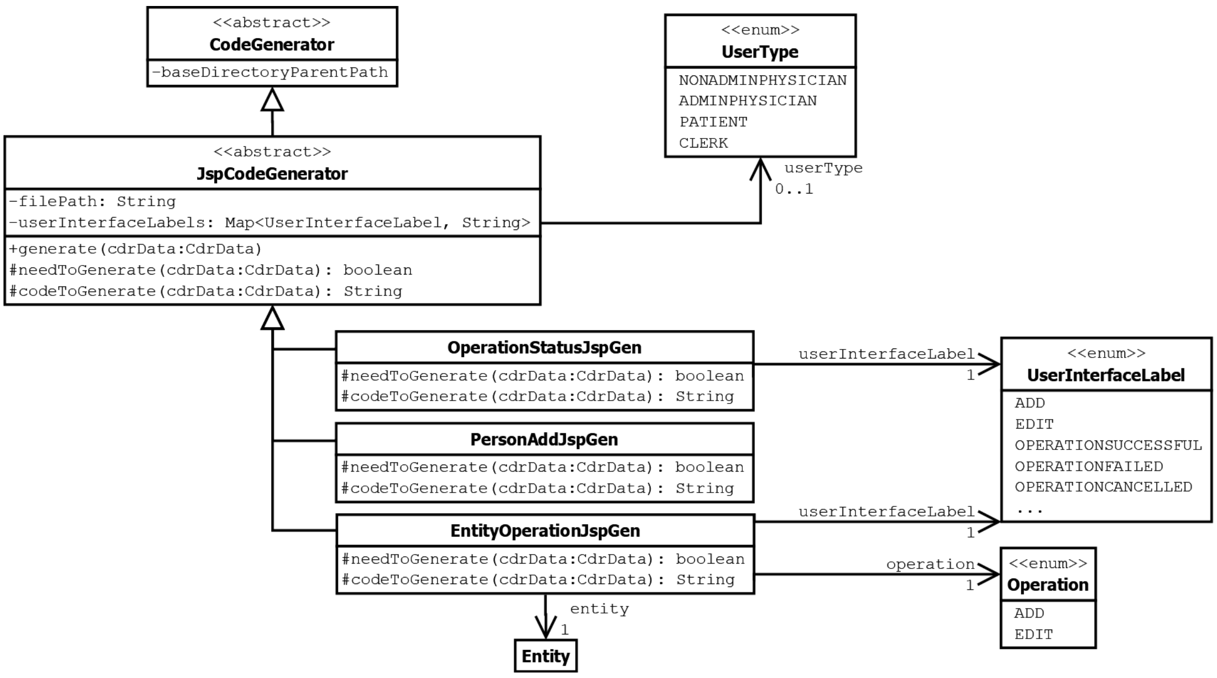
Figure 4.18: Form for editing an instance of the `Patient` entity in Simple Registry.

name is a string-type attribute and, therefore, a text box is used for the user to enter its value, as shown in Figure 4.18. In addition, in the case that the attribute is mandatory (i.e., if the attribute was specified with the `not null` property), then the label associated to that attribute is preceded by an asterisk (i.e., the "*" character), as is standard in forms. As an example, in the `Patient` entity, the patient's name is a mandatory attribute, so in Figure 4.18 its label is preceded by an asterisk.

A data entry form has also a title that is composed of the entity name preceded by the operation being performed (i.e., "Add" when adding a new instance and "Edit" when editing an entity instance). The data entry form displayed in Figure 4.18 is the form for editing an instance (i.e., edit operation) of the `Patient` entity (denominated *Patient*) and, thus, its title is *Patient - Edit*. Note, however, that the name of the operation corresponds to the name that is given to that operation in the JSON file that contains the key-value pairs corresponding to the user interfaces labels (i.e., the JSON file *user interface labels*) that are not related to the data that is described in the JSON file specification.

On the top side of the data entry forms are always two buttons, one on the left side and another on the right side, as also exemplified in Figure 4.18. The button on the left side is labelled with the name of the clinical data registry and redirects the user to its home page. On the other hand, the button on the right side allows the user to log out. Finally, on the bottom side of the data entry forms are also two or three buttons, depending on the operation and the type of the entity (i.e., whether it is a historic-type entity or not). In the data entry forms to add a new instance of an entity of any type or to edit an instance of a non-historic entity appear two buttons, one on each side, as exemplified in Figure 4.18. The button on the left side (i.e., the `Cancel` button) cancels the operation and the button on the right side (i.e., the `Submit` button) submits the operation. In addition to these buttons, in the data entry forms to edit an instance of an historic-type entity appears a button in the middle of these that allows the user to delete the concerned instance.

Figure 4.19 illustrates the UML class diagram of the classes that are used to generate the data entry forms. To avoid extending the diagram too much, we omit the attributes of the `Entity` class and most of the enumerated values of the `UserInterfaceLabel` enumerated-type class. In this diagram, we have



<<abstract>>
CodeGenerator
-baseDirectoryParentPath
<<enum>>
UserType
NONADMINPHYSICIAN
ADMINPHYSICIAN
PATIENT
CLERK
userType
0..1
<<abstract>>
JspCodeGenerator
-filePath: String
-userInterfaceLabels: Map<UserInterfaceLabel, String>
+generate(cdrData:CdrData)
#needToGenerate(cdrData:CdrData): boolean
#codeToGenerate(cdrData:CdrData): String
OperationStatusJspGen
#needToGenerate(cdrData:CdrData): boolean
#codeToGenerate(cdrData:CdrData): String
userInterfaceLabel
1
<<enum>>
UserInterfaceLabel
ADD
EDIT
OPERATIONSUCCESSFUL
OPERATIONFAILED
OPERATIONCANCELLED
...
PersonAddJspGen
#needToGenerate(cdrData:CdrData): boolean
#codeToGenerate(cdrData:CdrData): String
userInterfaceLabel
1
EntityOperationJspGen
#needToGenerate(cdrData:CdrData): boolean
#codeToGenerate(cdrData:CdrData): String
operation
1
<<enum>>
Operation
ADD
EDIT
entity
1
Entity

Figure 4.19: UML class diagram of the set of classes used to generate the data entry forms.

the following classes:

- `CodeGenerator` is the class that holds the directory path where the generated clinical data registry is created (i.e., the value of the `baseDirectoryParentPath` attribute).
- `JspCodeGenerator` generates a JSP file. The `filePath` attribute holds the directory path where the JSP file must be generated. The `userInterfaceLabels` attribute holds the labels of the buttons of the user interface and the labels of other aspects that are not related with the JSON file specification. Every class that generates a JSP file extends this class. Each JSP file is generated through the `generate` method. This method follows the Template Method design pattern and requires two methods: `needToGenerate` and `codeToGenerate`. Both these methods have the `cdrData` as a parameter (i.e., the object that holds the data extracted from the JSON file specification). The `needToGenerate` method is used to check if the JSP file needs to be generated according to a set of conditions (for example, the user type permissions with respect to an entity). The `codeToGenerate` method is used to build and return the code (i.e., text) that the JSP file to be generated must have. Given these two methods, the `generate` method starts to call the `needToGenerate` method in order to check if the JSP file needs to be generated. If the returned value is true, then it calls the `codeToGenerate` method to get the code that the JSP file to be generated must have. Then, it generates a JSP file with that code in the file path that is set in the `filePath` attribute.

Since the code that the JSP file must have depends on the user type (due to their permissions and supported functionalities, for example), there is also the `userType` attribute.

- `OperationStatusJspGen` generates the screens that appear after the user performs an operation (e.g., press the submit button in a data entry form). These screens display the status of the operation: if it was successful, if it failed or if it was cancelled.
- `PersonAddJspGen` generates the data entry form with respect to the person entity group that can be filled after creating a new patient. Since this form is the only one that encompasses multiple entities, it needs a class for itself.
- `EntityOperationJspGen` generates the data entry forms for an entity according to an operation (add or edit).

4.5.3.2 Data Navigation Screens

Concerning the *data navigation screens*, there are three types of screens:

- The screen that displays an overview of all the data of a patient.
- The screen that displays all data of an entity (i.e., all data of its instances) with respect to a patient.
- The screen that displays the most recent instance of an entity (for the historic-type entities) or the unique instance of an entity (for the non-historic entities) with respect to a patient.

All these three types of screens are included in the Umedicine user interface. However, some aspects were modified in order to handle scalability issues and other aspects that Umedicine does not support.

According to Example (1) introduced in Section 4.2, Figure 4.20 displays the *screen that gives an overview of all the data of a patient* in Simple Registry. This type of screen is divided in different sections,

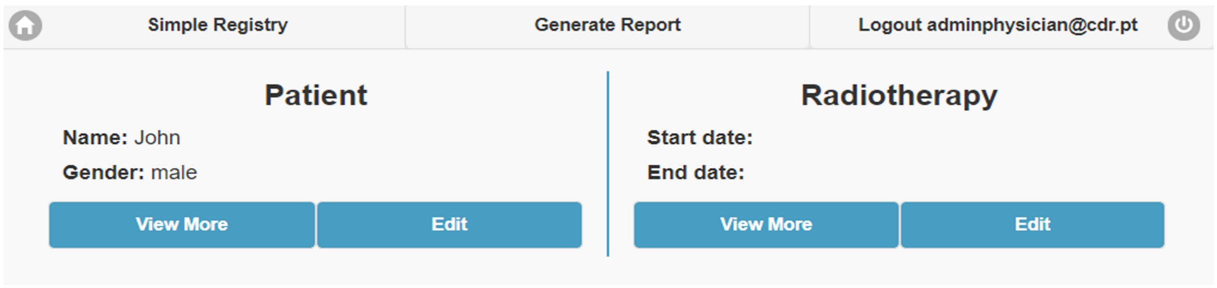


Figure 4.20: Screen that displays an overview of all data of a patient in Simple Registry.

each of which relates to an entity group that was defined in the specification. Since only two entity groups were specified for Simple Registry, the screen displayed in Figure 4.20 has only two sections. The left side of the screen displays the section concerning the `person` entity group, and the right side displays the section concerning the `treatment` entity group. In order to scale up, a section can occupy 50% of the screen vertically or 100%, depending on the number of entity groups that were specified. For example, if all six entity groups were specified, then there would be six sections (three on the top side of the screen and other three on the bottom side) that would occupy 50% of the screen vertically. This situation happens in the Umedicine clinical data registry, as we presented in Figure 2.6 in Section 2.3.1.

The way each section is presented in this type of screen depends on the entity group and the number of entities that it has. With respect to the `person` entity group and the other entity groups that only have one entity, it is displayed a set of attributes (name and value) in the corresponding section. This is the case of the two sections that are shown in Figure 4.20. The attributes shown in each section compose the single entity that each entity group has. In the particular case of the `person` entity group, the attributes of the entity that represents the patient are always displayed even if the `person` entity group has more entities. In the case that we are dealing with a historic-type entity (an entity that can have more than one instance), the attribute values displayed refer to the entity instance that is more recent chronologically. With respect to the sections of the `questionnaire` entity group and the entity groups that have more than one entity, a table is displayed. The rows of this table represent an entity from the respective entity group. As an example, suppose that in Simple Registry we had also specified the `First Exam` and `Second Exam` entities in the `medical examination` entity group. Figure 4.21 exemplifies what this screen type would look like in this case.

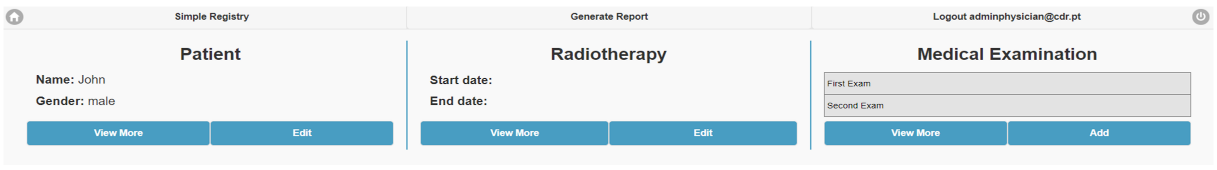


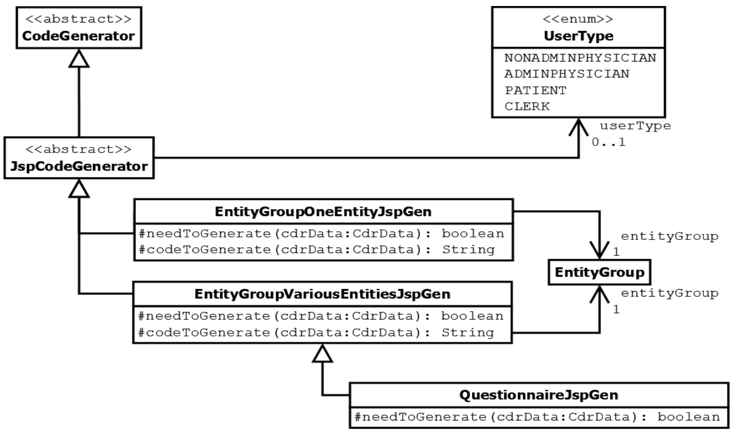
Figure 4.21: Screen that displays an overview of all data of a patient if, in Simple Registry, we had also specified the `First Exam` and `Second Exam` entities in the `medical examination` entity group.

As also shown in Figure 4.20, at the bottom of each section are several buttons. The existence of these buttons depends on what the section displays (whether an attribute list or a table) and on the permissions of the concerned user type.

The `View More`, the `Edit`, and the `Add` buttons may exist if a section displays a list of attributes of an entity. However, the `Add` button only exists if the concerned entity is of the historical type. This button redirects the user to the data entry form that allows the user to add a new instance of that entity. The `Edit` button redirects the user to the data entry form that allows the user to edit the entity instance that is shown in the section. For example, in Figure 4.20, if the user presses the `Edit` button displayed at the bottom of the section that represents the `person` entity group (i.e., the section displayed on the left), then he is redirected to the screen shown in Figure 4.18 (i.e., the data entry form for editing the concerned instance of the `Patient` entity). The `Add` and the `Edit` buttons only exist if the user type has write permissions over the concerned entity. On the other hand, the `View More` button only exists if the concerned user type has read permissions over the concerned entity. This button redirects the user to the screen that displays all data of an entity of the concerned entity group[3].

In the case that a section displays a table, as shown in Figure 4.21 on the right side, the `View More` and the `Add` buttons may exist. The existence and the functionality of the `View More` button follows the same reasoning as above. On the other hand, the `Add` button only exists if the concerned user type has write permissions over the concerned entity group. This button redirects the user to a screen that displays a list of entities that compose the concerned entity group. If the user clicks on an item of this list (i.e., an entity), the user is redirected to the data entry form that allows the user to add a new instance of that entity (i.e., a form similar[4] to that shown in Figure 4.18).

Figure 4.22 illustrates the UML class diagram of the classes that are used to generate the type of screen that gives an overview of all the data of a patient. Since the `CodeGenerator`, the `JspCodeGenerator` and the `EntityGroup` classes were described above, we omitted their methods and attributes. In resume, in this diagram we have a class for each section type. The `EntityGroupOneEntityJspGen` class generates the sections that are composed of a set of attributes of a given entity and the `EntityGroupVariousEntitiesJspGen` class generates the sections that display a table. In addition, the `QuestionnaireJspGen` class is required to generate the section that represents the `questionnaire` entity group. This happens because the table of this section is different, since it shows the last date on which each questionnaire was filled. The `needToGenerate` and `codeToGenerate` methods of these classes are also used in the same manner as we described for the classes that we presented in Figure 4.19.



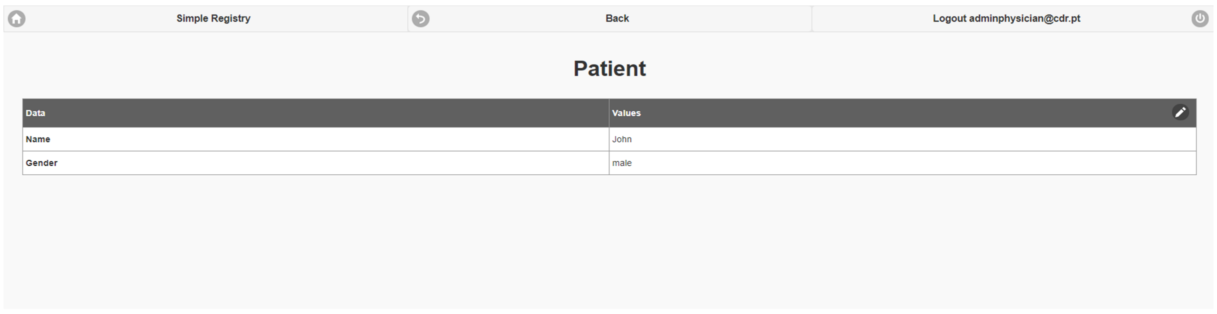
<<abstract>>
CodeGenerator
<<enum>>
UserType
NONADMINPHYSICIAN
ADMINPHYSICIAN
PATIENT
CLERK
userType
0..1
<<abstract>>
JspCodeGenerator
EntityGroupOneEntityJspGen
#needToGenerate(cdrData:CdrData): boolean
#codeToGenerate(cdrData:CdrData): String
entityGroup
1
EntityGroup
entityGroup
1
EntityGroupVariousEntitiesJspGen
#needToGenerate(cdrData:CdrData): boolean
#codeToGenerate(cdrData:CdrData): String
QuestionnaireJspGen
#needToGenerate(cdrData:CdrData): boolean

Figure 4.22: UML class diagram of the set of classes used to generate the screen that displays an overview of all data of a patient.

According to Example (1) introduced in Section 4.2, Figure 4.23 illustrates the *screen that displays all data of an entity* in Simple Registry. In concrete, it is displayed the screen that allows the user

[3]This is the next type of screen that we will detail
[4]Instead of a filled form, it is a blank form.

to visualize all data of the `Patient` entity that was specified in the `person` entity group. This type of



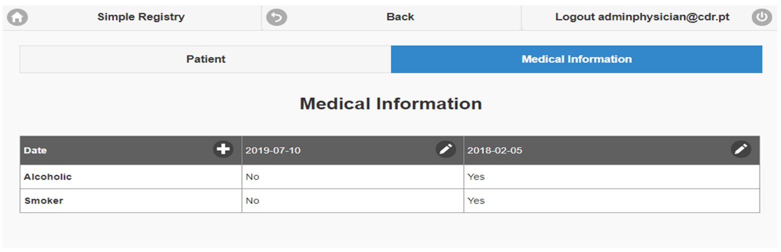
Data	Values
Name	John
Gender	male

Figure 4.23: Screen that displays all data of the `Patient` entity in Simple Registry.

screen can be accessed by pressing the `View More` buttons displayed in the first type of screen that is exemplified in Figure 4.20. In this type of screen, we can visualize all the attributes (names and values) and all instances of an entity of a certain entity group. This way, we can see the entities in more detail, unlike what happened on the screen that displays an overview of all data of a patient.

In this type of screen is displayed a table in which each row describes the name and the value(s) of an attribute, as shown in Figure 4.23. The attribute name appears in the first column, and their values appear in the preceding columns. If we are visualizing all the data of a non-historic entity, then the table has only two columns. This case is exemplified in Figure 4.23 because it shows the data of the `Patient` entity and this is a non-historic entity (i.e., an entity that only has one instance for a given patient). As we can see, in the first column of the table shown in Figure 4.23 are the attribute names of the `Patient` entity (i.e., `Name` and `Gender`) and in the second column are the corresponding values of the unique instance of the `Patient` entity (i.e., `John` and `male`, respectively). On the other hand, if we are visualizing all the data of a historic-type entity, then the table can have more than two columns.

To exemplify the case where the user visualizes all the data of a historic-type entity, let us suppose, in Simple Registry, that a historic-type entity denominated `Medical Information` has been specified in the `person` entity group as well. Figure 4.24 exemplifies what this screen type would look like if the `Medical Information` entity had two instances for the concerned patient. Each of these two instances



Date	2019-07-10	2018-02-05
Alcoholic	No	Yes
Smoker	No	Yes

Figure 4.24: Screen that displays all data of the `Medical Information` historic-type entity that has two instances for the concerned patient.

appears in a column chronologically ordered by the `Date` historic-type attribute. In addition, note that, in this example, the `person` entity group would have more than one entity and, in this case, this type of screen displays several tabs on the top side, one for each entity as shown in Figure 4.24. Therefore, the user can access to other entities of the concerned entity group by pressing other tabs. Since in the original example of Simple Registry only one entity was defined in the `Person` entity group (namely, the `Patient` entity), the screen shown in Figure 4.23 does not displays any tab on the top side.

Regarding the table that is displayed in this type of screen, in the first row of each column that represents an entity instance may exists a button (represented by a pencil) that redirects the user to the data entry form that allows him to edit that entity instance. In addition, in the first row of the first column may also exists a button (represented by plus symbol) that redirects the user to the data entry form that allows him to add a new instance of that entity. Both these buttons only exist if the user has write permissions over the concerned entity. In addition, both these buttons are displayed in Figure 4.24.

Figure 4.25 illustrates the UML class diagram of the classes that are used to generate this type of screen. The `EntityGroupViewMoreJspGen` class generates the screen for entity groups that are composed of more than one entity (i.e., the screen that has tabs to access each entity on the top side). On the other hand, the `EntityGroupOneEntityViewMoreJspGen` class generates the screen for entity groups that are composed of a single entity (the screen that has no tabs on the top side). The `needToGenerate` and `codeToGenerate` methods of these classes are also used in the same manner as we described for the classes that we presented above.

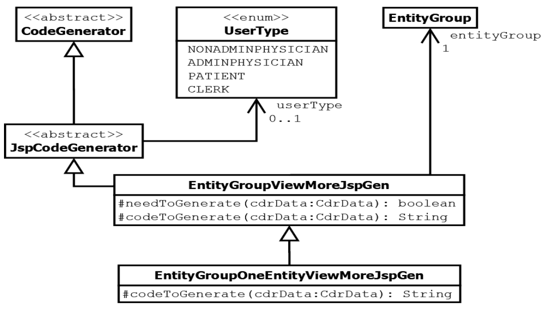

<<abstract>>
CodeGenerator
<<enum>>
UserType
NONADMINPHYSICIAN
ADMINPHYSICIAN
PATIENT
CLERK
EntityGroup
entityGroup
1
userType
0..1
<<abstract>>
JspCodeGenerator
EntityGroupViewMoreJspGen
#needToGenerate(cdrData:CdrData): boolean
#codeToGenerate(cdrData:CdrData): String
EntityGroupOneEntityViewMoreJspGen
#codeToGenerate(cdrData:CdrData): String

Figure 4.25: UML class diagram of the set of classes used to generate the screen that displays all data of an entity.

The *screen that displays the most recent instance of an entity* (for the historic-type entities) *or the unique instance of an entity* (for the non-historic entities) can be accessed by pressing a row of the tables that are displayed in the first type of screen that is exemplified in Figure 4.20. As happens in the second type of screen (i.e., the screen that displays all data of an entity), in this screen type a table is also displayed. Since the user only visualizes one instance (the most recent chronologically or the unique instance of an entity), the table displayed can only have two columns. Once more, each row describes the name and the value(s) of an attribute. The attribute name appears in the first column, and their values appear in the second column.

Figure 4.26 illustrates the UML class diagram of the classes that are used to generate this type of screen. The `EntityViewJspGen` class generates this screen type. The `needToGenerate` and `codeToGenerate` methods of this class are also used in the same manner as we described for the classes that we presented above.

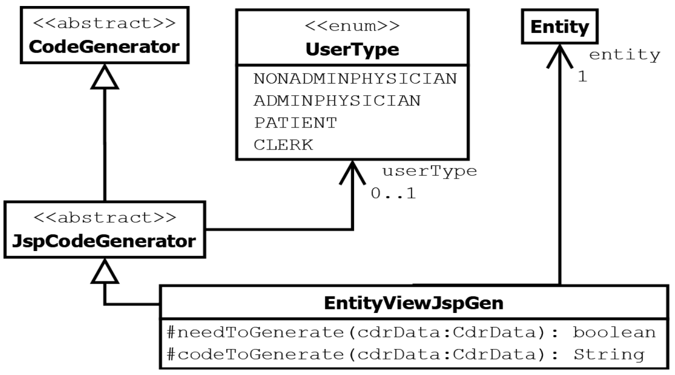


Figure 4.26: UML class diagram of the set of classes used to generate the screen that displays one instance (the unique or the most recent chronologically) of an entity.

4.5.4 Web Application Generator

The *web application generator* is responsible for generating the code regarding the communication between the database and the user interface (i.e., the web server's role). As in Umedicine, the web server must answer to requests sent by web clients, perform operations on the data, and store and retrieve data from the database [9]. The web server of the generated clinical data registry has a structure equal to the Umedicine's web server: a Java application that uses the Spring Framework ecosystem[5]. We decided to use the Spring Framework because we have access to a concrete example of the code that uses it and this greatly facilitates the process of generating a generalization of this code. This happens because we have access to Umedicine (our reference base as a clinical data registry) and Umedicine's web server is already developed with the Spring Framework, as we mentioned. In addition, the server component architecture is also the same as Umedicine's web server and is illustrated in Figure 4.27. It follows the Model-View-Controller design pattern, in which we have the model and the controller layers in the web server[6]. The model layer is composed of data access objects and services, while the controller layer is only composed of controllers. These three concepts are defined as follows:

- *Controllers* are responsible for the web client interactions, in which they receive web client requests, invoke services and send responses.
- *Services* are invoked by controllers or other services in order to process the data that is sent by clients or that is retrieved from the database. Basically, the services contain the business logic and

[5]https://spring.io/
[6]The view layer is generated by the web user interface client generator module

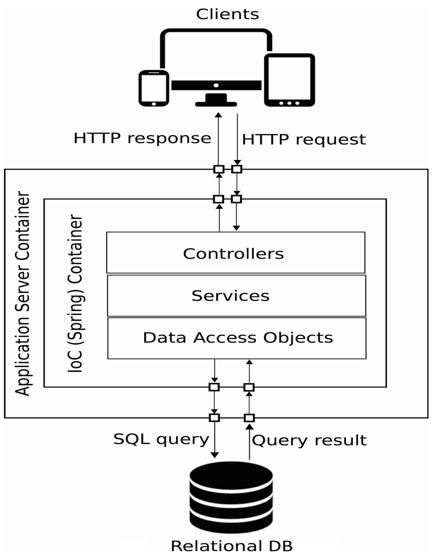


Figure 4.27: The web server component architecture. Figure from Lages *et al.* [9].

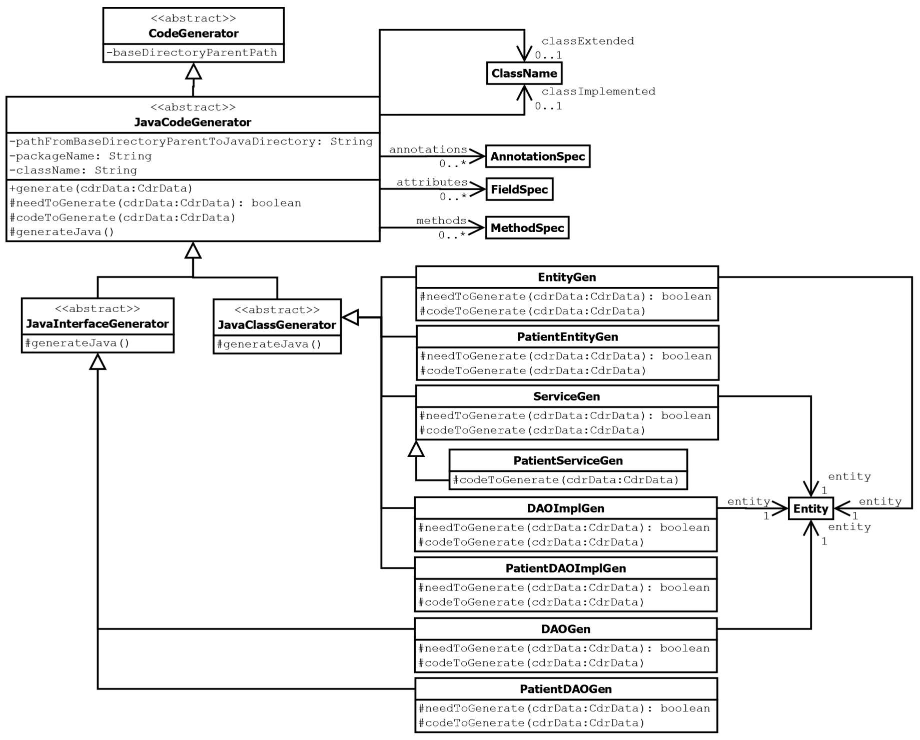
process the data for the controllers use in their responses to clients.

- *Data Access Objects* are responsible for the interaction between the server and the database. Namely, it contains Structured Query Language (SQL) instructions that enable to store and retrieve data from the database. The data retrieved is converted into Java objects and then forwarded to the service layer.

The code required to generate the web server of the clinical data registry to be generated is divided into three parts according to each of the three concepts described above. In concrete, we have classes to generate Java classes that represent the controllers, the services and the data access objects. In addition, there are also classes to generate Java classes in the generated clinical data registry that represent the entities that were defined in the specification file. As we referred when describing the data access objects, these are the classes used to convert the data retrieved from the database to Java objects.

Figure 4.28 illustrates the UML class diagram of the classes that are used to generate the *services*, the *data access objects* and the *Java classes that represent the defined entities in the specification file*. In this diagram, we have the following classes:

- `JavaCodeGenerator` is the class whose attributes hold the data required to generate a Java file. The `pathFromBaseDirectoryParentToJavaDirectory` attribute holds the partial directory path from the base directory of the generated clinical data registry to the base directory where the Java files are created. The `packageName` and `className` attributes holds, respectively, the package and the class names of the class to generate. The `classExtended` and `classImplemented` attributes holds the class name that the class to generate extends and implements, respectively.



<<abstract>>
CodeGenerator
-baseDirectoryParentPath
<<abstract>>
JavaCodeGenerator
-pathFromBaseDirectoryParentToJavaDirectory: String
-packageName: String
-className: String
+generate(cdrData:CdrData)
#needToGenerate(cdrData:CdrData): boolean
#codeToGenerate(cdrData:CdrData)
#generateJava()
classExtended
0..1
ClassName
classImplemented
0..1
annotations
0..*
AnnotationSpec
attributes
0..*
FieldSpec
methods
0..*
MethodSpec
<<abstract>>
JavaInterfaceGenerator
#generateJava()
<<abstract>>
JavaClassGenerator
#generateJava()
EntityGen
#needToGenerate(cdrData:CdrData): boolean
#codeToGenerate(cdrData:CdrData)
PatientEntityGen
#needToGenerate(cdrData:CdrData): boolean
#codeToGenerate(cdrData:CdrData)
ServiceGen
#needToGenerate(cdrData:CdrData): boolean
#codeToGenerate(cdrData:CdrData)
PatientServiceGen
#codeToGenerate(cdrData:CdrData)
DAOImplGen
#needToGenerate(cdrData:CdrData): boolean
#codeToGenerate(cdrData:CdrData)
PatientDAOImplGen
#needToGenerate(cdrData:CdrData): boolean
#codeToGenerate(cdrData:CdrData)
DAOGen
#needToGenerate(cdrData:CdrData): boolean
#codeToGenerate(cdrData:CdrData)
PatientDAOGen
#needToGenerate(cdrData:CdrData): boolean
#codeToGenerate(cdrData:CdrData)
entity
1
entity
1
Entity
entity
1
entity
1

Figure 4.28: UML class diagram of the set of classes required to generate the services, the data acess objects and the Java classes that represent the entities.

The `annotations` attribute holds the annotations of the class to create (e.g., @Controller). Finally, the `attributes` and `methods` attributes holds, respectively, the attributes and the methods of the class to generate. The `ClassName`, `AnnotationSpec`, `FieldSpec`, `MethodSpec` classes are part of the JavaPoet[7] library that helps us generate Java files. Every concrete class that extends this class generates a Java file through the `generate` method. This method follows the Template Method design pattern (as in the `JspCodeGenerator` class) and requires three methods: `needToGenerate`, `codeToGenerate` and `generateJava`. The `needToGenerate` method is used in the same manner as we described in the classes that generate JSP files. The `codeToGenerate` method is used by the concrete classes to set the attributes of this class (for example, the annotations, the attributes and the methods). Finally, the `generateJava` method is used to generate a Java file. The `generate` method starts to call the `needToGenerate` method in order to check if the Java file needs to be generated. If the returned value is true, then it calls the `codeToGenerate` and the `generateJava` methods.

- `JavaClassGenerator` describes the behaviour of the `generateJava` method in order to generate a Java class according to the values of the attributes of the super class (i.e., generates a Java class in a given package, with a given name, annotations, attributes and methods). Every concrete class

[7]https://github.com/square/javapoet

that extends this class generates a Java class.

- `JavaInterfaceGenerator` describes the behaviour of the `generateJava` method in order to generate a Java interface according to the values of the attributes of the super class. Every concrete class that extends this class generates a Java interface.

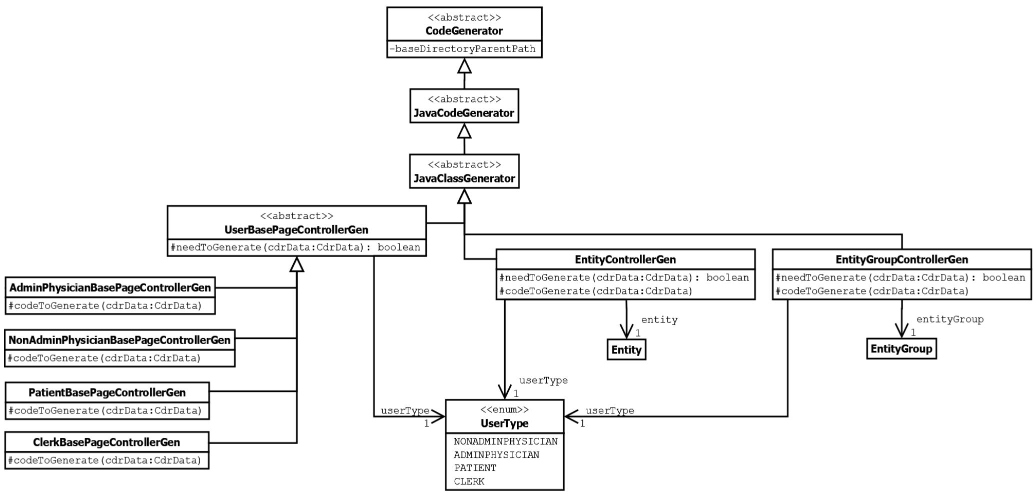
- `ServiceGen` generates the service for a given entity. Since the service of the entity that represents the patient requires more methods (for instance, to search a patient), the `PatientServiceGen` class is also needed.

- `DAOGen` generates the interface of the data access object for a given entity. Once again, since the entity that represents a patient requires more methods, the `PatientDAOGen` class is also needed.

- `DAOImplGen` generates the class that implements the `DAOGen` interface for a given entity. The `PatientDAOImplGen` class is also needed because it generates the class that implements the `PatientDAOGen` interface.

- `EntityGen` generates the class that represents a given entity that was defined in the JSON file specification. Since the entity that represents a patient is treated separately, the `PatientEntityGen` class is also needed.

Figure 4.29 illustrates the UML class diagram of the classes that are used to generate the *controllers*. Since we described the `JavaCodeGenerator` and the `JavaClassGenerator` classes above, we



<<abstract>>
CodeGenerator
-baseDirectoryParentPath
<<abstract>>
JavaCodeGenerator
<<abstract>>
JavaClassGenerator
<<abstract>>
UserBasePageControllerGen
#needToGenerate(cdrData:CdrData): boolean
AdminPhysicianBasePageControllerGen
#codeToGenerate(cdrData:CdrData)
NonAdminPhysicianBasePageControllerGen
#codeToGenerate(cdrData:CdrData)
PatientBasePageControllerGen
#codeToGenerate(cdrData:CdrData)
ClerkBasePageControllerGen
#codeToGenerate(cdrData:CdrData)
EntityControllerGen
#needToGenerate(cdrData:CdrData): boolean
#codeToGenerate(cdrData:CdrData)
EntityGroupControllerGen
#needToGenerate(cdrData:CdrData): boolean
#codeToGenerate(cdrData:CdrData)
entity
1
Entity
entityGroup
1
EntityGroup
userType
1
userType
1
userType
1
<<enum>>
UserType
NONADMINPHYSICIAN
ADMINPHYSICIAN
PATIENT
CLERK

Figure 4.29: UML class diagram of the set of classes required to generate the controllers.

omitted their attributes and methods. The four classes that extend the `UserBasePageControllerGen` class generate the controllers for each user type that are responsible for the web client interactions in the user's home page after login. The `EntityControllerGen` class generates the controller for a given entity according to a user type. This class contains the methods that are responsible for the web client interactions to visualize, edit, add and delete data from an entity. Finally, the `EntityGroupController-Gen` class generates the controller for a given entity group that is responsible for interactions regarding the screen where are displayed all the data of the entities that compose the concerned entity group. In addition, for the `person` entity group controller it is also generated the code that is responsible for the interactions with the data entry form that encompasses all the entities of the `person` entity group (i.e., the data entry form that can be filled after creating a new patient). The `needToGenerate` and `codeTo-Generate` methods of these classes are also used in the same manner as we described for the classes that we presented above.

4.6 Summary

In this chapter, we detailed the CDRGen software system that is used to generate clinical data registries. Specifically, we started by describing its input, output, and the software system modules that parse the input and that generate the clinical data registry's code.

CDRGen accepts two JSON files as input: the JSON file *user interface labels* and the JSON file *specification*. The JSON file *user interface labels* is used to describe the labels of the user interface that are not related with the data to be collected by the clinical data registry. The JSON file *specification* is used to describe the clinical data registry's name, the user types to support and the data to collect. In summary, in order to define the data to collect, we first need to indicate an entity group, then its entities and finally the attributes of each entity. In general, each attribute has a name, a type, and can also have a set of permissions and properties.

CDRGen generates a clinical data registry that is composed of a database, a user interface and the logic that provides the communication between the database and the user interface. In particular, its architecture consists of a MySQL relational database, a web server and a web user interface client. The web user interface client is used by users in order to access and interact with the clinical data registry through a web browser. The database stores the data that is entered by the users in the clinical data registry. The web server is responsible for responding to requests sent by web user interface clients and for storing and retrieving data from the database.

In order for CDRGen generate a clinical data registry, it parses the two JSON files (its input). Subsequently, with the data that is extracted from these files, it generates the database, the web user interface client and the web server of the clinical data registry. In order to generate the database, basically, an entity corresponds to a database table and an entity attribute to a table attribute. To generate the web user interface client, CDRGen creates JSP files that compose the forms for entering and the screens for navigating through the data. Finally, CDRGen also creates Java files in order to compose a Java application that represents the web server. In this way, a clinical data registry is generated.

5 Validation

Contents

5.1 Validation Methodology 99

5.2 Metrics 99

5.3 Testing with PRNC 100

5.4 Validation with PRECISE Stroke 104

5.5 Discussion 110

In this chapter, we describe the assessment of CDRGen against two clinical data registries that were not used during its development. In concrete, we detail the testing of CDRGen against PRNC (described in Section 2.2.5) and its validation against PRECISE Stroke (described in Section 2.3.2). We test CDRGen against PRNC in order to verify if our requirements analysis has been well performed, since PRNC was one of the clinical data registries used during that analysis. We chose these two clinical data registries to assess CDRGen since we know what data is collected and the functionalities that they should provide. These clinical data registries cover two different medical specialties: Cardiology in the case of PRNC and Neurology in the case of PRECISE Stroke. Thus, we end up assessing CDRGen against two medical specialties besides the one that was used for testing CDRGen during its development, i.e., Urology.

In Section 5.2, we describe the evaluation metrics that we used during testing and validation. In Section 5.3, we describe the testing performed with CDRGen against PRNC. In Section 5.4, we describe the validation performed with CDRGen against PRECISE Stroke. Finally, in Section 5.5, we conclude this chapter by discussing the obtained results. Theoretically, we expect better results with the test against PRNC than the validation against PRECISE Stroke, since PRNC was used during the requirements analysis, as we stated earlier.

5.1 Validation Methodology

In order to assess CDRGen against the two existing clinical data registries, we proceeded with the following methodology for each clinical data registry:

1. Identify the *data* that is collected by the existing clinical data registry. For example, PRNC collects the gender of the patient.
2. Write the *specification* of the data that is collected by the existing clinical data registry using the language described in Section 4.2.
3. Identify the *functionalities* that the existing clinical data registry provides. For example, in PRECISE Stroke we can create new patients (i.e., store data about them) and access the corresponding data.

5.2 Metrics

To test and validate CDRGen, we measure the following metrics:

1. **Performance** of the generation process: *time* that the developer (in this case the author, i.e., an expert) spends writing the specification and time that CDRGen takes to generate the clinical data registry. In the case of PRECISE Stroke, the sum of these times is compared to the time that

PRECISE Stroke took to be developed[1]. This value gives an idea of how much time can be saved by using CDRGen. We did not do the same for PRNC, since we do not know how long it took to be developed.

2. **Quality** of the generated clinical data registry, by measuring the *ratio* of:

   - The tables, their attributes and relationships that the generated clinical data registry's database has in relation to what it is expected to have based on the data that is collected by the existing clinical data registry.
   - The functionalities that we can perform through the generated clinical data registry's user interface in relation to the functionalities provided by the existing clinical data registry (i.e., the functionalities that were identified in the third step of the methodology followed).

5.3 Testing with PRNC

Regarding the data collected, for PRNC we identified 5 entity groups, 12 entities and 244 attributes based on the PDF file that describes the data that is collected by PRNC[2]. It took us[3] about 125 minutes (2 hours and 5 minutes) to write a specification file that covers 5 entity groups, 12 entities and 243 attributes[4]. Then, CDRGen took about 2 seconds to generate the clinical data registry based on that specification file.

Table 5.1: Time spent writing the specification for PRNC.

Entity group	**Entities**	**Attributes**	**Time spent (minutes)**
Person	3	23	15
Diagnosis	1	17	15
Treatment	2	56	22
Medical examination	5	138	69
Medical appointment	1	9	4
Total	12	243	125

Table 5.1 shows the time that we spent writing the specification for each entity group. In addition, it also details how many entities and attributes each entity group has. As expected, we can see that it took longer to specify entity groups that have more attributes. Figure 5.1 shows an excerpt of the specification regarding two entities that are part of the `person` entity group: (i) the entity that represents the patient (`Patient` entity); and (ii) the comorbidities entity (`Comorbilidades` entity). Since PRNC is a

[1]We know the development time of PRECISE Stroke because we talked to its developer
[2]https://spc.pt/documents/20143/109424/LISTA+VARIAVEIS+DE+MNC.pdf/39187f40-875d-0135-30af-da5df344fa25
[3]Specifically, it was the author (an expert) who performed the experiment
[4]Note that there is a difference between the number of attributes in the specification file compared to those collected by the existing PRNC since there was one attribute that we were unable to specify. This case is described further on.

Portuguese clinical data registry, the data defined in the specification is in Portuguese. In Annex B we present the full specification.

```
"entity groups": {
    "person": {
        "entities": [
            {   "entity": ["Patient",
                    { "attribute": ["Número Interno", "integer"], "properties": ["identifier"] },
                    { "attribute": ["Sexo", "enum"], "values": ["Masculino", "Feminino"] },
                    { "attribute": ["Raça", "enum+"], "values": ["Caucasiana", "Negra"] },
                    { "attribute": ["Data de nascimento", "date"] },
                    { "attribute": ["Peso", "integer"] },
                    { "attribute": ["Altura", "integer"] },
                    {   "attribute": ["País de nascimento", "enum"],
                        "values": ["Portugal", "Espanha", "França", "Alemanha"]
                    },
                    {   "attribute": ["Tipo de doente", "enum"],
                        "values": ["Índice", "Familiar com fenotipo", "Familiar portador de mutação"]
                    },
                    {   "attribute": ["Há familiares incluídos no registo?", "boolean"] }
                ],
                "label": "Identificação"
            },
            {   "entity": ["Comorbilidades",
                    { "attribute": ["HTA", "boolean"] },
                    { "attribute": ["DM", "boolean"] },
                    { "attribute": ["Doença coronária", "boolean"] },
                    { "attribute": ["Doença valvular", "boolean"] },
                    { "attribute": ["Outra doença cardiaca", "string"] },
                    { "attribute": ["Doença neuromuscular", "boolean"] },
                    { "attribute": ["Doença metabólica", "boolean"] },
                    { "attribute": ["Outra", "string"] }
                ]
            },
            ...
```

Figure 5.1: Excerpt of the JSON file specification for PRNC clinical data registry specifying two entities of the `person` entity group.

In Table 5.2, we compare the database of the generated clinical data registry with the expected database (i.e., the database based on the data collected by the existing PRNC). In concrete, we indicate the ratio of the tables, their attributes and relationships that the generated database has in relation to the expected database. The generated database has 18 tables that correspond to the expected database tables, where:

- 12 represent each of the 12 entities
- 6 represent each of the 6 list-type attributes that were defined in the specification

In addition, 243 of the 244 attributes collected by the existing PRNC appear in the corresponding tables of the generated database. There is 1 attribute that is not present in the generated database because

Table 5.2: The ratio of the tables, their attributes and relationships that the generated clinical data registry's database has in relation to what it is expected to have.

	Database		**Ratio (%)**
	Generated	**Expected**	
Tables	18	18	100%
Attributes	243	244	100%
Relationships	17	17	100%

we could not define it in the specification file. This happened because, in the existing PRNC, there is 1 attribute whose value is valid only if it is a value of another attribute. However, this validation is not supported by CDRGen because we were unable to implement one of the two constraints that existed between attributes[5], which stated that: the value of an attribute must be one of the values that were already given to another attribute.

Finally, as it was supposed to exist in the expected database, in the generated database there are the following 17 relationships between tables:

- 10 one-to-one relationships between the table that represents the patient and the other 10 tables that represent the non-historic entities
- 1 one-to-many relationship between the table that represents the patient and the table that represents the single historic-type entity
- 6 many-to-one relationships between each of the 6 tables that represent a list-type attribute and the corresponding table that represents the entity where that list-type attribute was defined in the specification.

Therefore, we can conclude that the generated database has exactly the tables and relationships between tables that it was supposed to have. In addition, the generated database has 243 of the 244 attributes (i.e., approximately 100%) that it was supposed to have.

Since we do not have access to the PRNC user interface, as far as we can understand, the existing PRNC supports adding, visualizing and editing all the patient data, and viewing statistical data (review Figure 3.9). In other words, PRNC supports adding, visualizing and editing data of all entities of the concerned entity groups, and viewing statistical data. In concrete, PRNC supports the following 16 functionalities:

- Add data to entities from the `person` entity group
- Visualize data of entities from the `person` entity group
- Edit data of entities from the `person` entity group

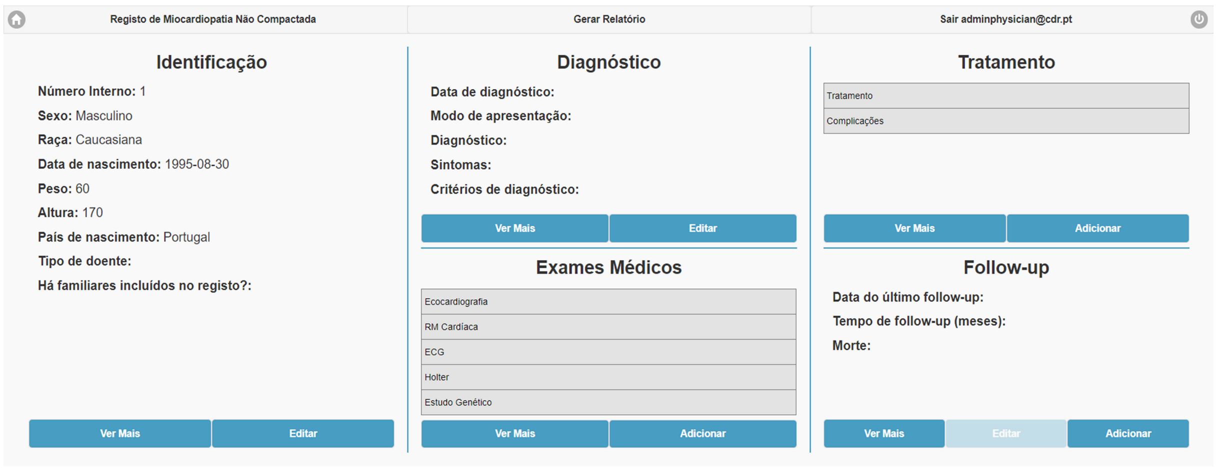
[5]This aspect was not implemented due to time restrictions

- Add data to entities from the `diagnosis` entity group
- Visualize data of entities from the `diagnosis` entity group
- Edit data of entities from the `diagnosis` entity group
- Add data to entities from the `treatment` entity group
- Visualize data of entities from the `treatment` entity group
- Edit data of entities from the `treatment` entity group
- Add data to entities from the `medical examination` entity group
- Visualize data of entities from the `medical examination` entity group
- Edit data of entities from the `medical examination` entity group
- Add data to entities from the `medical appointment` entity group
- Visualize data of entities from the `medical appointment` entity group
- Edit data of entities from the `medical appointment` entity group
- Visualize statistical data.

In the generated clinical data registry, we can perform all these functionalities except visualizing statistical data. Therefore, we can perform 15 of the 16 functionalities of PRNC (i.e., approximately 94%). Note that we cannot visualize statistical data in the generated clinical data registry since this functionality is not supported by the clinical data registries that are generated by CDRGen. This happens since this functionality was not set to be supported when we decided the functionalities of a generated clinical data registry (review Figure 3.10).

Although we do not have access to the existing PRNC user interface, we present some user interface screens of the generated clinical data registry. Figure 5.2 displays the screen where the user has an overview of all data of a patient. Figure 5.3 displays the form that enables the user to edit the instance of the entity that represents the patient (labelled as "Identificação"). Finally, Figure 5.4 displays the screen where the user visualizes all data of the `person` entity group, namely regarding the entity that represents the patient (the tab that is selected).

Given the test performed with the existing PRNC, we can conclude that we achieved almost excellent results. Since PRNC was one of the clinical data registries that we used during the requirements analysis, these results verify that this analysis was well performed. Still, we observed that we had an aspect to integrate into CDRGen related to the single attribute that we could not specify. However, this aspect ended up not to be integrated due to time constraints.


Registo de Miocardiopatia Não Compactada | Gerar Relatório | Sair adminphysician@cdr.pt

Identificação

Número Interno: 1
Sexo: Masculino
Raça: Caucasiana
Data de nascimento: 1995-08-30
Peso: 60
Altura: 170
País de nascimento: Portugal
Tipo de doente:
Há familiares incluídos no registo?:

Diagnóstico

Data de diagnóstico:
Modo de apresentação:
Diagnóstico:
Sintomas:
Critérios de diagnóstico:

Tratamento

Tratamento
Complicações

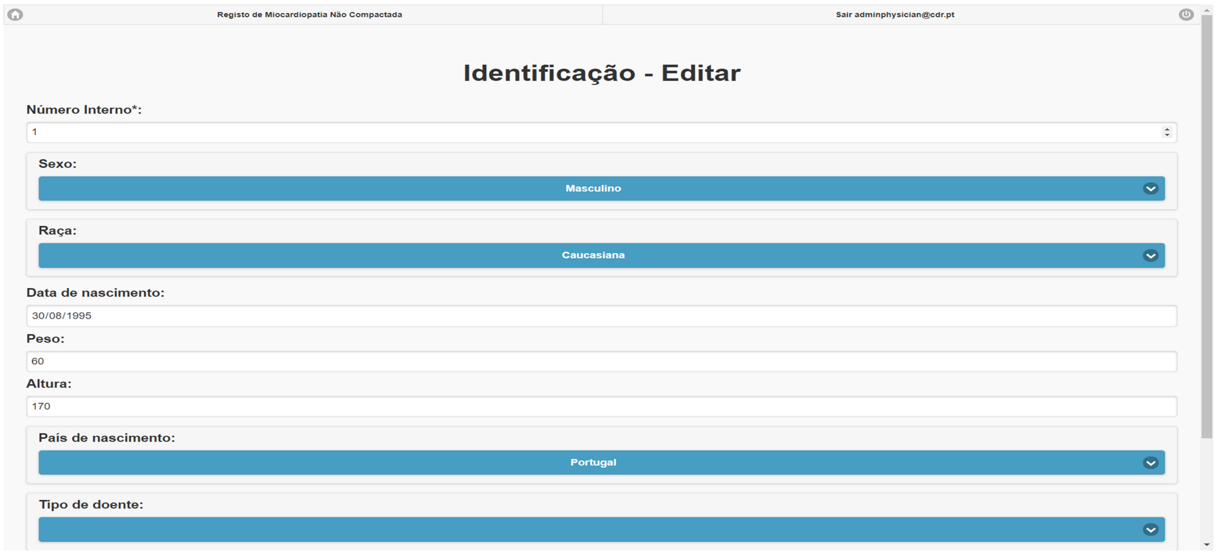
Exames Médicos

Ecocardiografia
RM Cardíaca
ECG
Holter
Estudo Genético

Follow-up

Data do último follow-up:
Tempo de follow-up (meses):
Morte:

Figure 5.2: Screen that displays an overview of all data of a patient in the generated clinical data registry based on PRNC's data.


Registo de Miocardiopatia Não Compactada | Sair adminphysician@cdr.pt

Identificação - Editar

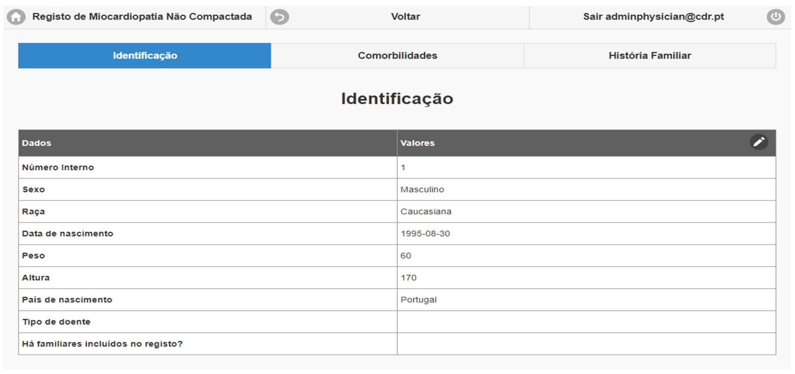
Número Interno*: 1
Sexo: Masculino
Raça: Caucasiana
Data de nascimento: 30/08/1995
Peso: 60
Altura: 170
País de nascimento: Portugal
Tipo de doente:

Figure 5.3: Form to edit the instance of the entity that represents the patient (labelled as "Identificação") in the generated clinical data registry based on PRNC's data.

5.4 Validation with PRECISE Stroke

Regarding the data collected, for PRECISE Stroke we identified 5 entity groups, 25 entities and 722 attributes. Before writing the specification, it took us[6] about 81 minutes (1 hour and 21 minutes) to study the data collected by the existing PRECISE Stroke. In concrete, we identified the entities and we assigned each entity to one entity group. To study the data collected, we took advantage of the test version of the PRECISE Stroke's user interface that is available online[7] and a PDF file that describes all

[6]Specifically, it was the author (an expert) who performed the experiment
[7]http://stroketest.sysresearch.org/login/


Registo de Miocardiopatia Não Compactada
Voltar
Sair adminphysician@cdr.pt
Identificação
Comorbilidades
História Familiar

Identificação

Dados	Valores
Número Interno	1
Sexo	Masculino
Raça	Caucasiana
Data de nascimento	1995-08-30
Peso	60
Altura	170
País de nascimento	Portugal
Tipo de doente	
Há familiares incluídos no registo?	

Figure 5.4: Screen that displays all data of the `person` entity group, in this case regarding the entity that represents the patient (labelled as "Identificação") in the generated clinical data registry based on PRNC's data.

the data that is collected by PRECISE Stroke[8]. Then, it took us about 284 minutes (4 hours and 44 minutes) to write a specification file that covers 5 entity groups, 23 entities and 633 attributes[9]. Afterwards, CDRGen took about 4 seconds to generate the clinical data registry based on that specification file. Table 5.3 shows the time that we spent writing the specification for each entity group. In addition, it also details how many entities and attributes each entity group has. We observe that the time spent writing the specification was not always linear with the number of attributes. This happens because there are quite a few repeated attributes in different entities. Therefore, this simply led us to copy the definition of these attributes to the entities where they appeared, and this saved us time.

Table 5.3: Time spent writing the specification for PRECISE Stroke.

Entity group	**Entities**	**Attributes**	**Time spent (minutes)**
Person	3	73	87
Diagnosis	4	80	33
Treatment	4	77	25
Medical examination	5	97	80
Medical appointment	7	306	59
Total	23	633	284

We talked to the developer of the existing PRECISE Stroke who told us that it took about 300 hours (i.e., 12 days and 12 hours) of work to develop it[10]. When comparing this duration with the time spent generating the clinical data registry using CDRGen, we have a difference of approximately 295 hours (or 293 hours if we include the time that we spent a priori studying the collected data). 295 hours is about 98% of the time spent developing the existing PRECISE Stroke, and that is the time that we could save.

[8]http://stroketest.sysresearch.org/api/public/file/doc/variaveis_PRECISE_STROKE_last_version.pdf

[9]Note that there is a difference between the number of entities and attributes in the specification file compared to those collected by the existing PRECISE Stroke since there were entities and attributes that we were unable to specify. These cases are described further on.

[10]This duration is an estimate of the time spent by the developer only. It does not include the tests and revisions of the PRECISE Stroke user interface that were performed by the users

Figure 5.5 shows an excerpt of the specification regarding one entity of the `person` entity group: the entity that represents the patient (`Patient` entity). Since PRECISE Stroke is a Portuguese clinical data registry, the data defined in the specification is in Portuguese. In Annex C we present the full specification.

```
"entity groups": {
    "person": {
        "entities": [
            {   "entity": [
                    "Patient",
                    {   "attribute": ["ID Processo Local", "string"],
                        "properties": ["identifier"] },
                    {   "attribute": ["Iniciais Nome", "string"],
                        "properties": ["not null", "searchable"] },
                    {   "attribute": ["Hospital", "string"],
                        "properties": ["not null"] },
                    {   "attribute": ["Altura (cm)", "integer"],
                        "min": "0"
                    },
                    {   "attribute": ["Peso (kg)", "float"],
                        "min": "0"
                    },
                    { "attribute": ["Data de Nascimento", "date"] },
                    { "attribute": ["Profissão", "string"] },
                    {   "attribute": ["Género", "enum"],
                        "values": ["Feminino", "Masculino"],
                        "properties": ["not null", "searchable"] },
                    {   "attribute": ["Vive Sozinho", "enum"],
                        "values": ["Desconhecido", "Não", "Sim"],
                        "properties": ["not null", "searchable"] },
                    {   "attribute": ["Etnia", "enum"],
                        "values": [ "Caucasiana", "Latino Americano",
                                    "Norte Africano", "Africano Negro",
                                    "Asiático de Este", "Asiático Oeste"],
                        "properties": ["not null", "searchable"] },
                    {   "attribute": ["Grau de Escolaridade", "enum"],
                        "values": [ "Desconhecido", "Analfabeto",
                                    "1 a 4 anos de escolaridade",
                                    "4 anos a 12 anos de escolaridade",
                                    "Curso superior" ],
                        "properties": ["not null", "searchable"] }

                ],
                "label": "Geral"
            },
            ...
```

Figure 5.5: Excerpt of the JSON file specification for PRECISE Stroke clinical data registry specifying one entity of the `person` entity group (the entity that represents the patient).

In Table 5.4, we compare the database of the generated clinical data registry with the expected database (i.e., the database based on the data collected by the existing PRECISE Stroke). In concrete, we indicate the ratio of the tables, their attributes and relationships that the generated database has in relation to the expected database. The generated database has 65 tables:

- 23 to represent each of the 23 entities
- 42 to represent each of the 42 list-type attributes that were defined in the specification

Table 5.4: The ratio of the tables, their attributes and relationships that the generated clinical data registry's database has in relation to what it is expected to have.

	Database		**Ratio (%)**
	Generated	**Expected**	
Tables	65	67	97%
Attributes	633	722	88%
Relationships	64	70	91%

The generated database lacks two tables from the expected database because we were unable to specify the two historic-type entities that appear in the data collected by the existing PRECISE Stroke. This happens because these two entities do not have a historical attribute defined explicitly. In particular, the user can enter two (or more) equal instances in one of these entities and both are stored. For example, the user can add two chemical biomarkers with the same characteristics (i.e., add two chemical biomarkers whose attribute values are the same). In CDRGen this is not supported since a historic-type entity has a historical date-type attribute that identifies the entity instance and, therefore, cannot be repeated. As a result, the generated database does not have the two tables that represent each historic-type entity. Therefore, the generated database has 65 of the 67 tables that it should have (i.e., approximately 97%).

Regarding attributes, the generated database has 633 attributes in their corresponding tables. The generated database is missing 89 attributes from the expected database because we were unable to specify them. In particular:

- 21 attributes compose the historic-type entities that we were unable to specify due to their historical attributes (as we stated above). If we were able to specify the historical attributes, then we could specify these 21 attributes since they are supported by CDRGen.

- 31 attributes are automatic computed fields (i.e., attributes that are calculated based on formulas that use the values of other attributes). For example, the patient age can be computed from her date of birth. CDRGen does not support this attribute type. Note that the fact that these attributes are automatically calculated implies that the user does not have to enter their value. Thus, these attributes exist only for the user to visualize their value in the user interface or for further data analysis purposes. Visualizing the value of these attributes in the user interface is only really important if the user needs to make decisions based on these values (for example, during a medical appointment). On the other hand, if the user never needs to know these values (for example, if they are used only for aggregate data analysis), the need to display the values of these attributes in the user interface turns out to be unimportant. In this case, it should be noted that these values could only be computed when data analysis was performed. As we seen in the Portuguese clinical data registries, the data analysis is often performed externally to the clinical data registries and this implies, in this case, that these attributes would not even need to be stored by the clinical data registries.

- 7 attributes are of the time type. CDRGen does not support this attribute type. A simple alternative solution could be to specify these attributes as string-type attributes. However, this solution would have the disadvantage that the user could enter an invalid value, i.e., a value that do not respect the time format.

- 18 attributes are of the datetime type. CDRGen does not support this attribute type. A simple alternative solution could be to split these attributes into two: a date-type attribute and a time-type attribute. However, once again, the time-type attribute would need to be specified as a string-type attribute, which would consequently imply the disadvantage that we stated above.

- 4 attributes are of the image list type, in which each image has 2 other attributes that detail it. Therefore, this encompasses 12 attributes (i.e., 4+4*2). CDRGen does not support attributes of the image list type and, consequently, the attributes that detail each image were not specified either.

Therefore, the generated database has 633 of the 722 attributes that it should have (i.e., approximately 88%).

Finally, in the generated database there are the following 64 relationships between tables:

- 22 one-to-one relationships between the table that represents the patient and the other 22 tables that represent the other entities

- 42 many-to-one relationships between each of the 42 tables that represent a list-type attribute and the corresponding table that represents the entity where that list-type attribute was defined in the specification.

The generated database is missing 6 relationships between tables from the expected database. 2 of these 6 relationships are between the table that represents the patient and the 2 tables that represent the historic-type entities that we were unable to specify. Since the database does not have the tables that represent each historic-type entity, consequently, there is no relationship between each of these tables and the table that represents the patient. The other 4 of the 6 relationships are between the 4 tables that represent each attribute of the image list type that we also were unable to specify. Consequently, there are also no relationship between each of the tables that represent these image list-type attributes and the corresponding table that represents the entity where these attributes were supposed to be defined in the specification. Therefore, we conclude that the generated database has 64 of the 70 relationships between tables that it should have (i.e., approximately 91%).

The existing PRECISE Stroke supports the following 18 functionalities:

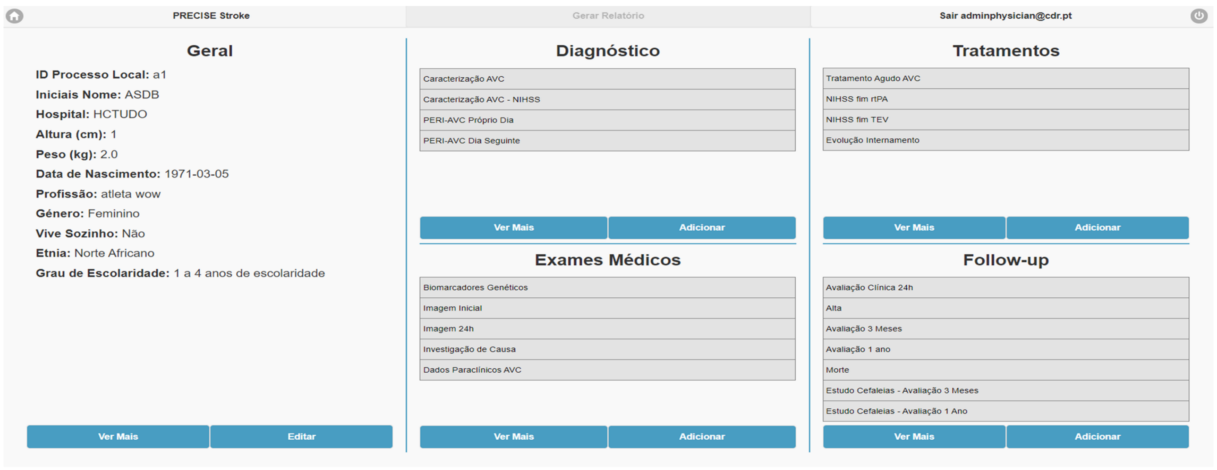
- Add data to entities from the `person` entity group
- Visualize data of entities from the `person` entity group
- Edit data of entities from the `person` entity group
- Add data to entities from the `diagnosis` entity group
- Visualize data of entities from the `diagnosis` entity group
- Edit data of entities from the `diagnosis` entity group
- Add data to entities from the `treatment` entity group
- Visualize data of entities from the `treatment` entity group
- Edit data of entities from the `treatment` entity group
- Add data to entities from the `medical examination` entity group
- Visualize data of entities from the `medical examination` entity group
- Edit data of entities from the `medical examination` entity group
- Add data to entities from the `medical appointment` entity group
- Visualize data of entities from the `medical appointment` entity group
- Edit data of entities from the `medical appointment` entity group
- Export all data of a set of patients
- Delete all data of a patient
- Search for patients by attributes of the entity that represents the patient.

In the generated clinical data registry, we can perform all of these functionalities except two: (i) export all data of a set of patients; and (ii) delete all data of a patient. Therefore, we can perform 16 of the 18 functionalities of PRECISE Stroke (i.e., approximately 89%). Although we cannot export all data of a set of patients, the generated clinical data registry is able to export the data that has changed (i.e., that has been added, edited or deleted) in relation to a patient. In addition, although we cannot delete all data of a patient, in the generated clinical data registry is possible to delete the instances of historic-type entities.

Although we have not performed usability tests due to time restrictions[11], we present some user interface screens of the generate clinical data registry. Figure 5.6 displays the screen where the user

[11] This would require more time and considerable effort in the context of this thesis. This aspect was therefore not considered since, in general terms, the demonstration of the concept was already done. Even so, we discuss this aspect as future work in Chapter 6.

has an overview of all data of a patient. The existing PRECISE Stroke does not have a screen like this, so we cannot compare them. On the other hand, Figures 5.7 and 5.8 display the form that can be used



Geral

ID Processo Local: a1
Iniciais Nome: ASDB
Hospital: HCTUDO
Altura (cm): 1
Peso (kg): 2.0
Data de Nascimento: 1971-03-05
Profissão: atleta wow
Género: Feminino
Vive Sozinho: Não
Etnia: Norte Africano
Grau de Escolaridade: 1 a 4 anos de escolaridade

Diagnóstico

- Caracterização AVC
- Caracterização AVC - NIHSS
- PERI-AVC Próprio Dia
- PERI-AVC Dia Seguinte

Tratamentos

- Tratamento Agudo AVC
- NIHSS fim rtPA
- NIHSS fim TEV
- Evolução Internamento

Exames Médicos

- Biomarcadores Genéticos
- Imagem Inicial
- Imagem 24h
- Investigação de Causa
- Dados Paraclínicos AVC

Follow-up

- Avaliação Clínica 24h
- Alta
- Avaliação 3 Meses
- Avaliação 1 ano
- Morte
- Estudo Cefaleias - Avaliação 3 Meses
- Estudo Cefaleias - Avaliação 1 Ano

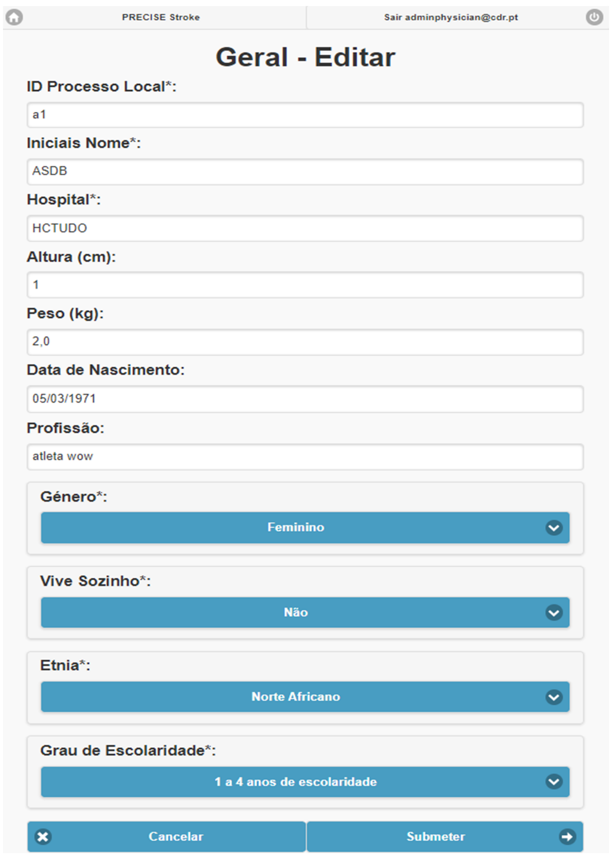
Figure 5.6: Screen that displays an overview of all data of a patient in the generated clinical data registry based on PRECISE Stroke's data.

to edit the instance of the entity that represents the patient in the generated clinical data registry and in the existing PRECISE Stroke test version, respectively. As expected, both forms have the same content (i.e., contain the same pairs of labels and input fields). Finally, Figure 5.9 displays the screen where the user visualizes all data of the `person` entity group, namely regarding the entity that represents the patient (the tab that is selected) that is labelled as "Geral". Once more, the existing PRECISE Stroke does not have a screen like this given the fact that the user visualizes the data in the same screen where she adds and edits it (i.e., the screen that is displayed in Figure 5.8).

5.5 Discussion

In this Chapter, we validated CDRGen against two existing clinical data registries: PRNC and PRECISE Stroke.

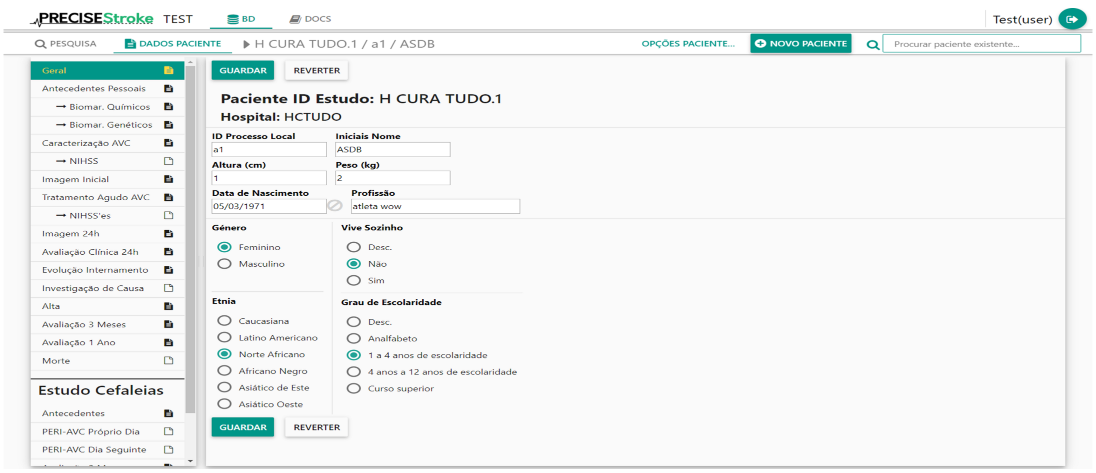
For PRNC, the author (i.e., an expert) spent about 125 minutes (2 hours and 5 minutes) to write a specification that covered 5 entity groups, 12 entities and 243 attributes. Then, CDRGen took about 2 seconds to generate the clinical data registry based on that specification. For PRECISE Stroke, the author spent about 284 minutes (4 hours and 44 minutes) to write a specification that covered 5 entity groups, 23 entities and 633 attributes. Then, CDRGen took about 4 seconds to generate the clinical data registry based on that specification. Given this, we observe that the time for CDRGen to generate a clinical data registry is irrelevant in this case because it is in the order of magnitude of seconds. In addition, we observe that the time to write the specification based on the PRECISE Stroke's data was less compared to the time to write the specification based on the PRNC's data with respect to the number



Geral - Editar

ID Processo Local*:	a1
Iniciais Nome*:	ASDB
Hospital*:	HCTUDO
Altura (cm):	1
Peso (kg):	2,0
Data de Nascimento:	05/03/1971
Profissão:	atleta wow
Género*:	Feminino
Vive Sozinho*:	Não
Etnia*:	Norte Africano
Grau de Escolaridade*:	1 a 4 anos de escolaridade

Figure 5.7: Form to edit the instance of the entity that represents the patient (labelled as "Geral") in the generated clinical data registry based on PRECISE Stroke's data.



H CURA TUDO.1 / a1 / ASDB

Paciente ID Estudo: H CURA TUDO.1
Hospital: HCTUDO

ID Processo Local	a1
Iniciais Nome	ASDB
Altura (cm)	1
Peso (kg)	2
Data de Nascimento	05/03/1971
Profissão	atleta wow

Género: [✓ Feminino] Masculino

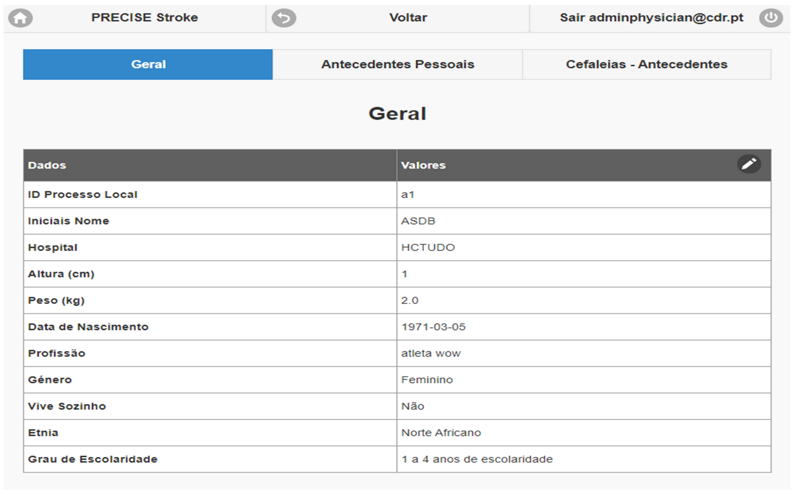
Vive Sozinho: Desc. [✓ Não] Sim

Etnia: Caucasiana, Latino Americano, [✓ Norte Africano], Africano Negro, Asiático de Este, Asiático Oeste

Grau de Escolaridade: Desc., Analfabeto, [✓ 1 a 4 anos de escolaridade], 4 anos a 12 anos de escolaridade, Curso superior

Figure 5.8: Form to edit the instance of the entity that represents the patient (labelled as "Geral") in the existing PRECISE Stroke test version.

of attributes that were specified in both cases (633 versus 243 attributes). This happened because many attributes of PRECISE Stroke are repeated in different entities. This repetition simply led us to copy the definition of these attributes to the entities in which they appeared. However, we needed to study the



Geral

Dados	Valores
ID Processo Local	a1
Iniciais Nome	ASDB
Hospital	HCTUDO
Altura (cm)	1
Peso (kg)	2.0
Data de Nascimento	1971-03-05
Profissão	atleta wow
Género	Feminino
Vive Sozinho	Não
Etnia	Norte Africano
Grau de Escolaridade	1 a 4 anos de escolaridade

Figure 5.9: Screen that displays all data of the `person` entity group, in this case regarding the entity that represents the patient (labelled as "Geral") in the generated clinical data registry based on PRECISE Stroke's data.

data collected by PRECISE Stroke before writing its specification, which led to spending an extra 81 minutes a priori. Therefore, we ended up spending 365 (284 + 81) minutes writing the PRECISE Stroke specification if we include the time to study its data. On the other hand, we did not need to study the data collected by PRNC since this is one of the clinical data registries that we use as a basis when we perform the requirements analysis. This analysis led us to already know which entity groups and entities we needed to define when writing the specification for PRNC, which saved us time.

For PRNC, CDRGen was able to generate a database that had all the tables and relationships between tables that it was supposed to have. Only 1 of the 244 attributes collected by the existing PRNC (i.e., approximately 100%) was not included in the generated database. Since PRNC was used as a basis when we performed the requirements analysis, the results of the generated database shows that the requirements analysis performed regarding the data collected was almost excellent. On the other hand, we have not achieved as good results as these with PRECISE Stroke with respect to the generated database. Since PRECISE Stroke was not used in the requirements analysis, we were completely unaware of the data that it collected. Still, the database generated based on the data collected by the existing PRECISE Stroke had approximately 97% of the tables, 88% of the attributes and 91% of the relationships between tables that it was supposed to have.

In fact, the existing PRECISE Stroke collects data of types that CDRGen does not support, such as datetime, time, and image lists where each image has attributes that detail it. These 3 data types do not appear in any of the clinical data registries that were used as basis when we performed the requirements analysis, and ended up not to be included when we developed the CDRGen. In addition, the existing PRECISE Stroke also collected automatic computed fields, which is also not supported by CDRGen. Finally, we observed that the historic-type entities that exist in PRECISE Stroke are also not supported by CDRGen.

The datetime and time types are easy to include in the attribute types supported by CDRGen. In fact, after we performed the validation of CDRGen with PRECISE Stroke, we added two subclasses (`AttributeTypeDateTime` and `AttributeTypeTime`) to the `AttributeType` class (review Figure 4.14) and CDRGen started to support these two attribute types. We, once more, generated a clinical data registry with a specification based on the data collected by the existing PRECISE Stroke that already covered these two attributes types. This time, the generated database had approximately 97% of the tables (same as before), 91% of the attributes (instead of 88%) and 91% of the relationships between tables (same as before) that it was supposed to have.

Regarding the functionalities of the generated clinical data registries, we observed that the user is able to perform about 94% and 89% of the functionalities of the existing PRNC and PRECISE Stroke, respectively.

Overall, as expected, we observe that we achieve a greater success with PRNC than with PRECISE Stroke, since PRNC was one of the clinical data registries that we used as a basis when we perform the requirements analysis. However, we also observe that we can generate a clinical data registry that support much of the existing PRECISE Stroke (a clinical data registry that we barely knew about its data and requirements). In addition, in both cases, we observe that the time required to generate a clinical data registry when using CDRGen is on the order of magnitude of hours if the developer is an expert in the specification language. Even if the developer does not know the specification language at all, the time required for him to learn it has little influence on the order of magnitude of the time required to generate a clinical data registry. Possibly in the worst case, it may reach the order of magnitude of days. Still, the time required to generate a clinical data registry without using CDRGen (i.e., what currently happens) is much longer than this. In fact, the existing PRECISE Stroke took about 300 hours (12 days and 12 hours) of work to develop. Since 300 hours exceed one week and not one month, this duration reaches the order of magnitude of weeks.

6 Conclusions and Future Work

Contents

6.1 Conclusions 117
6.2 Limitations and Future Work 117

6.1 Conclusions

In this thesis, we presented a Clinical Data Registry Generator (CDRGen) software system to generate clinical data registries with a minimum effort in terms of software design and development. A clinical data registry is generated from two JSON files: one that describes, in a high-level language, the characteristics of the data that needs to be collected by the clinical data registry; and another that describes the user interface labels that are unrelated to the data to be collected by the clinical data registry. The syntax used in both JSON files was developed in the context of this work, and described in Sections 4.2 and 4.3.

Throughout the development of CDRGen, we used one existing clinical data registry: Umedicine for Urology. In this way, in conjunction with the requirements analysis that we performed based on the existing Portuguese clinical data registries, we sought to develop CDRGen to be generic enough to generate any clinical data registry.

To assess CDRGen, we used another two existing clinical data registries: the PRNC for Cardiology and the PRECISE Stroke for Neurology. PRNC was used to test and the PRECISE Stroke was used to validate. When testing with PRNC, we were able to generate a database that had 100% of the tables, 100% of the relationships between tables and approximately 100% of the attributes that it was supposed to have. In addition, the generated clinical data registry supported approximately 94% of the functionalities that the existing PRNC provides. These results confirmed that our requirements analysis was well performed. When validating with PRECISE Stroke, we were able to generate a database that had approximately 97% of tables, 88% of attributes and 91% of the relationships between tables that it was supposed to have. In addition, the generated clinical data registry supported approximately 89% of the functionalities that the existing PRECISE Stroke provides. Finally, during this assessment process, we observed that by using CDRGen, we can generate clinical data registries in just a few hours, as opposed to the standard procedure which can take weeks.

6.2 Limitations and Future Work

The limitations and the future work aspects of this thesis are separated in three groups.

In the first group, we detail the aspects that ended up not being implemented in CDRGen as it was supposed to. These aspects were not implemented due to time restrictions. Specifically, we did not implement:

- The attribute constraint that states that the value of an attribute must be one of the values that were already given to another attribute (as it happens with a foreign key in the relational model). The fact that this is not implemented may lead to the concerned attributes to hold invalid values.

One easy way to implement this is through a foreign key, something that is supported by a MySQL relational database. Basically, the attribute whose value must be one of the values that has already been given to another attribute references that same attribute in the database through a foreign key.

- The relationships between some specific entities (such as medicine and disease) as we identified in Umedicine during the requirements analysis. As a consequence, the user interface can display more information that is supposed to. For example, it may be the case that a given medicine only makes sense to be prescribed for a given disease. If a patient does not have that disease, it makes no sense for a physician user to be able to prescribe that medicine. This is what happens if we do not have a relationship between the medicine and disease entities. In this case, the user interface shows all medicines regardless of the disease of the concerned patient. One possible solution to implement this could be to define keywords in the specification that encapsulated the semantics of each of these entities (as happens with the entity that represents the patient, that must be specifically defined with the `Patient` string in the specification file).

In the second group, we detail the limitations that were identified during the evaluation of CDRGen. These limitations correspond to aspects that CDRGen does not support. This is an important factor to bear in mind, since the fact that certain aspects are not supported may be the deciding factor for using a clinical data registry generated by CDRGen. If CDRGen is unable to meet the users's requirements, they may prefer to wait for a clinical data registry to be developed from scratch in order to satisfy their requirements. As such, part of the future work would be to implement what is not currently supported. In particular, CDRGen does not support:

- Attributes of the image list type in which each image has other attributes that detail it. The image list type is easy to include in the attribute types supported by CDRGen, as we did with the datetime and time types during the validation against PRECISE Stroke. However, the fact that each image has attributes that detail it would lead to more substantial code changes. In concrete:
  - The specification would have to support a set of multiple sub-attributes for list-type attributes
  - In the database, a table would be needed to represent each list-type attribute and the attributes of that table would be the sub-attributes
  - In the user interface, the set of sub-attributes would have to be displayed each time the user wanted to add a new element to the list. In addition, the screens where the user can visualize an entity instance would have to be able to show all elements of that list concerning that instance.

  Note that currently list-type attributes are supported only as single elements (i.e., consisting of only one attribute), and not as many as for image lists that appear in the existing PRECISE Stroke.

- Automatic computed fields. To support this, we would also need to make some changes to the code. For the most generic cases of an automatic computed field, a simple solution could be to define keywords in the specification. For example, the patient's age (which is computed from the patient's birth date) could be defined, in the specification, as an automatic computed field that had the *AGE* keyword associated. In addition, the *BIRTH DATE* keyword would have to be associated with the attribute that represents the patient's birth date. This way, the system would know that the patient's age is an automatic computed field (concretely, the age) that is calculated from the value of the attribute that has the *BIRTH DATE* keyword associated (i.e., the attribute that represents the patient's birth date). However, this solution would only work for automatic computed fields whose calculation formula was already predefined in the system. Another simple solution would be to indicate the automatic computed fields in the specification and let the developer define the code that calculates the value of each field. In concrete, in the web server of the generated clinical data registry, a Java class would be generated with an empty method for each automatic computed field defined in the specification. Each method would receive an instance of the class that represented the entity where the automatic computed field was defined. Then, the developer would need to define the code of a method given the attribute values of the entity instance received in order to return the value of the automatic computed field. This solution would require the developer to have programming skills beyond the knowledge of the specification language. However, defining the value of an automatic computed field would not be easy to do in the JSON file specification in a more formal way. Unlike the previous solution, this solution can be used to define the value of any automatic computed field of one entity based on the attribute values of that entity.

- Historic-type entities whose attribute that identifies each entity instance (i.e., the historical attribute) is not an attribute that is collected by the clinical data registry explicitly[1]. For CDRGen to support this, in the specification a historic attribute may or may not be defined in the historic-type entities. In the case the historic attribute is not defined, each instance would have to be identified by an internal identifier (unknown by the user). The simplest solution would be for this internal identifier to be an integer that was managed internally. This integer would be automatically incremented each time a new instance was added to the entity and would be used to sort the entity instances on the user interface screen where they are displayed.

- The functionality of visualizing statistical data. This is a functionality that is specific from clinical data registry to clinical data registry. In particular, to support this functionality it would be necessary to support a way to define the statistical data to obtain. One way to do this could be by specifying queries as we do when we query databases. Then, in the specification we could have an extra section where these queries would be defined. For example, suppose that we wanted to know how

[1]This means that this attribute value is not entered by the user, but is only managed internally.

many patients existed in the clinical data registry. For this, it could be defined in the specification that we wanted the number of instances of the entity that represents the patient.

- The functionality of exporting all data of a given set of patients. To support this functionality it would be necessary to query all tables in the database. From each table, the data of the concerning patients would be extracted and afterwards inserted in a file according to a predefined template. In this manner, this would only lead to adding extra code (i.e., it would not imply making changes to the current code).

- The functionality that enables to delete all data of a patient. To support this functionality, it would also be necessary to query all tables in the database. From each table, we would have to remove all instances that were related with the concerned patient. An easier way to do this would be to create the database tables with the "ON DELETE CASCADE" option (supported by MySQL relational databases). This way, we would only need to delete an instance of the entity that represents the patient and all instances of the other tables that reference this patient would be also deleted. Therefore, a patient would be deleted.

Finally, in the third group, we detail other limitations that we consider. Namely:

- The user interacts with the generated clinical data registries always in the same way, since the user interface always has the same aspect and characteristics. This can be problematic if the user interface does not have good usability. However, the user interface of a clinical data registry that is generated by CDRGen was built based on the Umedicine user interface. In addition, the Umedicine user interface was built aiming at having good usability. Even so, measuring usability metrics is crucial to be held in the future in order to compare, for instance, the generated user interface with the user interface of the existing clinical data registries that were used to evaluate CDRGen[2]. Note, however, that generating clinical data registries with a non-uniform (i.e., customizable) user interface leads to a huge price that the CDRGen's developer needs to pay with respect to the code required to do it. In fact, the more generic the clinical data registry to generate, the more complex the code required to do it.

- The maintenance of the generated clinical data registry. Since the user requirements are always changing over time, it is important to keep in mind that, for instance, the data to be collected by a clinical data registry may change. As such, it would be important to consider how CDRGen could support such changes after the generation of a clinical data registry for the first time (i.e., its first version). A simple solution could be to store the last JSON file specification that was used to generate the clinical data registry. Then, the developer could modify a copy of this JSON file,

[2]Unfortunately, this was not evaluated under the scope of this thesis due to time restrictions.

making changes on the data to be collected (for instance, adding, editing and removing attributes). Finally, CDRGen could use the modified JSON file to generate the new code of the web user interface client and the web server (as currently happens). Note that we can discard all the old code of the web user interface client and the web server, because they do not store the data of the clinical data registry (only the database does that). Regarding the database, in order to retain the data that already existed, CDRGen could use both JSON files to detect the modifications and to generate only the SQL commands that enable to transform the structure of the former database version into the new one.

- The way the developer specifies the data to be collected by the clinical data registry to be generated. Although the developer specifies the data to be collected in a high-level language, he must follow a verbose syntax that is described in the JSON format. As such, part of future work would be to develop a user interface that allowed the developer to specify the data to be collected by the clinical data registry to be generated. Thus, it would no longer be necessary to describe this data using the JSON format, but rather by using a user interface. Consequently, this would diminish the knowledge and skills required to specify the data to be collected and thus give physicians the opportunity and the means to develop and generate clinical data registries by themselves.

Bibliography

[1] D. Brito, N. Cardim, L. Rocha-Lopes, A. Freitas, A. Pais De Lacerda, M. Menezes, A. Belo, E. Martins, M. Peres, L. Goncalves, J. Mimoso, and On behalf of Portuguese Myocarditis Registry Investigators. Diagnosis and treatment of acute myocarditis in Portugal. Data from the national multicenter registry on myocarditis. *European Heart Journal*, 38(1):742–3, August 2017.

[2] Dulce Brito, Nuno Cardim, Luís Rocha Lopes, Adriana Belo, Jorge Mimoso, Lino Gonçalves, and Hugo Madeira. Awareness of Fabry disease in cardiology: A gap to be filled. *Revista Portuguesa de Cardiologia*, 37(6):457–466, June 2018.

[3] Helena Canhão, Augusto Faustino, Fernando Martins, João Eurico Fonseca, Patrícia Nero, and Jaime C. Branco. Reuma.pt - the rheumatic diseases portuguese register. *Acta Reumatologica Portuguesa*, 36:45–56, January 2011.

[4] Nuno Cardim, Dulce Brito, Luís Rocha Lopes, António Freitas, Carla Araújo, Adriana Belo, Lino Gonçalves, Jorge Mimoso, Iacopo Olivotto, Perry Elliott, and Hugo Madeira. The Portuguese Registry of Hypertrophic Cardiomyopathy: Overall results. *Revista Portuguesa de Cardiologia*, 37(1):1–10, January 2018.

[5] Nuno Cardim, António Freitas, and Dulce Brito. From hypertrophic cardiomyopathy centers to inherited cardiovascular disease centers in Europe. A small or a major step? A position paper from the Nucleus of the Working Group on Myocardial and Pericardial Diseases of the Portuguese Society of Cardiology. *Revista Portuguesa de Cardiologia*, 30(11):829–835, November 2011.

[6] Augusto Faustino. Reuma.pt - the start and the purpose. *Acta Reumatologica Portuguesa*, 43(1):6–7, January 2018.

[7] Ana L. P. A. Ferreira. Psoriasis in the elderly: observational study from DERMA.PT registry. Master's thesis, Faculdade de Medicina da Universidade do Porto, March 2017.

[8] M. Rachel Flynn, Conor Barrett, Francisco G. Cosío, Anselm K. Gitt, Lars Wallentin, Peter Kearney, Moira Lonergan, Emer Shelley, and Maarten L. Simoons. The Cardiology Audit and Registration

Data Standards (CARDS), European data standards for clinical cardiology practice. *European Heart Journal*, 26(3):308–313, 2005.

[9] Nuno F. Lages, Bernardo Caetano, Manuel J. Fonseca, João D. Pereira, Helena Galhardas, and Rui Farinha. Umedicine: A System for Clinical Practice Support and Data Analysis. In *Data Management and Analytics for Medicine and Healthcare (DMAH), VLDB Workshop*, pages 102–120, 2017.

[10] Nuno F. G. Lages. Umedicine: A System for Clinical Practice Support and Data Analysis. Master's thesis, Instituto Superior Técnico, May 2017.

[11] Conceição Outeirinho. Registos clínicos: pedra angular na qualidade da prestação de cuidados de saúde. *Revista Portuguesa de Medicina Geral e Familiar*, 34:6 – 8, 2018.

[12] Hélder Pereira, Daniel Caldeira, Rui Campante Teles, Marco Costa, Pedro Canas da Silva, Vasco da Gama Ribeiro, Vítor Brandão, Dinis Martins, Fernando Matias, Francisco Pereira-Machado, José Baptista, Pedro Farto e. Abreu, Ricardo Santos, António Drummond, Henrique Cyrne de Carvalho, João Calisto, João Carlos Silva, João Luís Pipa, Jorge Marques, Paulino Sousa, Renato Fernandes, Rui Cruz Ferreira, Sousa Ramos, Eduardo Infante Oliveira, Manuel de Sousa Almeida, and on behalf of the investigators of Portuguese Registry on Interventional Cardiology (Registo Nacional de Cardiologia de Intervenção). Thrombus aspiration in patients with ST-elevation myocardial infarction: results of a national registry of interventional cardiology. *BMC Cardiovascular Disorders*, 18(1):69, April 2018.

[13] Hélder Pereira, Rui Campante Teles, Marco Costa, Pedro Canas da Silva, Vasco da Gama Ribeiro, Vítor Brandão, Dinis Martins, Fernando Matias, Francisco Pereira-Machado, José Baptista, Pedro Farto e Abreu, Ricardo Santos, António Drummond, Henrique Cyrne de Carvalho, João Calisto, João Carlos Silva, João Luís Pipa, Jorge Marques, Paulino Sousa, Renato Fernandes, Rui Cruz Ferreira, Sousa Ramos, Eduardo Oliveira, and Manuel Almeida. Angioplastia primária em Portugal entre 2002-2013. Atividade segundo o Registo Nacional de Cardiologia de Intervenção. *Revista Portuguesa de Cardiologia*, 35(7):395–404, January 2016.

[14] Hélder Pereira, Rui Campante Teles, Marco Costa, Pedro Canas da Silva, Rui Cruz Ferreira, Vasco da Gama Ribeiro, Ricardo Santos, Pedro Farto e Abreu, Henrique Cyrne de Carvalho, Jorge Marques, Renato Fernandes, Vítor Brandão, Dinis Martins, António Drummond, João Luís Pipa, Luís Seca, João Calisto, José Baptista, Fernando Matias, José Sousa Ramos, Francisco Pereira-Machado, João Carlos Silva, and Manuel Almeida. Evolução da intervenção coronária percutânea entre 2004-2013. Atividade em Portugal segundo o Registo Nacional de Cardiologia de Intervenção. *Revista Portuguesa de Cardiologia*, 34(11):673–681, October 2015.

[15] M. J. Santos, H. Canhão, A. F. Mourão, F. Oliveira Ramos, C. Ponte, C. Duarte, A. Barcelos, F. Martins, and J. A. Melo Gomes. Reuma.pt contribution to the knowledge of immune-mediated systemic rheumatic diseases. *Acta Reumatologica Portuguesa*, 42:232–239, July 2017.

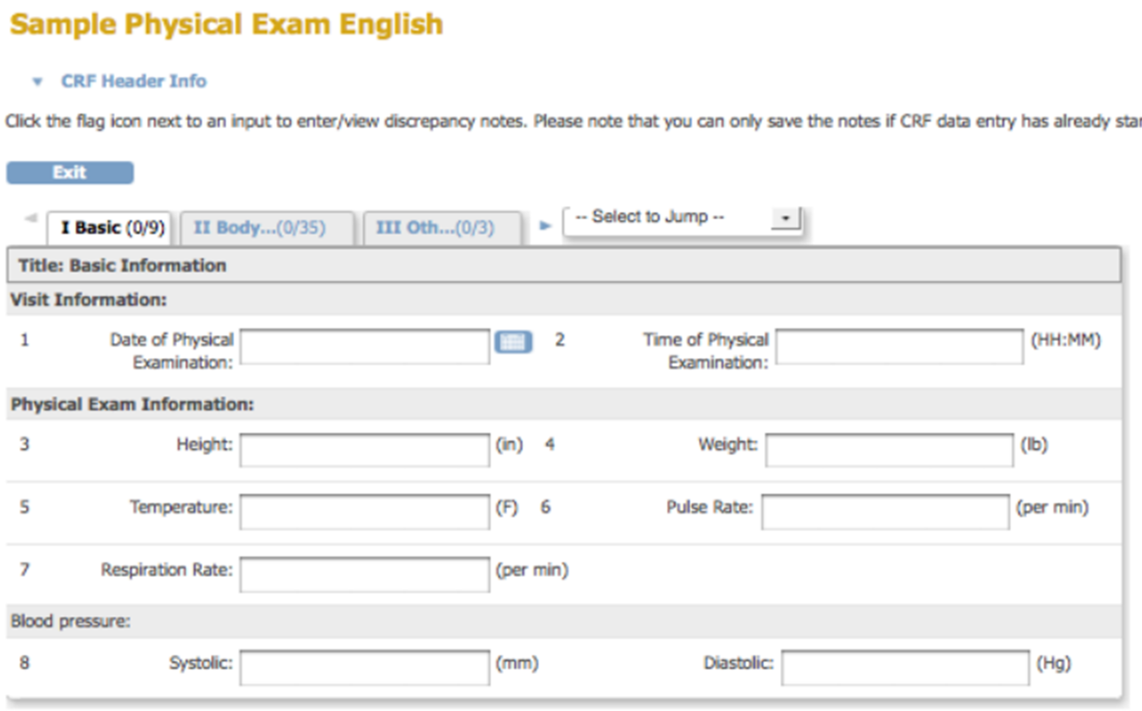
[16] M. J. Santos and J. Canas da Silva. Reuma.pt - structure and innovation. *Acta Reumatologica Portuguesa*, 43(1):8–9, January 2018.

[17] Maria José Santos, Helena Canhão, Augusto Faustino, and João Eurico Fonseca. Reuma.pt: A Case Study. *Acta médica portuguesa*, 29(2):83–4, February 2016.

[18] Ana Teresa Timóteo and Jorge Mimoso. Registo Nacional de Síndromes Coronárias Agudas: 15 anos de um registo prospetivo contínuo. *Revista Portuguesa de Cardiologia*, 37(7):563–573, June 2018.

A

Example of a Case Report Form (CRF)


Sample Physical Exam English

CRF Header Info

Click the flag icon next to an input to enter/view discrepancy notes. Please note that you can only save the notes if CRF data entry has already started.

Title: Basic Information

Visit Information:

1	Date of Physical Examination:		2	Time of Physical Examination:		(HH:MM)

Physical Exam Information:

3	Height:	(in)	4	Weight:		(lb)
5	Temperature:	(F)	6	Pulse Rate:		(per min)
7	Respiration Rate:	(per min)				

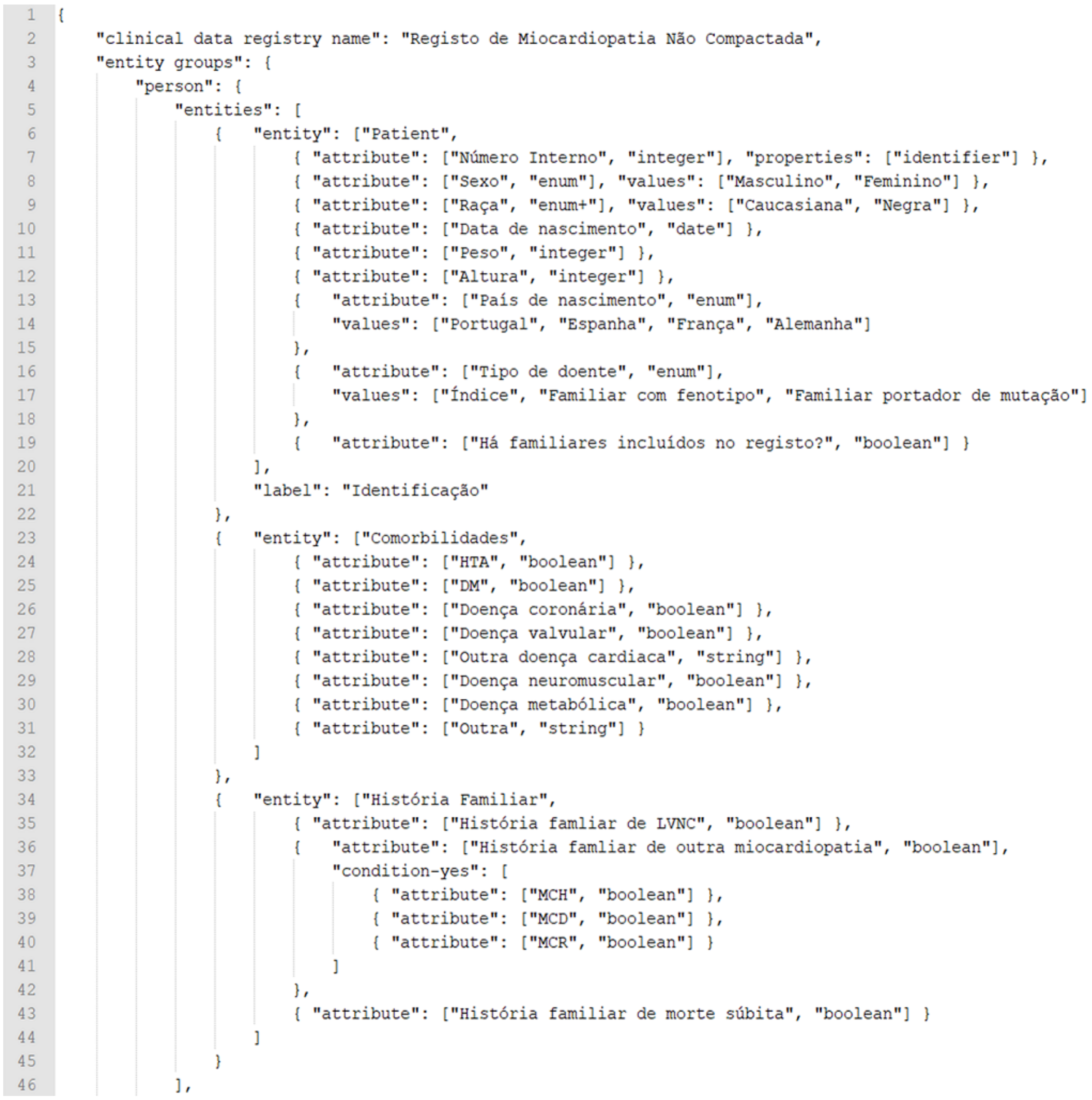
Blood pressure:

8	Systolic:	(mm)	Diastolic:		(Hg)

Figure A.1: Example of a CRF. Figure from https://docs.openclinica.com/sites/fileuploads/akaza/cms-docs/3.1/CRFWebView.png.

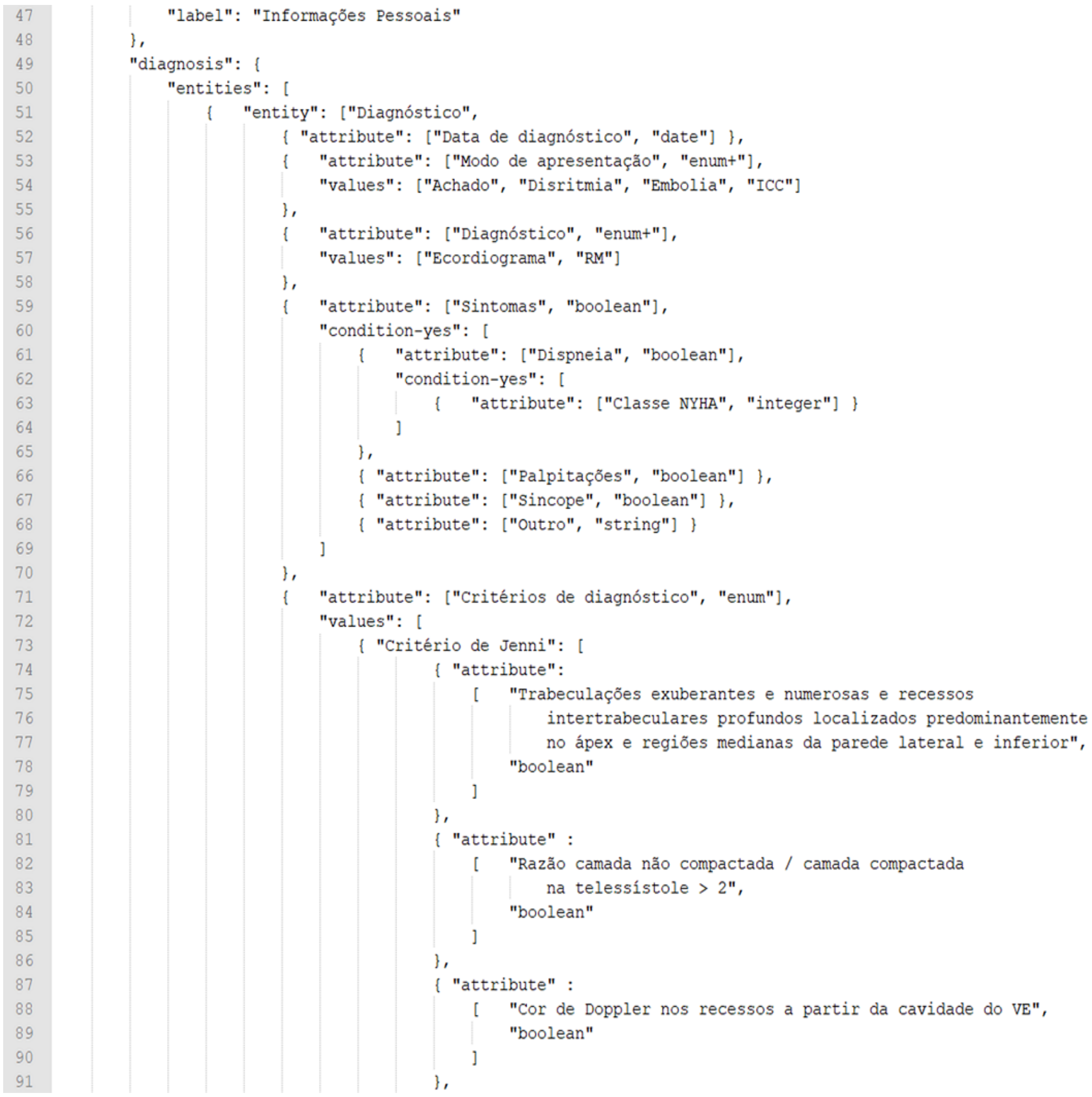
B

PRNC - JSON file specification



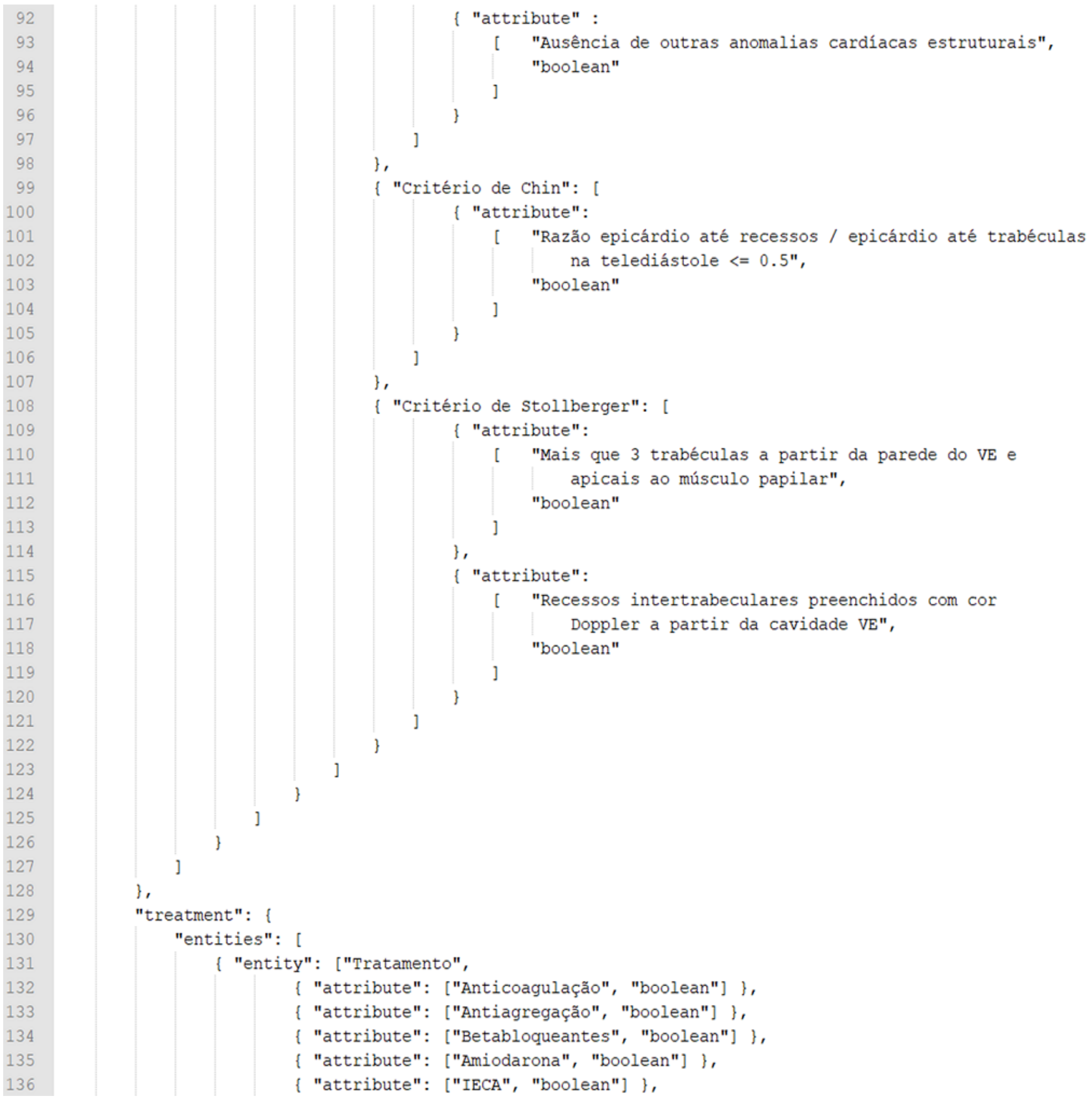
```
{
    "clinical data registry name": "Registo de Miocardiopatia Não Compactada",
    "entity groups": {
        "person": {
            "entities": [
                {   "entity": ["Patient",
                        { "attribute": ["Número Interno", "integer"], "properties": ["identifier"] },
                        { "attribute": ["Sexo", "enum"], "values": ["Masculino", "Feminino"] },
                        { "attribute": ["Raça", "enum+"], "values": ["Caucasiana", "Negra"] },
                        { "attribute": ["Data de nascimento", "date"] },
                        { "attribute": ["Peso", "integer"] },
                        { "attribute": ["Altura", "integer"] },
                        {   "attribute": ["País de nascimento", "enum"],
                            "values": ["Portugal", "Espanha", "França", "Alemanha"]
                        },
                        {   "attribute": ["Tipo de doente", "enum"],
                            "values": ["Índice", "Familiar com fenotipo", "Familiar portador de mutação"]
                        },
                        {   "attribute": ["Há familiares incluídos no registo?", "boolean"] }
                    ],
                    "label": "Identificação"
                },
                {   "entity": ["Comorbilidades",
                        { "attribute": ["HTA", "boolean"] },
                        { "attribute": ["DM", "boolean"] },
                        { "attribute": ["Doença coronária", "boolean"] },
                        { "attribute": ["Doença valvular", "boolean"] },
                        { "attribute": ["Outra doença cardiaca", "string"] },
                        { "attribute": ["Doença neuromuscular", "boolean"] },
                        { "attribute": ["Doença metabólica", "boolean"] },
                        { "attribute": ["Outra", "string"] }
                    ]
                },
                {   "entity": ["História Familiar",
                        { "attribute": ["História famliar de LVNC", "boolean"] },
                        {   "attribute": ["História famliar de outra miocardiopatia", "boolean"],
                            "condition-yes": [
                                { "attribute": ["MCH", "boolean"] },
                                { "attribute": ["MCD", "boolean"] },
                                { "attribute": ["MCR", "boolean"] }
                            ]
                        },
                        { "attribute": ["História familiar de morte súbita", "boolean"] }
                    ]
                }
            ],
```

Figure B.1: PRNC - JSON file specification 1/14.



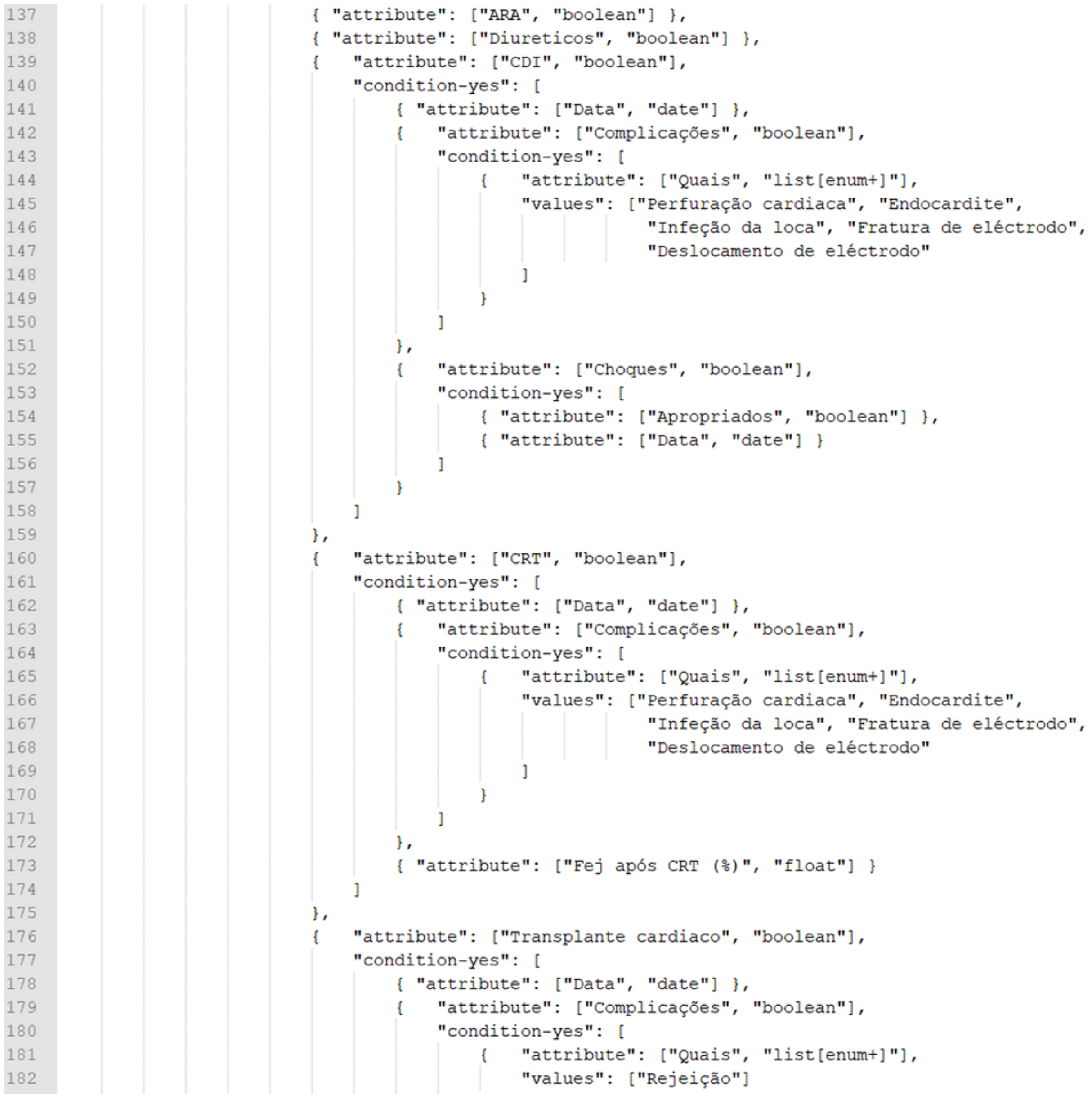
```
        "label": "Informações Pessoais"
    },
    "diagnosis": {
        "entities": [
            {   "entity": ["Diagnóstico",
                    { "attribute": ["Data de diagnóstico", "date"] },
                    {   "attribute": ["Modo de apresentação", "enum+"],
                        "values": ["Achado", "Disritmia", "Embolia", "ICC"]
                    },
                    {   "attribute": ["Diagnóstico", "enum+"],
                        "values": ["Ecordiograma", "RM"]
                    },
                    {   "attribute": ["Sintomas", "boolean"],
                        "condition-yes": [
                            {   "attribute": ["Dispneia", "boolean"],
                                "condition-yes": [
                                    {   "attribute": ["Classe NYHA", "integer"] }
                                ]
                            },
                            { "attribute": ["Palpitações", "boolean"] },
                            { "attribute": ["Sincope", "boolean"] },
                            { "attribute": ["Outro", "string"] }
                        ]
                    },
                    {   "attribute": ["Critérios de diagnóstico", "enum"],
                        "values": [
                            { "Critério de Jenni": [
                                    { "attribute":
                                        [   "Trabeculações exuberantes e numerosas e recessos
                                                intertrabeculares profundos localizados predominantemente
                                                no ápex e regiões medianas da parede lateral e inferior",
                                            "boolean"
                                        ]
                                    },
                                    { "attribute" :
                                        [   "Razão camada não compactada / camada compactada
                                                na telessístole > 2",
                                            "boolean"
                                        ]
                                    },
                                    { "attribute" :
                                        [   "Cor de Doppler nos recessos a partir da cavidade do VE",
                                            "boolean"
                                        ]
                                    },
```

Figure B.2: PRNC - JSON file specification 2/14.



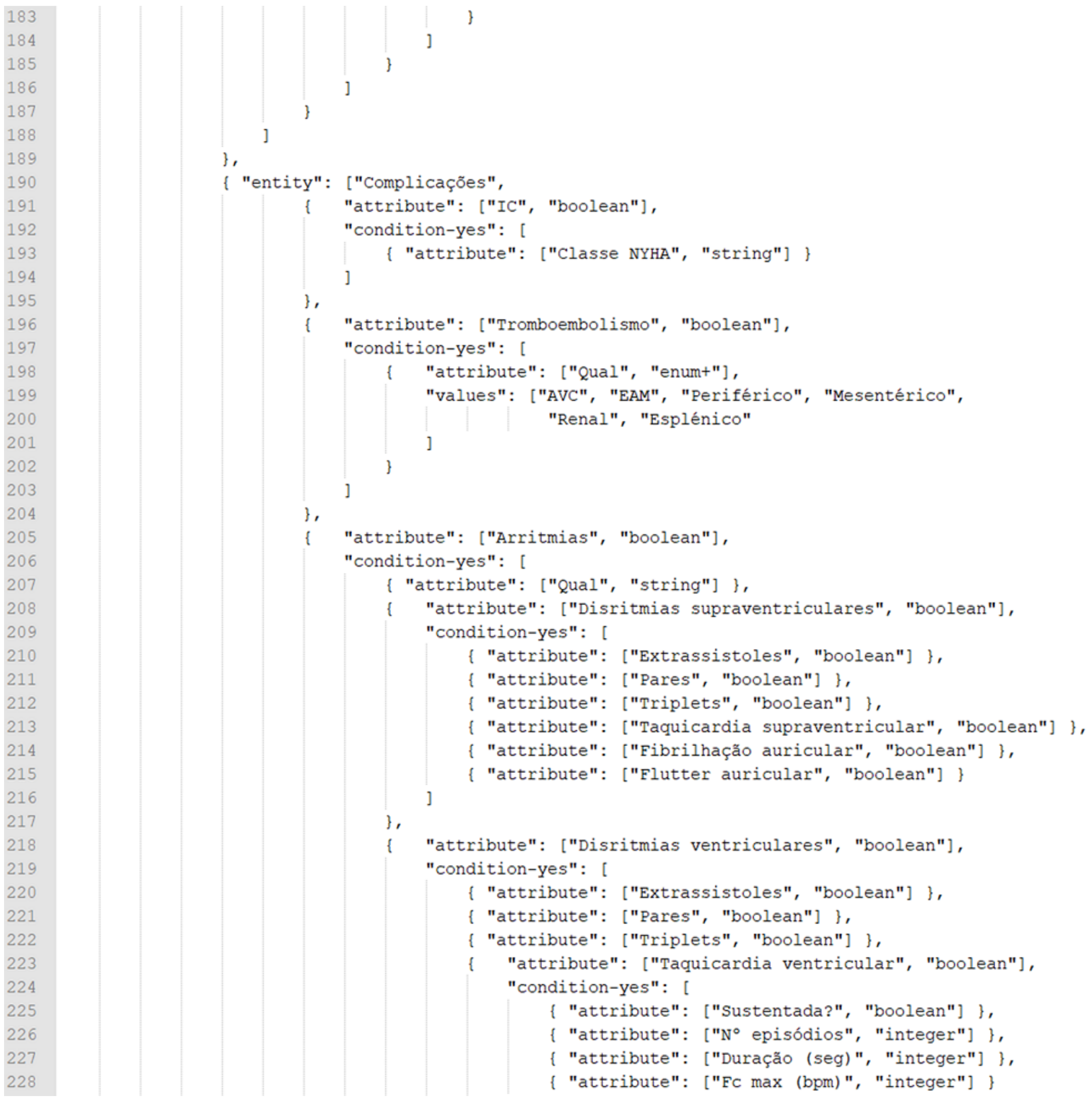
```
                                    { "attribute" :
                                        [   "Ausência de outras anomalias cardíacas estruturais",
                                            "boolean"
                                        ]
                                    }
                                ]
                            },
                            { "Critério de Chin": [
                                    { "attribute":
                                        [   "Razão epicárdio até recessos / epicárdio até trabéculas
                                                na telediástole <= 0.5",
                                            "boolean"
                                        ]
                                    }
                                ]
                            },
                            { "Critério de Stollberger": [
                                    { "attribute":
                                        [   "Mais que 3 trabéculas a partir da parede do VE e
                                                apicais ao músculo papilar",
                                            "boolean"
                                        ]
                                    },
                                    { "attribute":
                                        [   "Recessos intertrabeculares preenchidos com cor
                                                Doppler a partir da cavidade VE",
                                            "boolean"
                                        ]
                                    }
                                ]
                            }
                        ]
                    }
                ]
            }
        ]
    },
    "treatment": {
        "entities": [
            { "entity": ["Tratamento",
                    { "attribute": ["Anticoagulação", "boolean"] },
                    { "attribute": ["Antiagregação", "boolean"] },
                    { "attribute": ["Betabloqueantes", "boolean"] },
                    { "attribute": ["Amiodarona", "boolean"] },
                    { "attribute": ["IECA", "boolean"] },
```

Figure B.3: PRNC - JSON file specification 3/14.



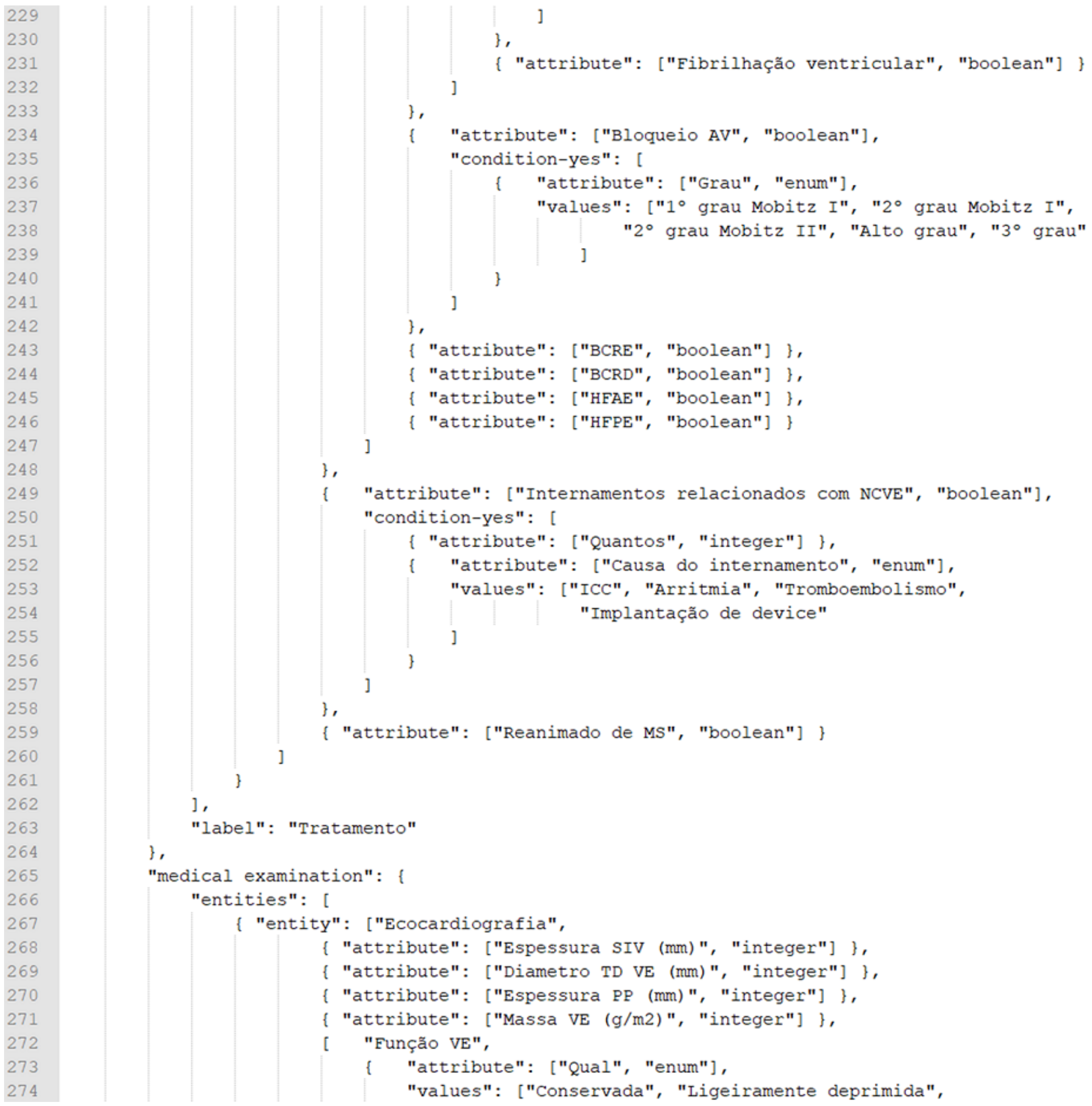
```json
{ "attribute": ["ARA", "boolean"] },
{ "attribute": ["Diureticos", "boolean"] },
{   "attribute": ["CDI", "boolean"],
    "condition-yes": [
        { "attribute": ["Data", "date"] },
        {   "attribute": ["Complicações", "boolean"],
            "condition-yes": [
                {   "attribute": ["Quais", "list[enum+]"],
                    "values": ["Perfuração cardiaca", "Endocardite",
                                "Infeção da loca", "Fratura de eléctrodo",
                                "Deslocamento de eléctrodo"
                    ]
                }
            ]
        },
        {   "attribute": ["Choques", "boolean"],
            "condition-yes": [
                { "attribute": ["Apropriados", "boolean"] },
                { "attribute": ["Data", "date"] }
            ]
        }
    ]
},
{   "attribute": ["CRT", "boolean"],
    "condition-yes": [
        { "attribute": ["Data", "date"] },
        {   "attribute": ["Complicações", "boolean"],
            "condition-yes": [
                {   "attribute": ["Quais", "list[enum+]"],
                    "values": ["Perfuração cardiaca", "Endocardite",
                                "Infeção da loca", "Fratura de eléctrodo",
                                "Deslocamento de eléctrodo"
                    ]
                }
            ]
        },
        { "attribute": ["Fej após CRT (%)", "float"] }
    ]
},
{   "attribute": ["Transplante cardiaco", "boolean"],
    "condition-yes": [
        { "attribute": ["Data", "date"] },
        {   "attribute": ["Complicações", "boolean"],
            "condition-yes": [
                {   "attribute": ["Quais", "list[enum+]"],
                    "values": ["Rejeição"]
```

Figure B.4: PRNC - JSON file specification 4/14.



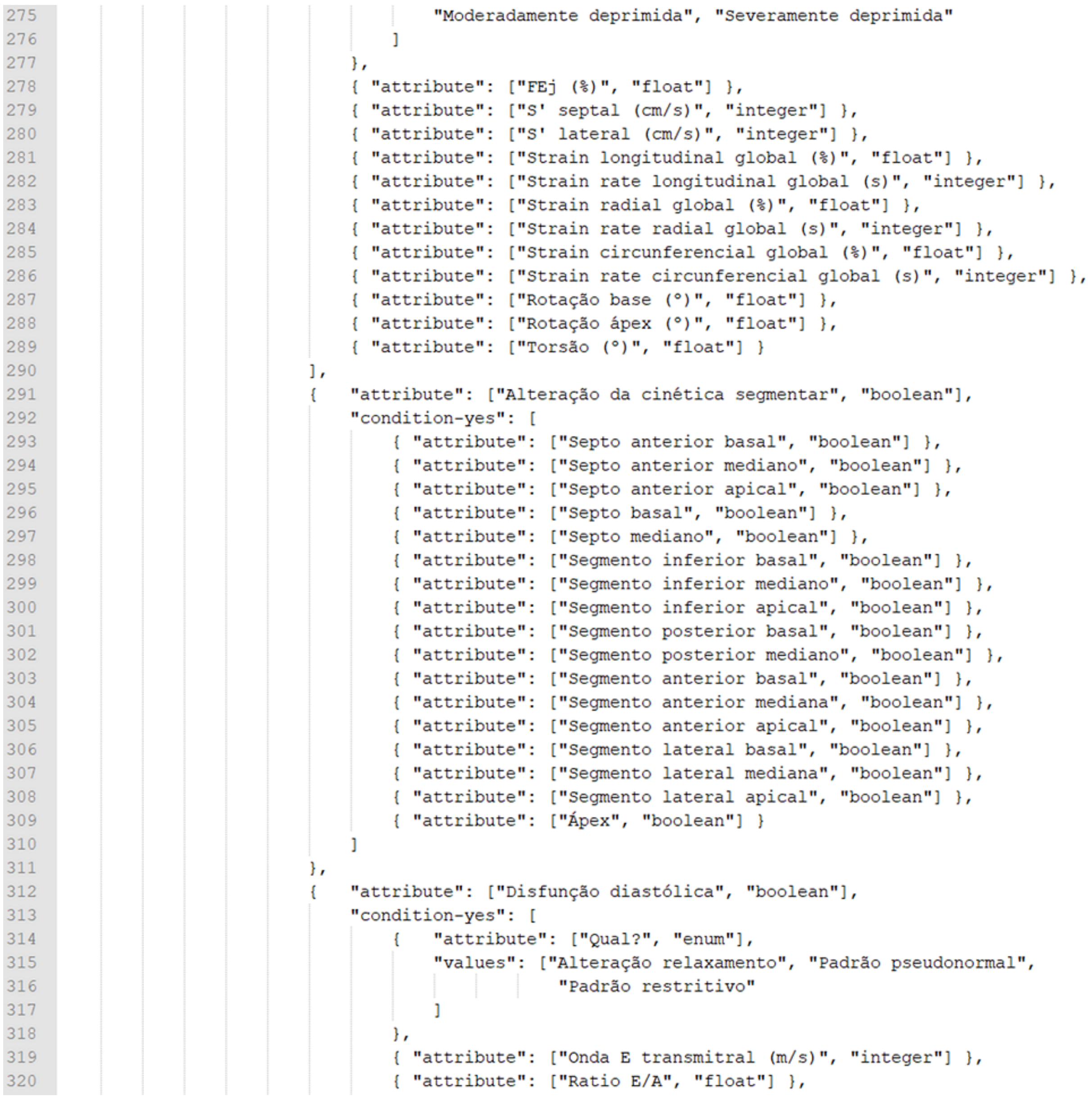
```
                                        }
                                    ]
                                }
                            ]
                        }
                    ]
                },
                { "entity": ["Complicações",
                        {   "attribute": ["IC", "boolean"],
                            "condition-yes": [
                                { "attribute": ["Classe NYHA", "string"] }
                            ]
                        },
                        {   "attribute": ["Tromboembolismo", "boolean"],
                            "condition-yes": [
                                {   "attribute": ["Qual", "enum+"],
                                    "values": ["AVC", "EAM", "Periférico", "Mesentérico",
                                                "Renal", "Esplénico"
                                    ]
                                }
                            ]
                        },
                        {   "attribute": ["Arritmias", "boolean"],
                            "condition-yes": [
                                { "attribute": ["Qual", "string"] },
                                {   "attribute": ["Disritmias supraventriculares", "boolean"],
                                    "condition-yes": [
                                        { "attribute": ["Extrassistoles", "boolean"] },
                                        { "attribute": ["Pares", "boolean"] },
                                        { "attribute": ["Triplets", "boolean"] },
                                        { "attribute": ["Taquicardia supraventricular", "boolean"] },
                                        { "attribute": ["Fibrilhação auricular", "boolean"] },
                                        { "attribute": ["Flutter auricular", "boolean"] }
                                    ]
                                },
                                {   "attribute": ["Disritmias ventriculares", "boolean"],
                                    "condition-yes": [
                                        { "attribute": ["Extrassistoles", "boolean"] },
                                        { "attribute": ["Pares", "boolean"] },
                                        { "attribute": ["Triplets", "boolean"] },
                                        {   "attribute": ["Taquicardia ventricular", "boolean"],
                                            "condition-yes": [
                                                { "attribute": ["Sustentada?", "boolean"] },
                                                { "attribute": ["Nº episódios", "integer"] },
                                                { "attribute": ["Duração (seg)", "integer"] },
                                                { "attribute": ["Fc max (bpm)", "integer"] }
```

Figure B.5: PRNC - JSON file specification 5/14.



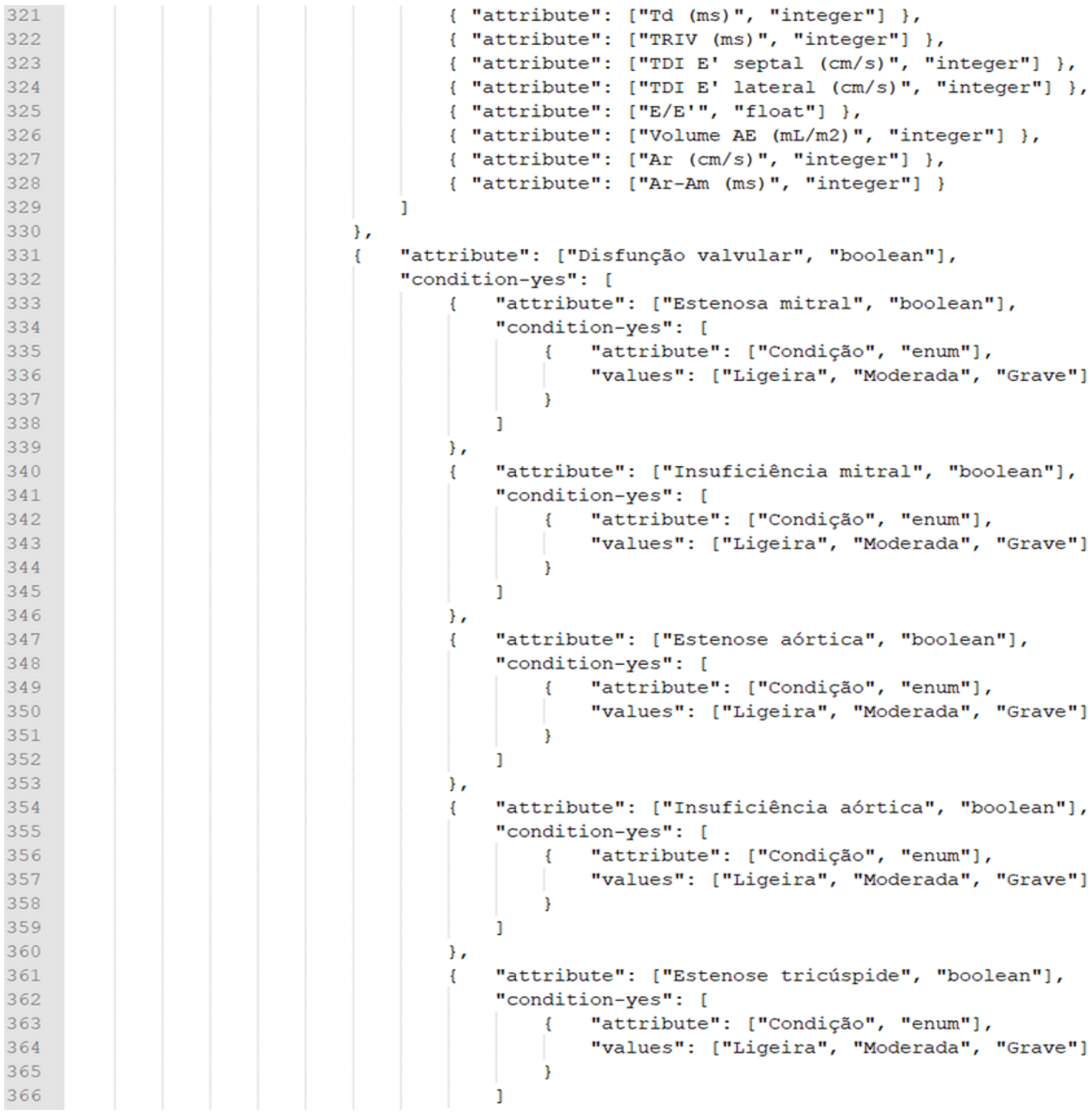
```
                                            ]
                                        },
                                        { "attribute": ["Fibrilhação ventricular", "boolean"] }
                                    ]
                                },
                                {   "attribute": ["Bloqueio AV", "boolean"],
                                    "condition-yes": [
                                        {   "attribute": ["Grau", "enum"],
                                            "values": ["1° grau Mobitz I", "2° grau Mobitz I",
                                                    "2° grau Mobitz II", "Alto grau", "3° grau"
                                                ]
                                        }
                                    ]
                                },
                                { "attribute": ["BCRE", "boolean"] },
                                { "attribute": ["BCRD", "boolean"] },
                                { "attribute": ["HFAE", "boolean"] },
                                { "attribute": ["HFPE", "boolean"] }
                            ]
                        },
                        {   "attribute": ["Internamentos relacionados com NCVE", "boolean"],
                            "condition-yes": [
                                { "attribute": ["Quantos", "integer"] },
                                {   "attribute": ["Causa do internamento", "enum"],
                                    "values": ["ICC", "Arritmia", "Tromboembolismo",
                                                "Implantação de device"
                                    ]
                                }
                            ]
                        },
                        { "attribute": ["Reanimado de MS", "boolean"] }
                    ]
                }
            ],
            "label": "Tratamento"
        },
        "medical examination": {
            "entities": [
                { "entity": ["Ecocardiografia",
                        { "attribute": ["Espessura SIV (mm)", "integer"] },
                        { "attribute": ["Diametro TD VE (mm)", "integer"] },
                        { "attribute": ["Espessura PP (mm)", "integer"] },
                        { "attribute": ["Massa VE (g/m2)", "integer"] },
                        [   "Função VE",
                            {   "attribute": ["Qual", "enum"],
                                "values": ["Conservada", "Ligeiramente deprimida",
```

Figure B.6: PRNC - JSON file specification 6/14.



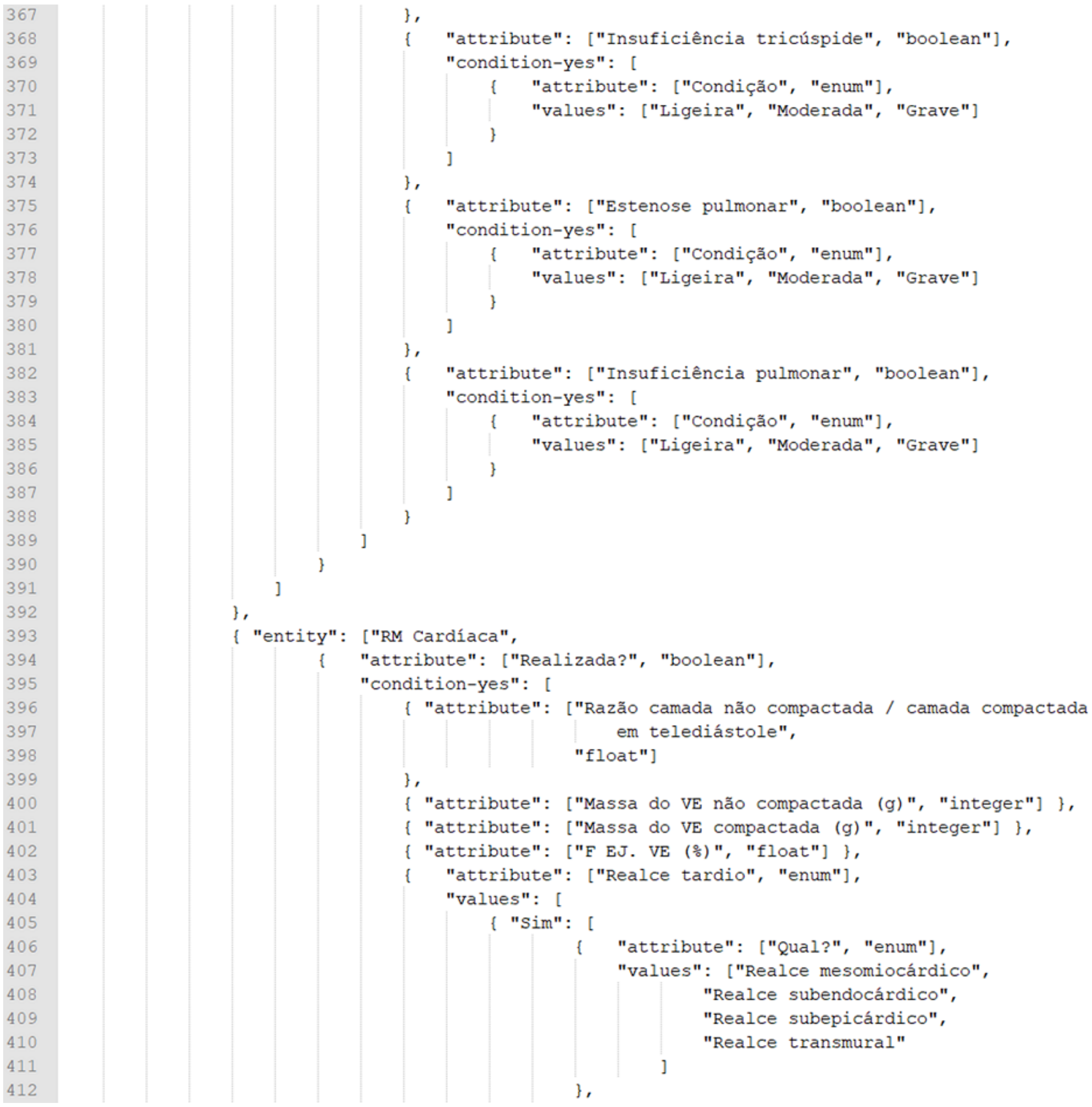
```
                                    "Moderadamente deprimida", "Severamente deprimida"
                                ]
                            },
                            { "attribute": ["FEj (%)", "float"] },
                            { "attribute": ["S' septal (cm/s)", "integer"] },
                            { "attribute": ["S' lateral (cm/s)", "integer"] },
                            { "attribute": ["Strain longitudinal global (%)", "float"] },
                            { "attribute": ["Strain rate longitudinal global (s)", "integer"] },
                            { "attribute": ["Strain radial global (%)", "float"] },
                            { "attribute": ["Strain rate radial global (s)", "integer"] },
                            { "attribute": ["Strain circunferencial global (%)", "float"] },
                            { "attribute": ["Strain rate circunferencial global (s)", "integer"] },
                            { "attribute": ["Rotação base (°)", "float"] },
                            { "attribute": ["Rotação ápex (°)", "float"] },
                            { "attribute": ["Torsão (°)", "float"] }
                        ],
                        {   "attribute": ["Alteração da cinética segmentar", "boolean"],
                            "condition-yes": [
                                { "attribute": ["Septo anterior basal", "boolean"] },
                                { "attribute": ["Septo anterior mediano", "boolean"] },
                                { "attribute": ["Septo anterior apical", "boolean"] },
                                { "attribute": ["Septo basal", "boolean"] },
                                { "attribute": ["Septo mediano", "boolean"] },
                                { "attribute": ["Segmento inferior basal", "boolean"] },
                                { "attribute": ["Segmento inferior mediano", "boolean"] },
                                { "attribute": ["Segmento inferior apical", "boolean"] },
                                { "attribute": ["Segmento posterior basal", "boolean"] },
                                { "attribute": ["Segmento posterior mediano", "boolean"] },
                                { "attribute": ["Segmento anterior basal", "boolean"] },
                                { "attribute": ["Segmento anterior mediana", "boolean"] },
                                { "attribute": ["Segmento anterior apical", "boolean"] },
                                { "attribute": ["Segmento lateral basal", "boolean"] },
                                { "attribute": ["Segmento lateral mediana", "boolean"] },
                                { "attribute": ["Segmento lateral apical", "boolean"] },
                                { "attribute": ["Ápex", "boolean"] }
                            ]
                        },
                        {   "attribute": ["Disfunção diastólica", "boolean"],
                            "condition-yes": [
                                {   "attribute": ["Qual?", "enum"],
                                    "values": ["Alteração relaxamento", "Padrão pseudonormal",
                                                "Padrão restritivo"
                                    ]
                                },
                                { "attribute": ["Onda E transmitral (m/s)", "integer"] },
                                { "attribute": ["Ratio E/A", "float"] },
```

Figure B.7: PRNC - JSON file specification 7/14.



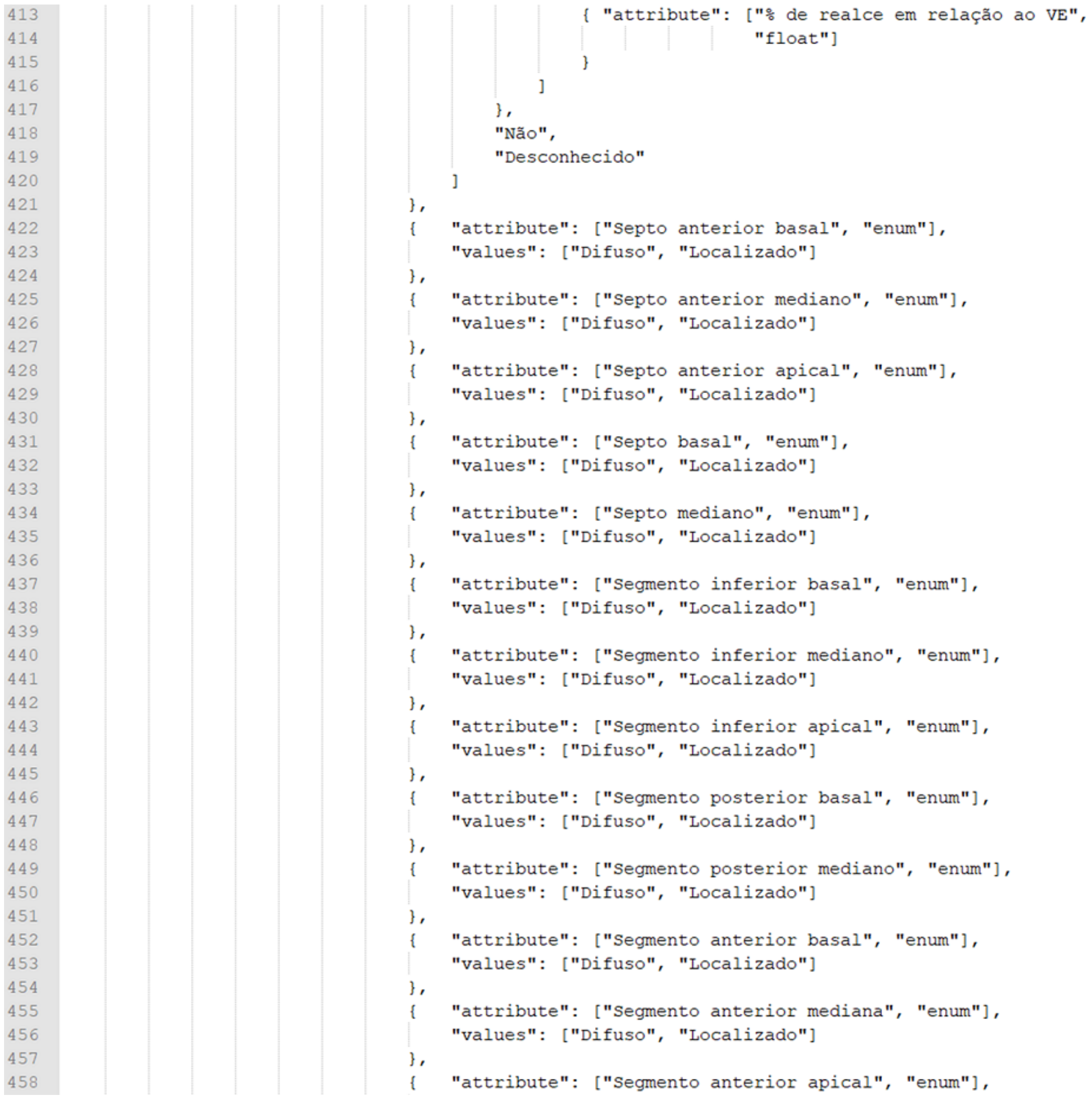
```
        { "attribute": ["Td (ms)", "integer"] },
        { "attribute": ["TRIV (ms)", "integer"] },
        { "attribute": ["TDI E' septal (cm/s)", "integer"] },
        { "attribute": ["TDI E' lateral (cm/s)", "integer"] },
        { "attribute": ["E/E'", "float"] },
        { "attribute": ["Volume AE (mL/m2)", "integer"] },
        { "attribute": ["Ar (cm/s)", "integer"] },
        { "attribute": ["Ar-Am (ms)", "integer"] }
    ]
},
{   "attribute": ["Disfunção valvular", "boolean"],
    "condition-yes": [
        {   "attribute": ["Estenosa mitral", "boolean"],
            "condition-yes": [
                {   "attribute": ["Condição", "enum"],
                    "values": ["Ligeira", "Moderada", "Grave"]
                }
            ]
        },
        {   "attribute": ["Insuficiência mitral", "boolean"],
            "condition-yes": [
                {   "attribute": ["Condição", "enum"],
                    "values": ["Ligeira", "Moderada", "Grave"]
                }
            ]
        },
        {   "attribute": ["Estenose aórtica", "boolean"],
            "condition-yes": [
                {   "attribute": ["Condição", "enum"],
                    "values": ["Ligeira", "Moderada", "Grave"]
                }
            ]
        },
        {   "attribute": ["Insuficiência aórtica", "boolean"],
            "condition-yes": [
                {   "attribute": ["Condição", "enum"],
                    "values": ["Ligeira", "Moderada", "Grave"]
                }
            ]
        },
        {   "attribute": ["Estenose tricúspide", "boolean"],
            "condition-yes": [
                {   "attribute": ["Condição", "enum"],
                    "values": ["Ligeira", "Moderada", "Grave"]
                }
            ]
```

Figure B.8: PRNC - JSON file specification 8/14.



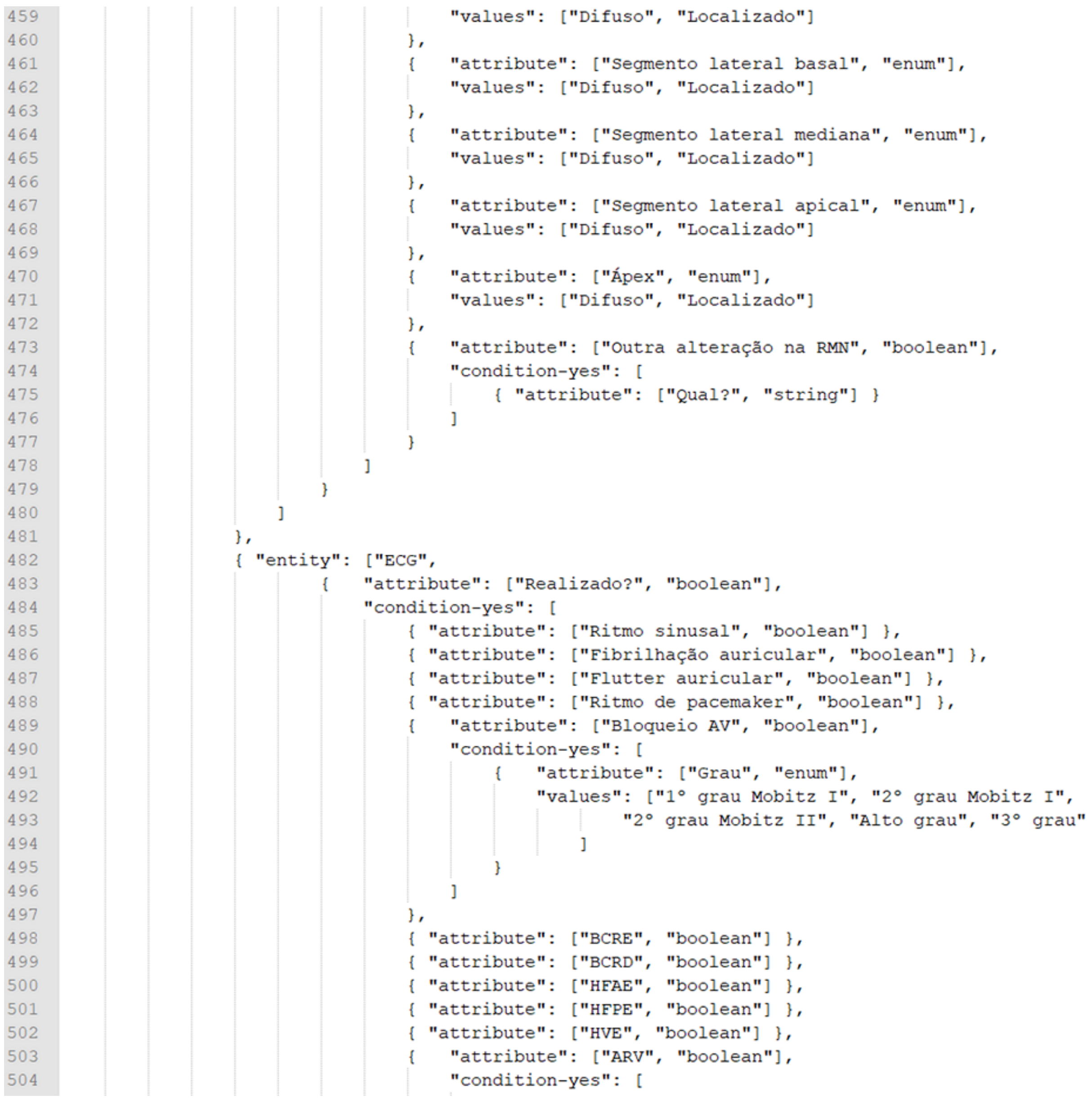
```
                                },
                                {   "attribute": ["Insuficiência tricúspide", "boolean"],
                                    "condition-yes": [
                                        {   "attribute": ["Condição", "enum"],
                                            "values": ["Ligeira", "Moderada", "Grave"]
                                        }
                                    ]
                                },
                                {   "attribute": ["Estenose pulmonar", "boolean"],
                                    "condition-yes": [
                                        {   "attribute": ["Condição", "enum"],
                                            "values": ["Ligeira", "Moderada", "Grave"]
                                        }
                                    ]
                                },
                                {   "attribute": ["Insuficiência pulmonar", "boolean"],
                                    "condition-yes": [
                                        {   "attribute": ["Condição", "enum"],
                                            "values": ["Ligeira", "Moderada", "Grave"]
                                        }
                                    ]
                                }
                            ]
                        }
                    ]
                },
                { "entity": ["RM Cardíaca",
                        {   "attribute": ["Realizada?", "boolean"],
                            "condition-yes": [
                                { "attribute": ["Razão camada não compactada / camada compactada
                                                    em telediástole",
                                                "float"]
                                },
                                { "attribute": ["Massa do VE não compactada (g)", "integer"] },
                                { "attribute": ["Massa do VE compactada (g)", "integer"] },
                                { "attribute": ["F EJ. VE (%)", "float"] },
                                {   "attribute": ["Realce tardio", "enum"],
                                    "values": [
                                        { "Sim": [
                                                {   "attribute": ["Qual?", "enum"],
                                                    "values": ["Realce mesomiocárdico",
                                                            "Realce subendocárdico",
                                                            "Realce subepicárdico",
                                                            "Realce transmural"
                                                        ]
                                                },
```

Figure B.9: PRNC - JSON file specification 9/14.



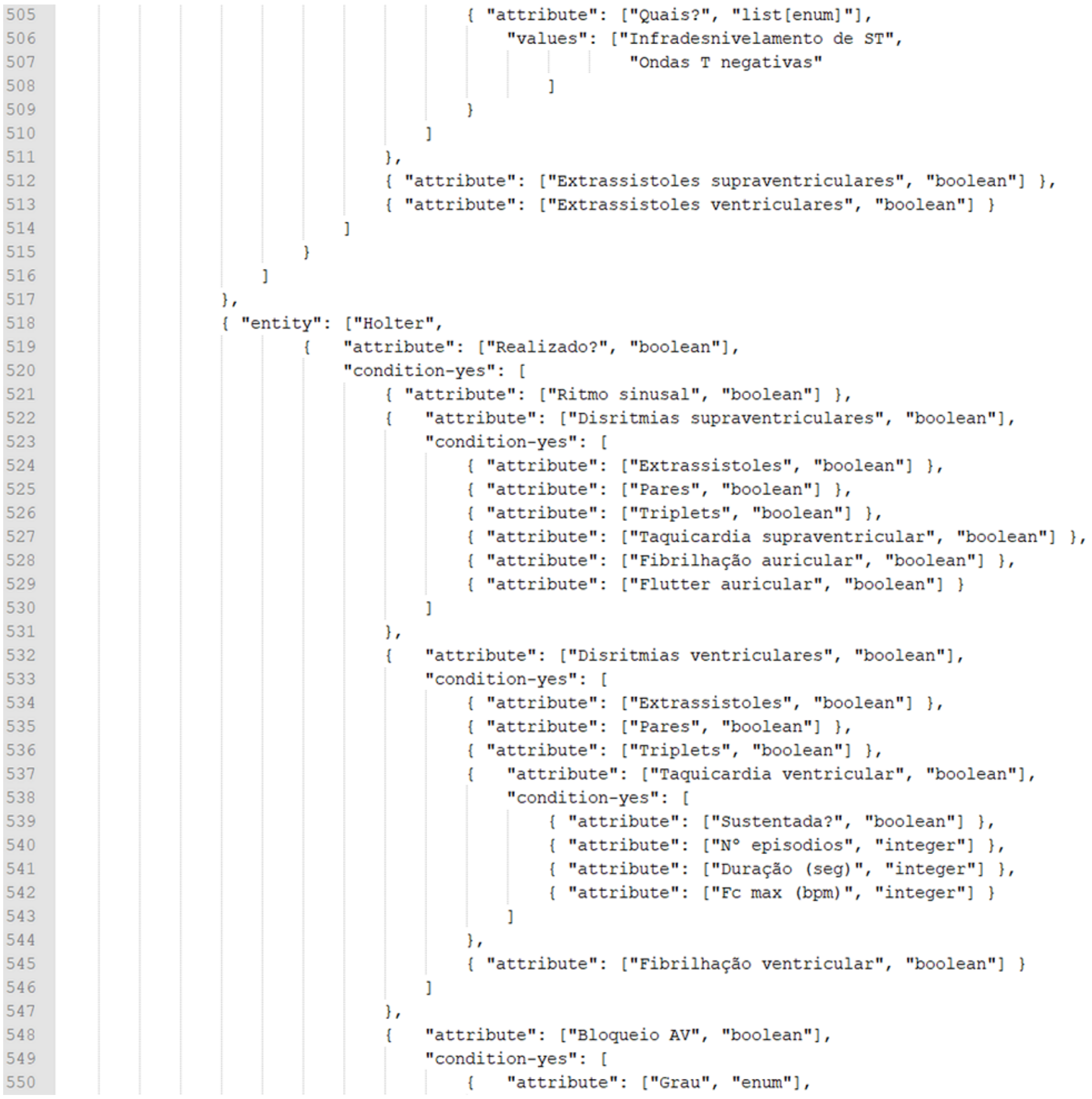
```
                                                { "attribute": ["% de realce em relação ao VE",
                                                                "float"]
                                                }
                                            ]
                                        },
                                        "Não",
                                        "Desconhecido"
                                    ]
                                },
                                {   "attribute": ["Septo anterior basal", "enum"],
                                    "values": ["Difuso", "Localizado"]
                                },
                                {   "attribute": ["Septo anterior mediano", "enum"],
                                    "values": ["Difuso", "Localizado"]
                                },
                                {   "attribute": ["Septo anterior apical", "enum"],
                                    "values": ["Difuso", "Localizado"]
                                },
                                {   "attribute": ["Septo basal", "enum"],
                                    "values": ["Difuso", "Localizado"]
                                },
                                {   "attribute": ["Septo mediano", "enum"],
                                    "values": ["Difuso", "Localizado"]
                                },
                                {   "attribute": ["Segmento inferior basal", "enum"],
                                    "values": ["Difuso", "Localizado"]
                                },
                                {   "attribute": ["Segmento inferior mediano", "enum"],
                                    "values": ["Difuso", "Localizado"]
                                },
                                {   "attribute": ["Segmento inferior apical", "enum"],
                                    "values": ["Difuso", "Localizado"]
                                },
                                {   "attribute": ["Segmento posterior basal", "enum"],
                                    "values": ["Difuso", "Localizado"]
                                },
                                {   "attribute": ["Segmento posterior mediano", "enum"],
                                    "values": ["Difuso", "Localizado"]
                                },
                                {   "attribute": ["Segmento anterior basal", "enum"],
                                    "values": ["Difuso", "Localizado"]
                                },
                                {   "attribute": ["Segmento anterior mediana", "enum"],
                                    "values": ["Difuso", "Localizado"]
                                },
                                {   "attribute": ["Segmento anterior apical", "enum"],
```

Figure B.10: PRNC - JSON file specification 10/14.



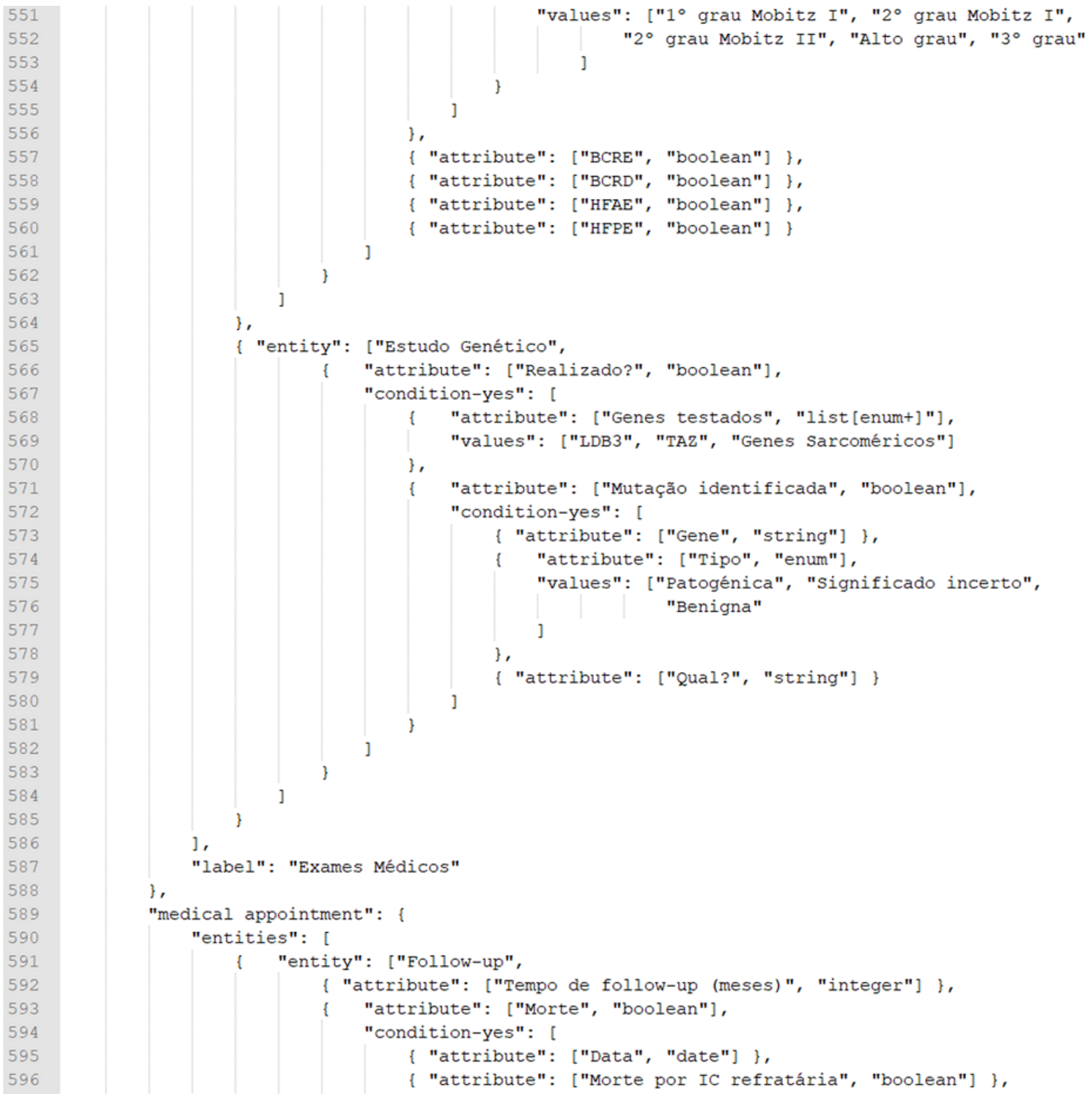
```
                                    "values": ["Difuso", "Localizado"]
                                },
                                {   "attribute": ["Segmento lateral basal", "enum"],
                                    "values": ["Difuso", "Localizado"]
                                },
                                {   "attribute": ["Segmento lateral mediana", "enum"],
                                    "values": ["Difuso", "Localizado"]
                                },
                                {   "attribute": ["Segmento lateral apical", "enum"],
                                    "values": ["Difuso", "Localizado"]
                                },
                                {   "attribute": ["Ápex", "enum"],
                                    "values": ["Difuso", "Localizado"]
                                },
                                {   "attribute": ["Outra alteração na RMN", "boolean"],
                                    "condition-yes": [
                                        { "attribute": ["Qual?", "string"] }
                                    ]
                                }
                            ]
                        }
                    ]
                },
                { "entity": ["ECG",
                        {   "attribute": ["Realizado?", "boolean"],
                            "condition-yes": [
                                { "attribute": ["Ritmo sinusal", "boolean"] },
                                { "attribute": ["Fibrilhação auricular", "boolean"] },
                                { "attribute": ["Flutter auricular", "boolean"] },
                                { "attribute": ["Ritmo de pacemaker", "boolean"] },
                                {   "attribute": ["Bloqueio AV", "boolean"],
                                    "condition-yes": [
                                        {   "attribute": ["Grau", "enum"],
                                            "values": ["1° grau Mobitz I", "2° grau Mobitz I",
                                                    "2° grau Mobitz II", "Alto grau", "3° grau"
                                                ]
                                        }
                                    ]
                                },
                                { "attribute": ["BCRE", "boolean"] },
                                { "attribute": ["BCRD", "boolean"] },
                                { "attribute": ["HFAE", "boolean"] },
                                { "attribute": ["HFPE", "boolean"] },
                                { "attribute": ["HVE", "boolean"] },
                                {   "attribute": ["ARV", "boolean"],
                                    "condition-yes": [
```

Figure B.11: PRNC - JSON file specification 11/14.



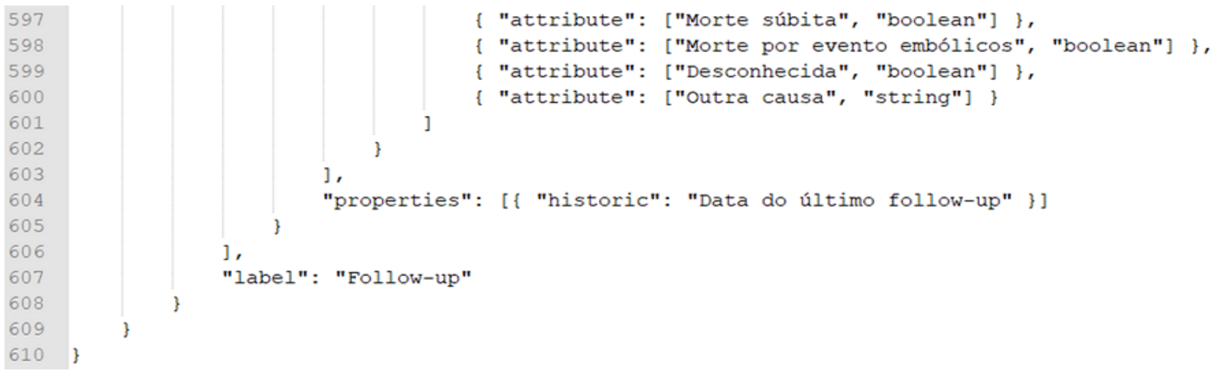
```
                                        { "attribute": ["Quais?", "list[enum]"],
                                            "values": ["Infradesnivelamento de ST",
                                                        "Ondas T negativas"
                                                ]
                                        }
                                    ]
                                },
                                { "attribute": ["Extrassistoles supraventriculares", "boolean"] },
                                { "attribute": ["Extrassistoles ventriculares", "boolean"] }
                            ]
                        }
                    ]
                },
                { "entity": ["Holter",
                        {   "attribute": ["Realizado?", "boolean"],
                            "condition-yes": [
                                { "attribute": ["Ritmo sinusal", "boolean"] },
                                {   "attribute": ["Disritmias supraventriculares", "boolean"],
                                    "condition-yes": [
                                        { "attribute": ["Extrassistoles", "boolean"] },
                                        { "attribute": ["Pares", "boolean"] },
                                        { "attribute": ["Triplets", "boolean"] },
                                        { "attribute": ["Taquicardia supraventricular", "boolean"] },
                                        { "attribute": ["Fibrilhação auricular", "boolean"] },
                                        { "attribute": ["Flutter auricular", "boolean"] }
                                    ]
                                },
                                {   "attribute": ["Disritmias ventriculares", "boolean"],
                                    "condition-yes": [
                                        { "attribute": ["Extrassistoles", "boolean"] },
                                        { "attribute": ["Pares", "boolean"] },
                                        { "attribute": ["Triplets", "boolean"] },
                                        {   "attribute": ["Taquicardia ventricular", "boolean"],
                                            "condition-yes": [
                                                { "attribute": ["Sustentada?", "boolean"] },
                                                { "attribute": ["Nº episodios", "integer"] },
                                                { "attribute": ["Duração (seg)", "integer"] },
                                                { "attribute": ["Fc max (bpm)", "integer"] }
                                            ]
                                        },
                                        { "attribute": ["Fibrilhação ventricular", "boolean"] }
                                    ]
                                },
                                {   "attribute": ["Bloqueio AV", "boolean"],
                                    "condition-yes": [
                                        {   "attribute": ["Grau", "enum"],
```

Figure B.12: PRNC - JSON file specification 12/14.



```
                                            "values": ["1° grau Mobitz I", "2° grau Mobitz I",
                                                    "2° grau Mobitz II", "Alto grau", "3° grau"
                                                ]
                                        }
                                    ]
                                },
                                { "attribute": ["BCRE", "boolean"] },
                                { "attribute": ["BCRD", "boolean"] },
                                { "attribute": ["HFAE", "boolean"] },
                                { "attribute": ["HFPE", "boolean"] }
                            ]
                        }
                    ]
                },
                { "entity": ["Estudo Genético",
                        {   "attribute": ["Realizado?", "boolean"],
                            "condition-yes": [
                                {   "attribute": ["Genes testados", "list[enum+]"],
                                    "values": ["LDB3", "TAZ", "Genes Sarcoméricos"]
                                },
                                {   "attribute": ["Mutação identificada", "boolean"],
                                    "condition-yes": [
                                        { "attribute": ["Gene", "string"] },
                                        {   "attribute": ["Tipo", "enum"],
                                            "values": ["Patogénica", "Significado incerto",
                                                        "Benigna"
                                            ]
                                        },
                                        { "attribute": ["Qual?", "string"] }
                                    ]
                                }
                            ]
                        }
                    ]
                }
            ],
            "label": "Exames Médicos"
        },
        "medical appointment": {
            "entities": [
                {   "entity": ["Follow-up",
                        { "attribute": ["Tempo de follow-up (meses)", "integer"] },
                        {   "attribute": ["Morte", "boolean"],
                            "condition-yes": [
                                { "attribute": ["Data", "date"] },
                                { "attribute": ["Morte por IC refratária", "boolean"] },
```

Figure B.13: PRNC - JSON file specification 13/14.

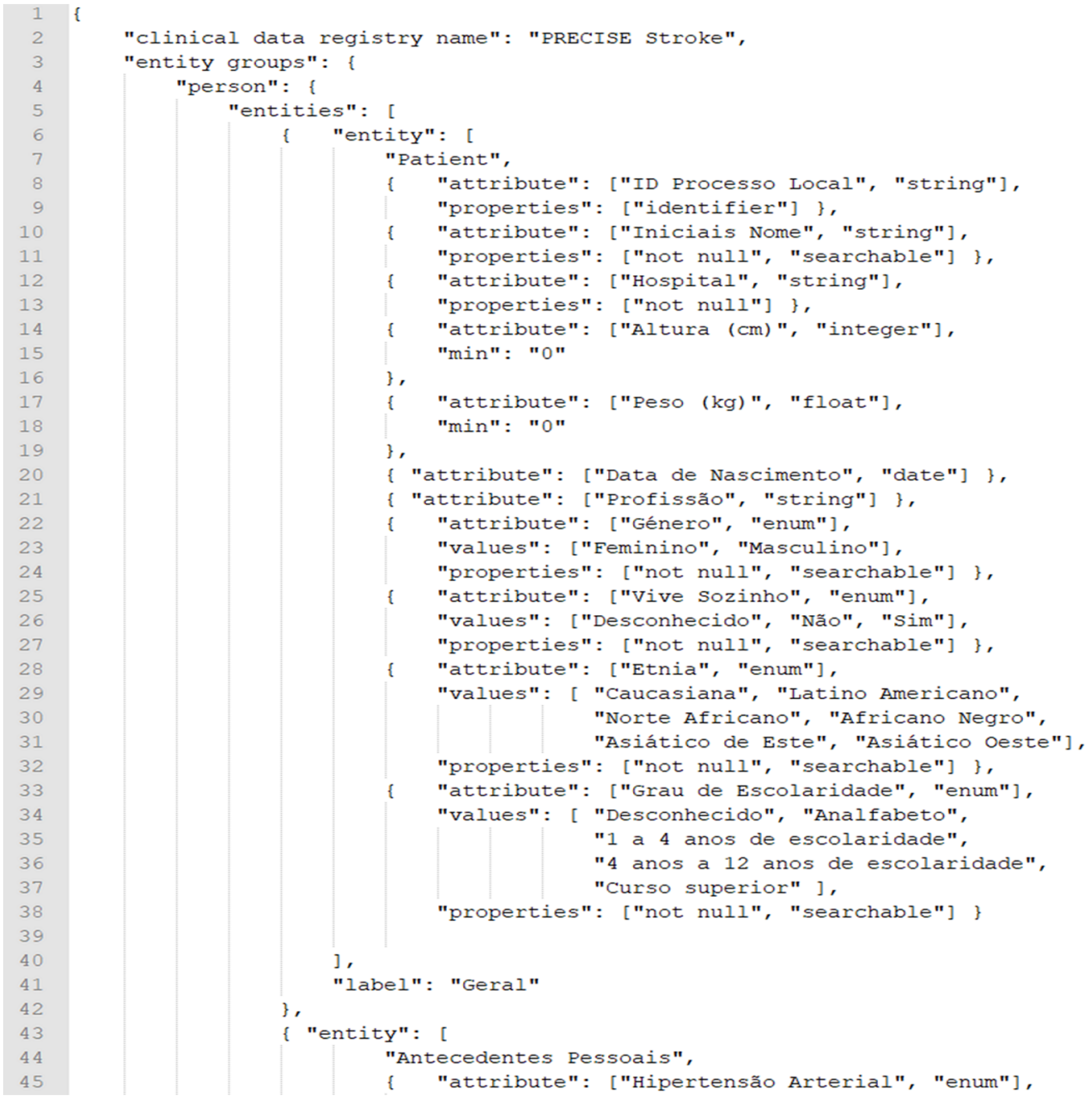


```
                                { "attribute": ["Morte súbita", "boolean"] },
                                { "attribute": ["Morte por evento embólicos", "boolean"] },
                                { "attribute": ["Desconhecida", "boolean"] },
                                { "attribute": ["Outra causa", "string"] }
                            ]
                        }
                    ],
                    "properties": [{ "historic": "Data do último follow-up" }]
                }
            ],
            "label": "Follow-up"
        }
    }
}
```

Figure B.14: PRNC - JSON file specification 14/14.

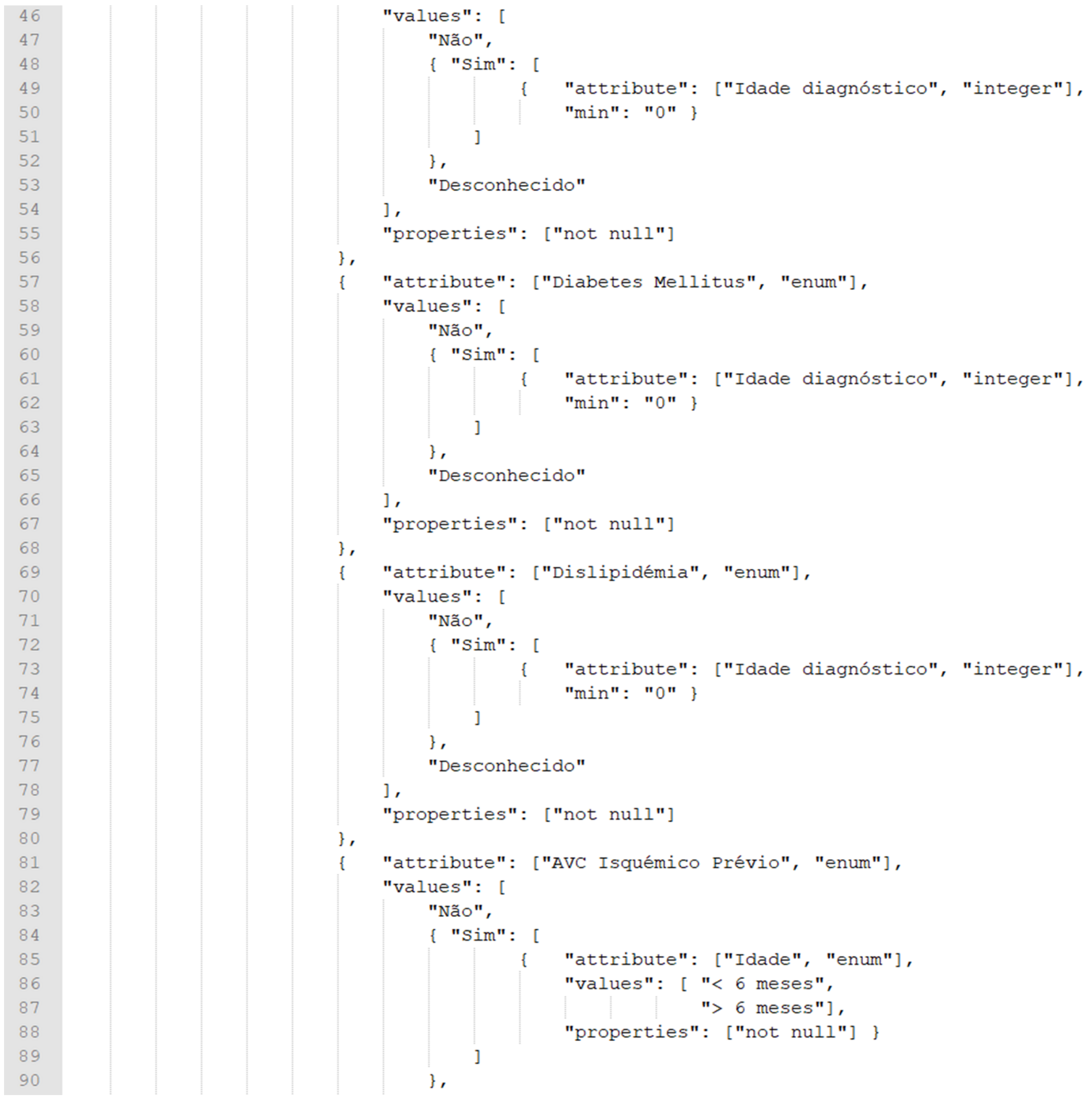
C

PRECISE Stroke - JSON file specification



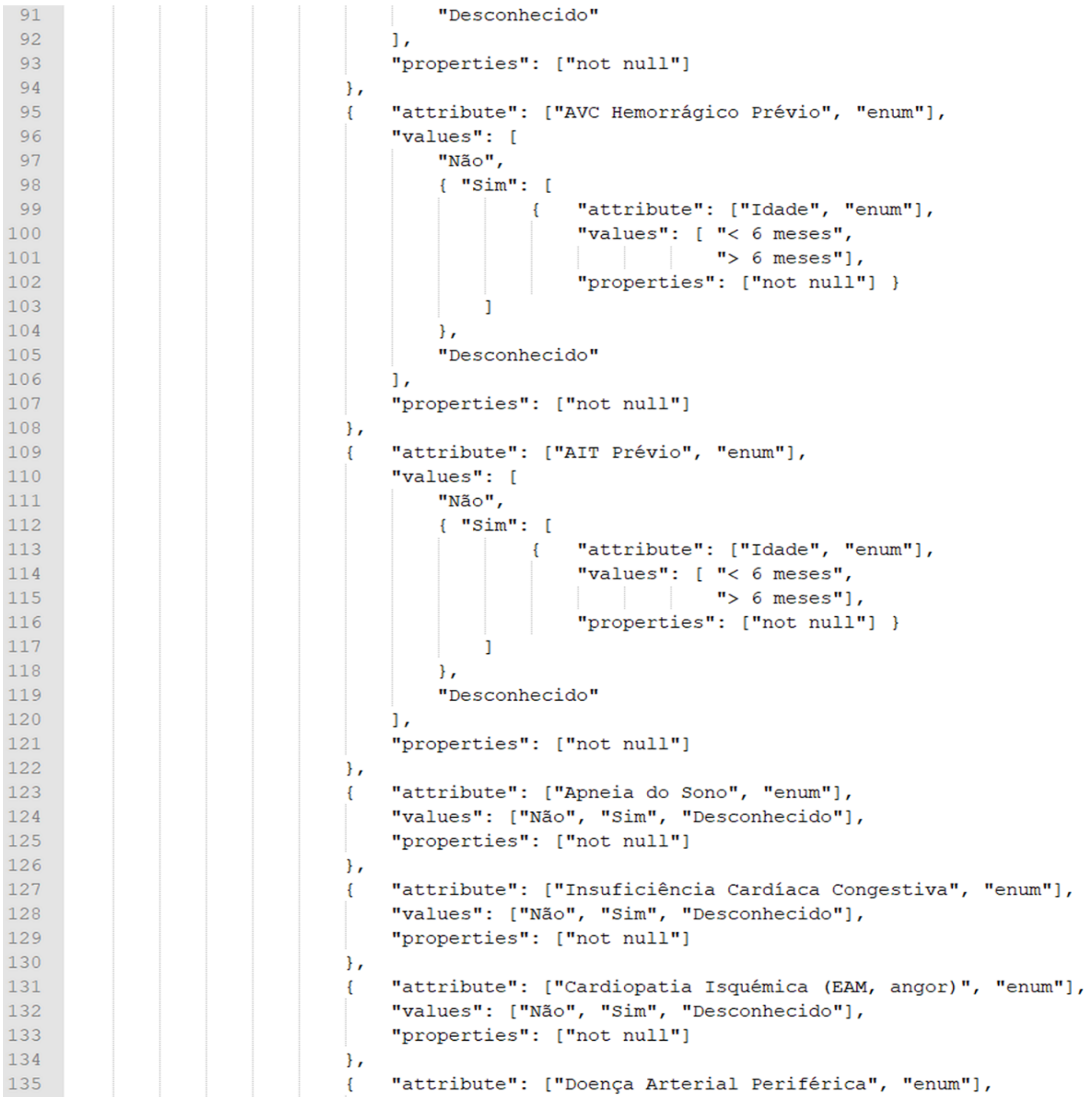
```
{
    "clinical data registry name": "PRECISE Stroke",
    "entity groups": {
        "person": {
            "entities": [
                {   "entity": [
                        "Patient",
                        {   "attribute": ["ID Processo Local", "string"],
                            "properties": ["identifier"] },
                        {   "attribute": ["Iniciais Nome", "string"],
                            "properties": ["not null", "searchable"] },
                        {   "attribute": ["Hospital", "string"],
                            "properties": ["not null"] },
                        {   "attribute": ["Altura (cm)", "integer"],
                            "min": "0"
                        },
                        {   "attribute": ["Peso (kg)", "float"],
                            "min": "0"
                        },
                        { "attribute": ["Data de Nascimento", "date"] },
                        { "attribute": ["Profissão", "string"] },
                        {   "attribute": ["Género", "enum"],
                            "values": ["Feminino", "Masculino"],
                            "properties": ["not null", "searchable"] },
                        {   "attribute": ["Vive Sozinho", "enum"],
                            "values": ["Desconhecido", "Não", "Sim"],
                            "properties": ["not null", "searchable"] },
                        {   "attribute": ["Etnia", "enum"],
                            "values": [ "Caucasiana", "Latino Americano",
                                        "Norte Africano", "Africano Negro",
                                        "Asiático de Este", "Asiático Oeste"],
                            "properties": ["not null", "searchable"] },
                        {   "attribute": ["Grau de Escolaridade", "enum"],
                            "values": [ "Desconhecido", "Analfabeto",
                                        "1 a 4 anos de escolaridade",
                                        "4 anos a 12 anos de escolaridade",
                                        "Curso superior" ],
                            "properties": ["not null", "searchable"] }

                    ],
                    "label": "Geral"
                },
                { "entity": [
                        "Antecedentes Pessoais",
                        {   "attribute": ["Hipertensão Arterial", "enum"],
```

Figure C.1: PRECISE Stroke - JSON file specification 1/57.



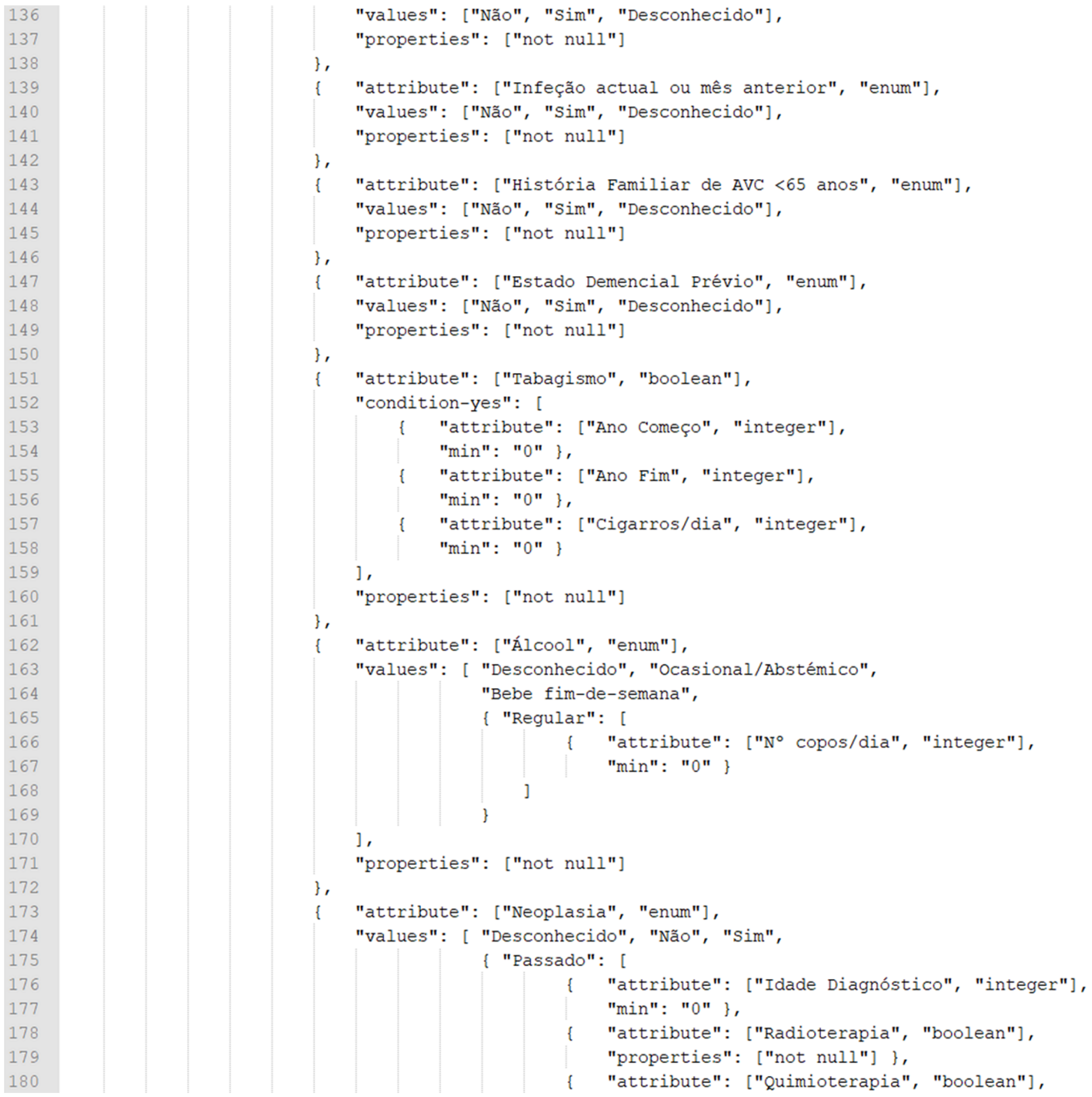
```
                            "values": [
                                "Não",
                                { "Sim": [
                                        {   "attribute": ["Idade diagnóstico", "integer"],
                                            "min": "0" }
                                    ]
                                },
                                "Desconhecido"
                            ],
                            "properties": ["not null"]
                        },
                        {   "attribute": ["Diabetes Mellitus", "enum"],
                            "values": [
                                "Não",
                                { "Sim": [
                                        {   "attribute": ["Idade diagnóstico", "integer"],
                                            "min": "0" }
                                    ]
                                },
                                "Desconhecido"
                            ],
                            "properties": ["not null"]
                        },
                        {   "attribute": ["Dislipidémia", "enum"],
                            "values": [
                                "Não",
                                { "Sim": [
                                        {   "attribute": ["Idade diagnóstico", "integer"],
                                            "min": "0" }
                                    ]
                                },
                                "Desconhecido"
                            ],
                            "properties": ["not null"]
                        },
                        {   "attribute": ["AVC Isquémico Prévio", "enum"],
                            "values": [
                                "Não",
                                { "Sim": [
                                        {   "attribute": ["Idade", "enum"],
                                            "values": [ "< 6 meses",
                                                        "> 6 meses"],
                                            "properties": ["not null"] }
                                    ]
                                },
```

Figure C.2: PRECISE Stroke - JSON file specification 2/57.



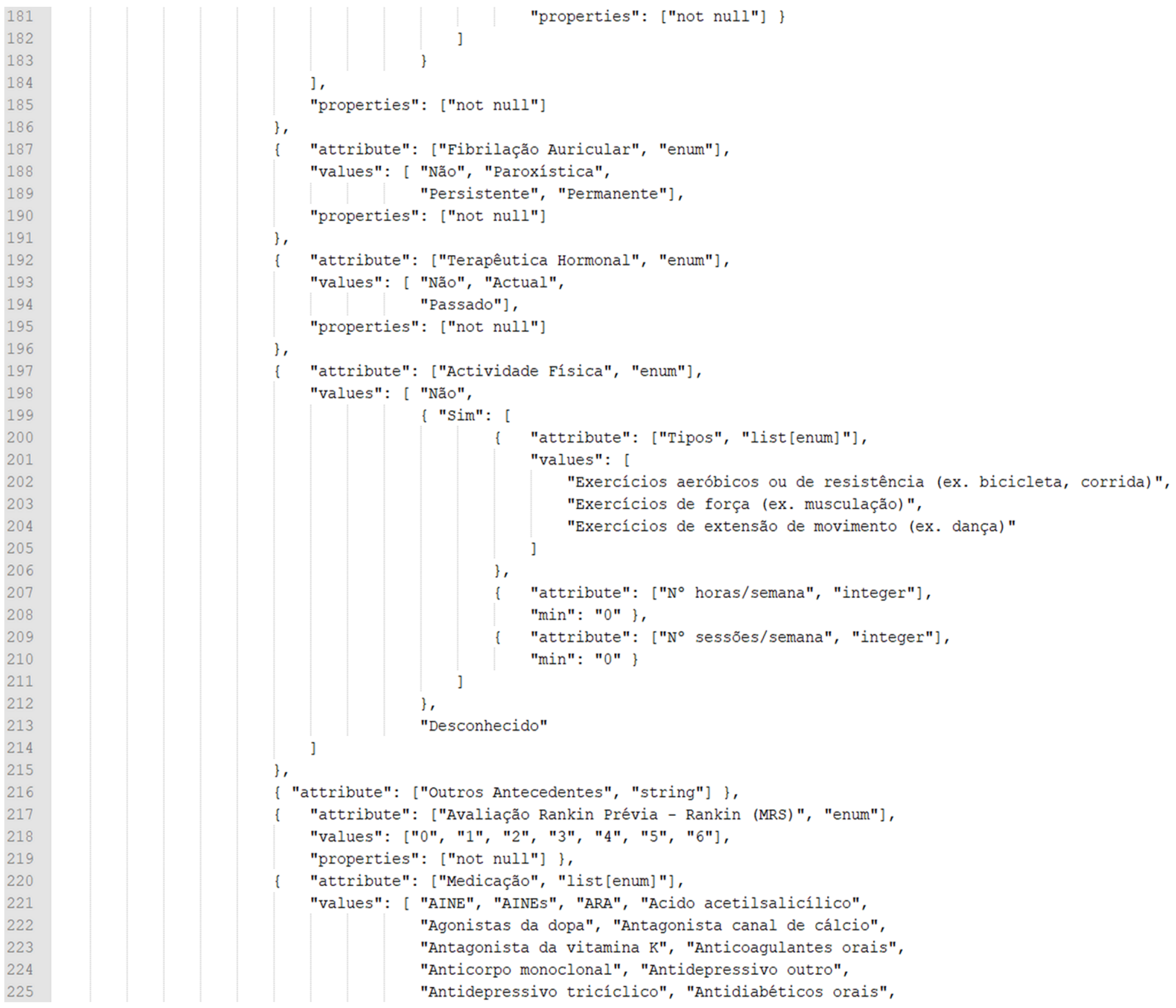
```json
        "Desconhecido"
    ],
    "properties": ["not null"]
},
{   "attribute": ["AVC Hemorrágico Prévio", "enum"],
    "values": [
        "Não",
        { "Sim": [
                {   "attribute": ["Idade", "enum"],
                    "values": [ "< 6 meses",
                                "> 6 meses"],
                    "properties": ["not null"] }
            ]
        },
        "Desconhecido"
    ],
    "properties": ["not null"]
},
{   "attribute": ["AIT Prévio", "enum"],
    "values": [
        "Não",
        { "Sim": [
                {   "attribute": ["Idade", "enum"],
                    "values": [ "< 6 meses",
                                "> 6 meses"],
                    "properties": ["not null"] }
            ]
        },
        "Desconhecido"
    ],
    "properties": ["not null"]
},
{   "attribute": ["Apneia do Sono", "enum"],
    "values": ["Não", "Sim", "Desconhecido"],
    "properties": ["not null"]
},
{   "attribute": ["Insuficiência Cardíaca Congestiva", "enum"],
    "values": ["Não", "Sim", "Desconhecido"],
    "properties": ["not null"]
},
{   "attribute": ["Cardiopatia Isquémica (EAM, angor)", "enum"],
    "values": ["Não", "Sim", "Desconhecido"],
    "properties": ["not null"]
},
{   "attribute": ["Doença Arterial Periférica", "enum"],
```

Figure C.3: PRECISE Stroke - JSON file specification 3/57.



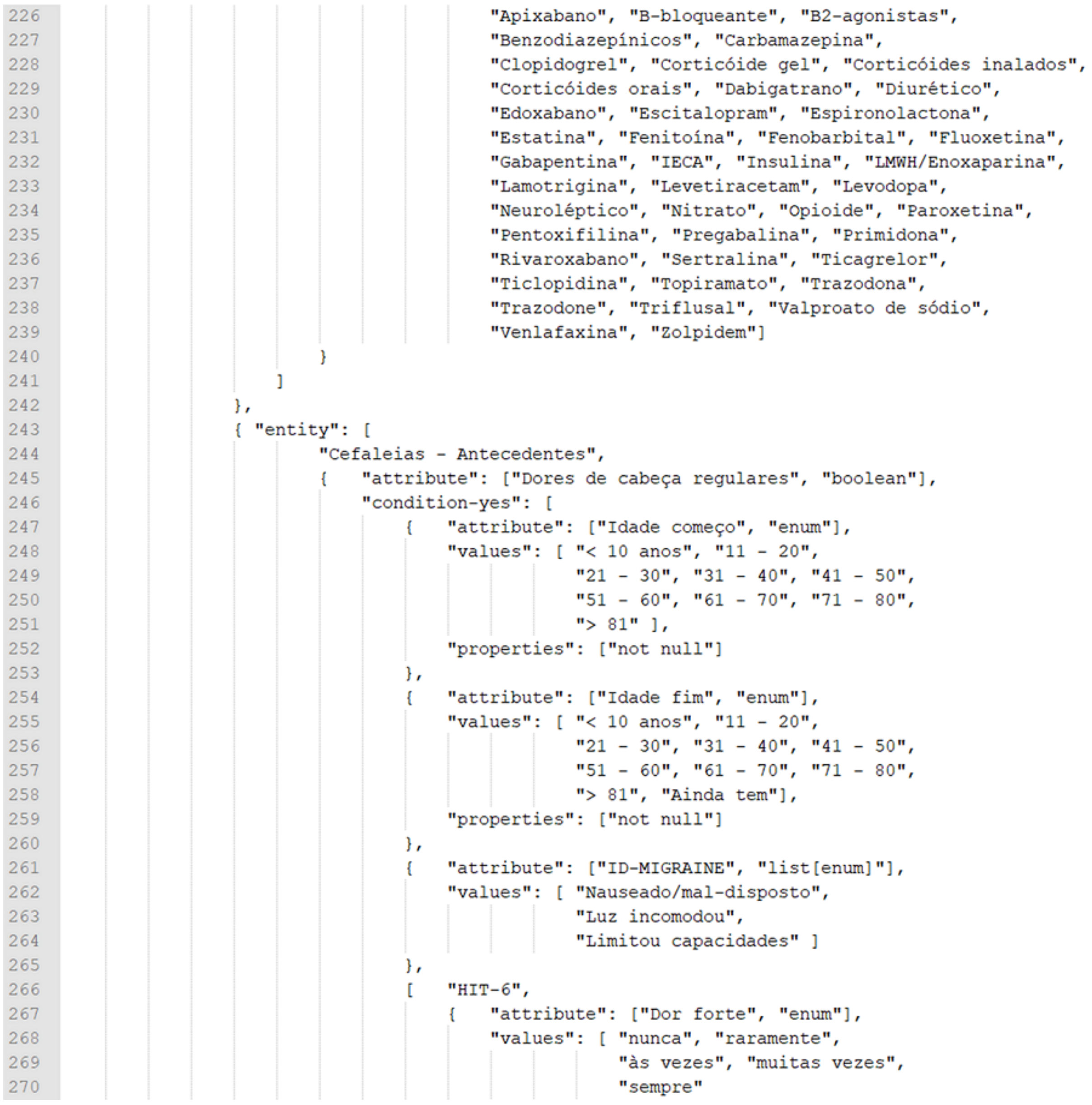
```
                            "values": ["Não", "Sim", "Desconhecido"],
                            "properties": ["not null"]
                        },
                        {   "attribute": ["Infeção actual ou mês anterior", "enum"],
                            "values": ["Não", "Sim", "Desconhecido"],
                            "properties": ["not null"]
                        },
                        {   "attribute": ["História Familiar de AVC <65 anos", "enum"],
                            "values": ["Não", "Sim", "Desconhecido"],
                            "properties": ["not null"]
                        },
                        {   "attribute": ["Estado Demencial Prévio", "enum"],
                            "values": ["Não", "Sim", "Desconhecido"],
                            "properties": ["not null"]
                        },
                        {   "attribute": ["Tabagismo", "boolean"],
                            "condition-yes": [
                                {   "attribute": ["Ano Começo", "integer"],
                                    "min": "0" },
                                {   "attribute": ["Ano Fim", "integer"],
                                    "min": "0" },
                                {   "attribute": ["Cigarros/dia", "integer"],
                                    "min": "0" }
                            ],
                            "properties": ["not null"]
                        },
                        {   "attribute": ["Álcool", "enum"],
                            "values": [ "Desconhecido", "Ocasional/Abstémico",
                                        "Bebe fim-de-semana",
                                        { "Regular": [
                                                {   "attribute": ["Nº copos/dia", "integer"],
                                                    "min": "0" }
                                            ]
                                        }
                            ],
                            "properties": ["not null"]
                        },
                        {   "attribute": ["Neoplasia", "enum"],
                            "values": [ "Desconhecido", "Não", "Sim",
                                        { "Passado": [
                                                {   "attribute": ["Idade Diagnóstico", "integer"],
                                                    "min": "0" },
                                                {   "attribute": ["Radioterapia", "boolean"],
                                                    "properties": ["not null"] },
                                                {   "attribute": ["Quimioterapia", "boolean"],
```

Figure C.4: PRECISE Stroke - JSON file specification 4/57.



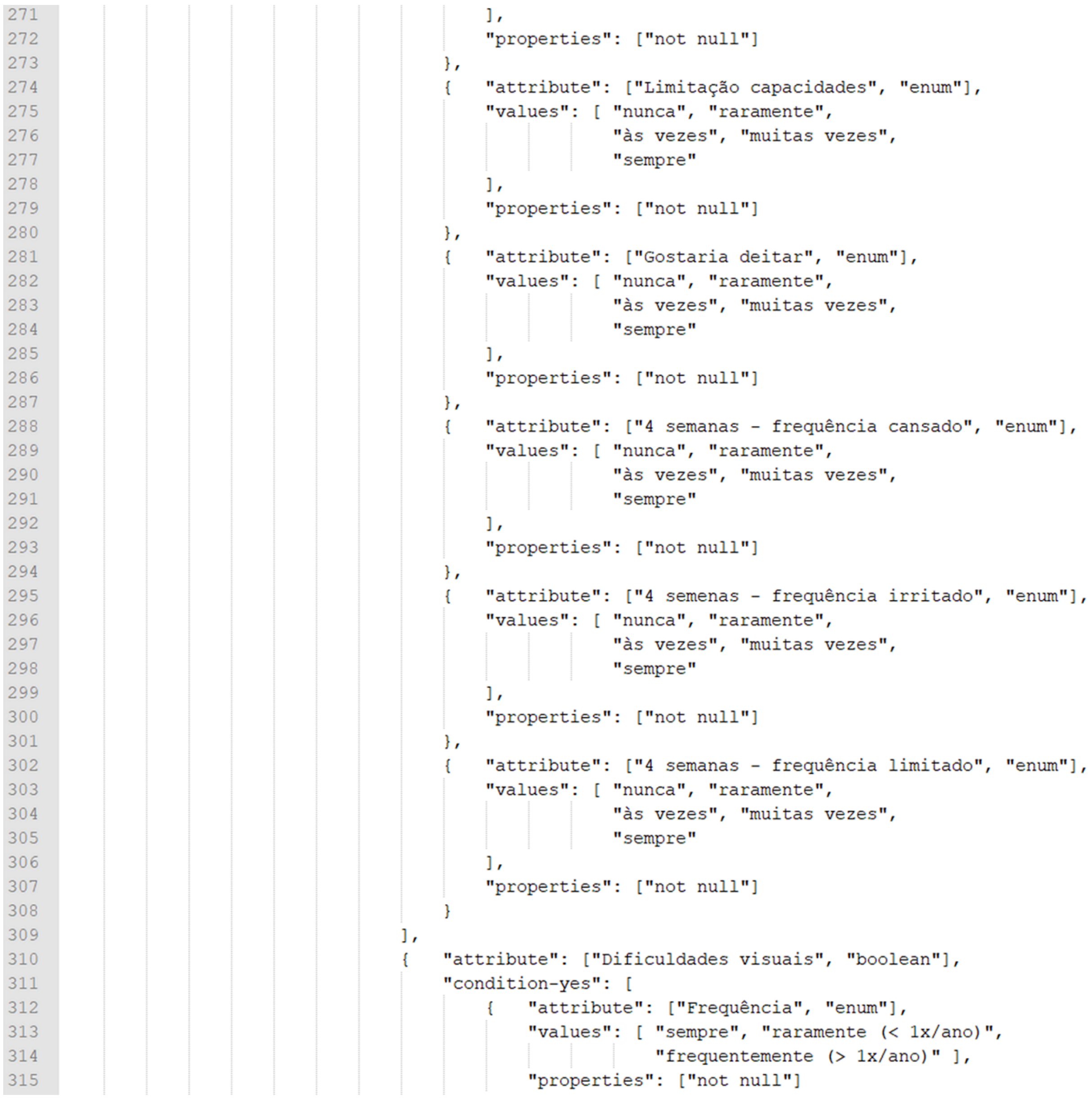
```
                                                    "properties": ["not null"] }
                                            ]
                                        }
                            ],
                            "properties": ["not null"]
                        },
                        {   "attribute": ["Fibrilação Auricular", "enum"],
                            "values": [ "Não", "Paroxística",
                                        "Persistente", "Permanente"],
                            "properties": ["not null"]
                        },
                        {   "attribute": ["Terapêutica Hormonal", "enum"],
                            "values": [ "Não", "Actual",
                                        "Passado"],
                            "properties": ["not null"]
                        },
                        {   "attribute": ["Actividade Física", "enum"],
                            "values": [ "Não",
                                        { "Sim": [
                                                {   "attribute": ["Tipos", "list[enum]"],
                                                    "values": [
                                                        "Exercícios aeróbicos ou de resistência (ex. bicicleta, corrida)",
                                                        "Exercícios de força (ex. musculação)",
                                                        "Exercícios de extensão de movimento (ex. dança)"
                                                    ]
                                                },
                                                {   "attribute": ["N° horas/semana", "integer"],
                                                    "min": "0" },
                                                {   "attribute": ["N° sessões/semana", "integer"],
                                                    "min": "0" }
                                            ]
                                        },
                                        "Desconhecido"
                            ]
                        },
                        { "attribute": ["Outros Antecedentes", "string"] },
                        {   "attribute": ["Avaliação Rankin Prévia - Rankin (MRS)", "enum"],
                            "values": ["0", "1", "2", "3", "4", "5", "6"],
                            "properties": ["not null"] },
                        {   "attribute": ["Medicação", "list[enum]"],
                            "values": [ "AINE", "AINEs", "ARA", "Acido acetilsalicílico",
                                        "Agonistas da dopa", "Antagonista canal de cálcio",
                                        "Antagonista da vitamina K", "Anticoagulantes orais",
                                        "Anticorpo monoclonal", "Antidepressivo outro",
                                        "Antidepressivo tricíclico", "Antidiabéticos orais",
```

Figure C.5: PRECISE Stroke - JSON file specification 5/57.



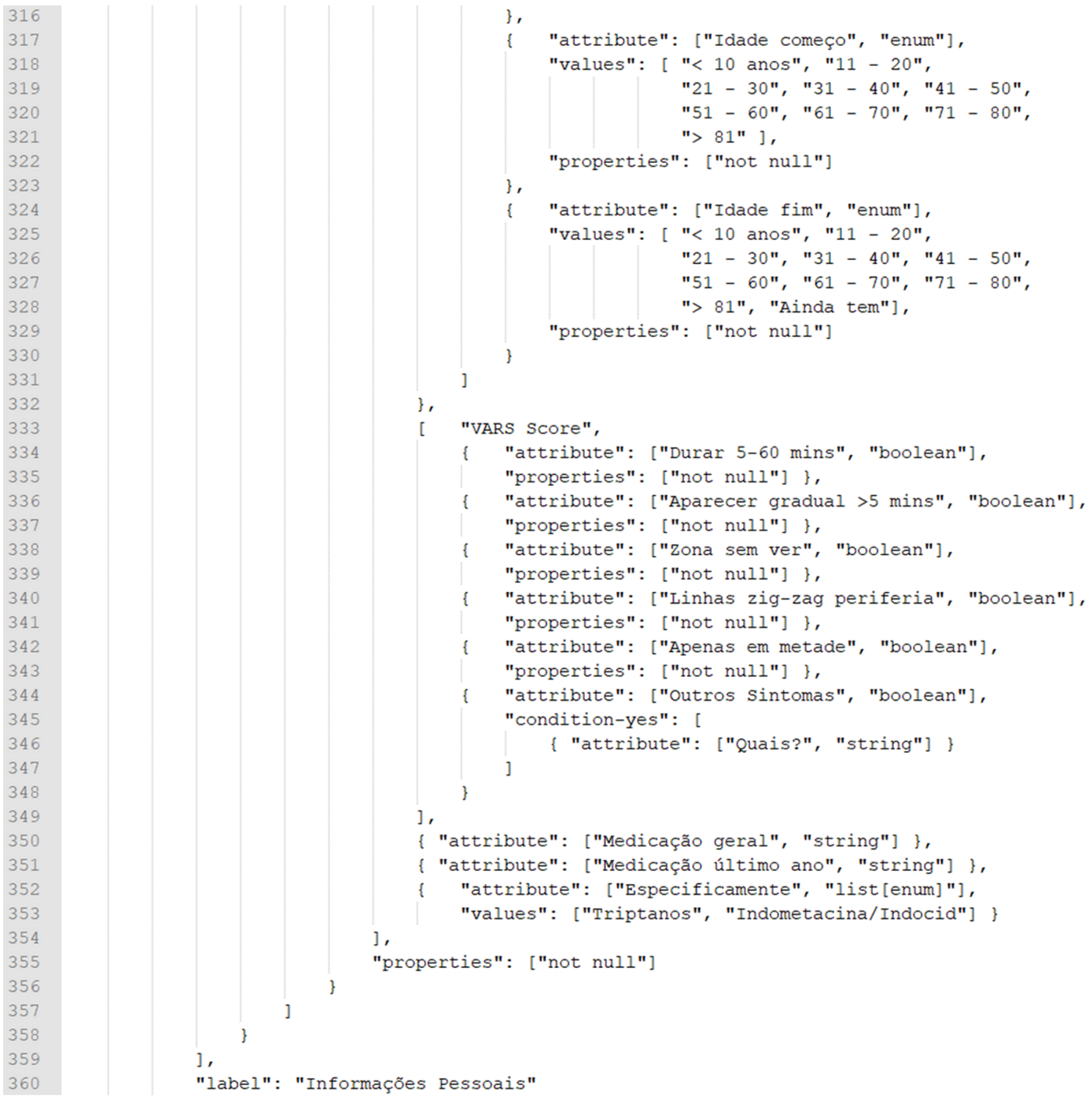
```
                        "Apixabano", "B-bloqueante", "B2-agonistas",
                        "Benzodiazepínicos", "Carbamazepina",
                        "Clopidogrel", "Corticóide gel", "Corticóides inalados",
                        "Corticóides orais", "Dabigatrano", "Diurético",
                        "Edoxabano", "Escitalopram", "Espironolactona",
                        "Estatina", "Fenitoína", "Fenobarbital", "Fluoxetina",
                        "Gabapentina", "IECA", "Insulina", "LMWH/Enoxaparina",
                        "Lamotrigina", "Levetiracetam", "Levodopa",
                        "Neuroléptico", "Nitrato", "Opioide", "Paroxetina",
                        "Pentoxifilina", "Pregabalina", "Primidona",
                        "Rivaroxabano", "Sertralina", "Ticagrelor",
                        "Ticlopidina", "Topiramato", "Trazodona",
                        "Trazodone", "Triflusal", "Valproato de sódio",
                        "Venlafaxina", "Zolpidem"]
        }
    ]
},
{ "entity": [
        "Cefaleias - Antecedentes",
        {   "attribute": ["Dores de cabeça regulares", "boolean"],
            "condition-yes": [
                {   "attribute": ["Idade começo", "enum"],
                    "values": [ "< 10 anos", "11 - 20",
                                "21 - 30", "31 - 40", "41 - 50",
                                "51 - 60", "61 - 70", "71 - 80",
                                "> 81" ],
                    "properties": ["not null"]
                },
                {   "attribute": ["Idade fim", "enum"],
                    "values": [ "< 10 anos", "11 - 20",
                                "21 - 30", "31 - 40", "41 - 50",
                                "51 - 60", "61 - 70", "71 - 80",
                                "> 81", "Ainda tem"],
                    "properties": ["not null"]
                },
                {   "attribute": ["ID-MIGRAINE", "list[enum]"],
                    "values": [ "Nauseado/mal-disposto",
                                "Luz incomodou",
                                "Limitou capacidades" ]
                },
                [   "HIT-6",
                    {   "attribute": ["Dor forte", "enum"],
                        "values": [ "nunca", "raramente",
                                    "às vezes", "muitas vezes",
                                    "sempre"
```

Figure C.6: PRECISE Stroke - JSON file specification 6/57.



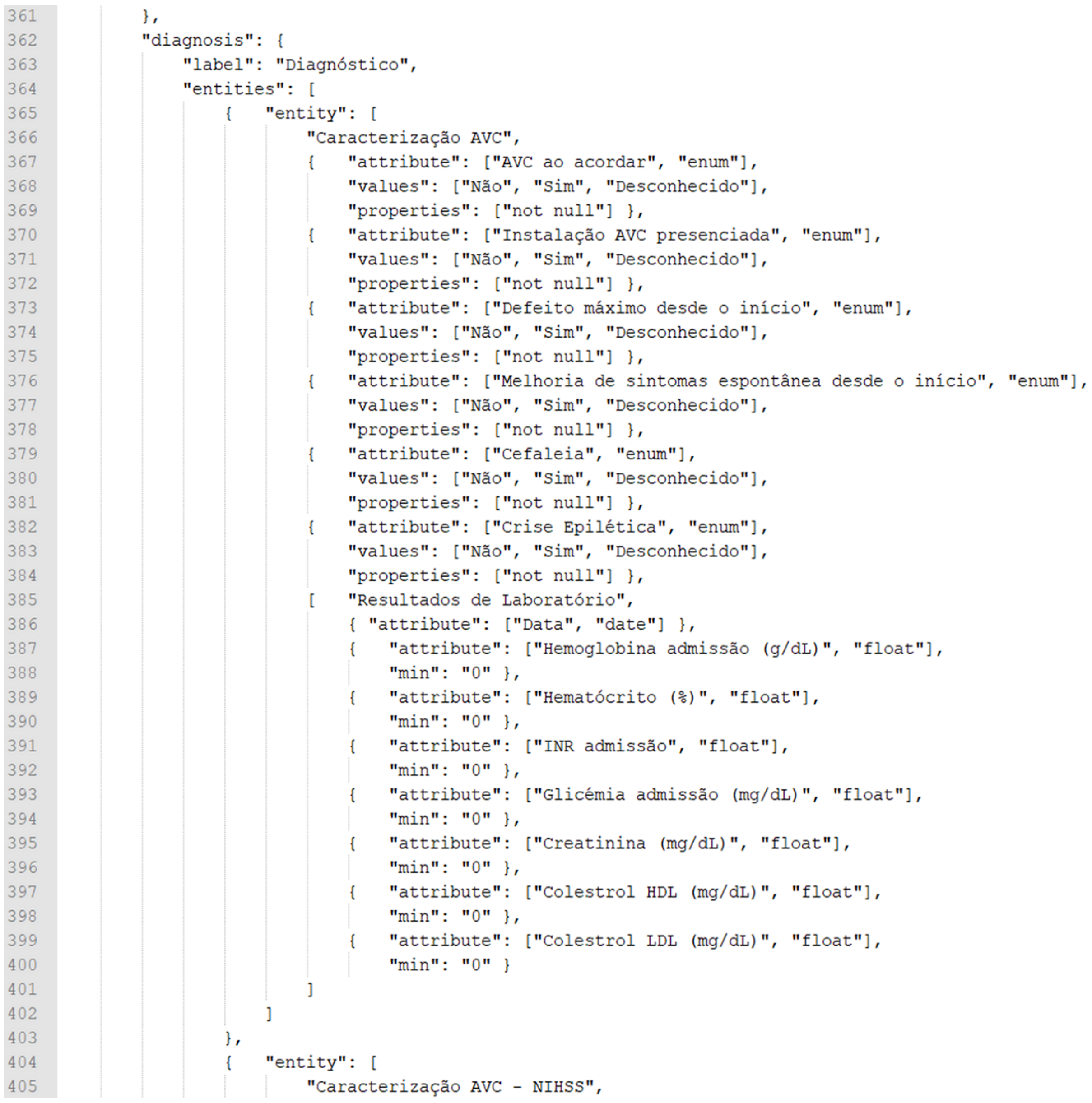
```
                                        ],
                                        "properties": ["not null"]
                                    },
                                    {   "attribute": ["Limitação capacidades", "enum"],
                                        "values": [ "nunca", "raramente",
                                                    "às vezes", "muitas vezes",
                                                    "sempre"
                                        ],
                                        "properties": ["not null"]
                                    },
                                    {   "attribute": ["Gostaria deitar", "enum"],
                                        "values": [ "nunca", "raramente",
                                                    "às vezes", "muitas vezes",
                                                    "sempre"
                                        ],
                                        "properties": ["not null"]
                                    },
                                    {   "attribute": ["4 semanas - frequência cansado", "enum"],
                                        "values": [ "nunca", "raramente",
                                                    "às vezes", "muitas vezes",
                                                    "sempre"
                                        ],
                                        "properties": ["not null"]
                                    },
                                    {   "attribute": ["4 semenas - frequência irritado", "enum"],
                                        "values": [ "nunca", "raramente",
                                                    "às vezes", "muitas vezes",
                                                    "sempre"
                                        ],
                                        "properties": ["not null"]
                                    },
                                    {   "attribute": ["4 semanas - frequência limitado", "enum"],
                                        "values": [ "nunca", "raramente",
                                                    "às vezes", "muitas vezes",
                                                    "sempre"
                                        ],
                                        "properties": ["not null"]
                                    }
                                ],
                                {   "attribute": ["Dificuldades visuais", "boolean"],
                                    "condition-yes": [
                                        {   "attribute": ["Frequência", "enum"],
                                            "values": [ "sempre", "raramente (< 1x/ano)",
                                                        "frequentemente (> 1x/ano)" ],
                                            "properties": ["not null"]
```

Figure C.7: PRECISE Stroke - JSON file specification 7/57.



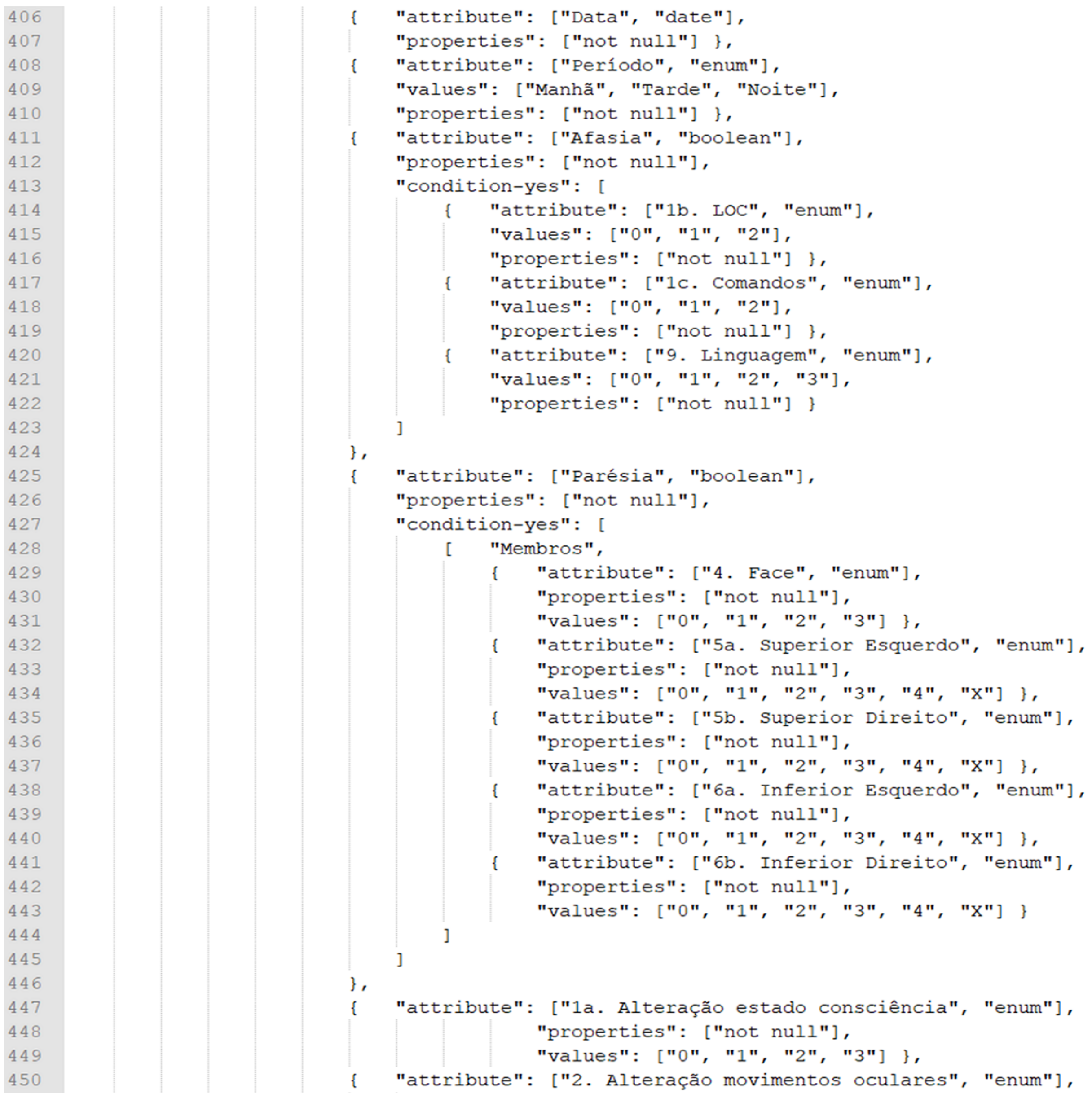
```
                            },
                            {   "attribute": ["Idade começo", "enum"],
                                "values": [ "< 10 anos", "11 - 20",
                                            "21 - 30", "31 - 40", "41 - 50",
                                            "51 - 60", "61 - 70", "71 - 80",
                                            "> 81" ],
                                "properties": ["not null"]
                            },
                            {   "attribute": ["Idade fim", "enum"],
                                "values": [ "< 10 anos", "11 - 20",
                                            "21 - 30", "31 - 40", "41 - 50",
                                            "51 - 60", "61 - 70", "71 - 80",
                                            "> 81", "Ainda tem"],
                                "properties": ["not null"]
                            }
                        ]
                    },
                    [   "VARS Score",
                        {   "attribute": ["Durar 5-60 mins", "boolean"],
                            "properties": ["not null"] },
                        {   "attribute": ["Aparecer gradual >5 mins", "boolean"],
                            "properties": ["not null"] },
                        {   "attribute": ["Zona sem ver", "boolean"],
                            "properties": ["not null"] },
                        {   "attribute": ["Linhas zig-zag periferia", "boolean"],
                            "properties": ["not null"] },
                        {   "attribute": ["Apenas em metade", "boolean"],
                            "properties": ["not null"] },
                        {   "attribute": ["Outros Sintomas", "boolean"],
                            "condition-yes": [
                                { "attribute": ["Quais?", "string"] }
                            ]
                        }
                    ],
                    { "attribute": ["Medicação geral", "string"] },
                    { "attribute": ["Medicação último ano", "string"] },
                    {   "attribute": ["Especificamente", "list[enum]"],
                        "values": ["Triptanos", "Indometacina/Indocid"] }
                ],
                "properties": ["not null"]
            }
        ]
    }
],
"label": "Informações Pessoais"
```

Figure C.8: PRECISE Stroke - JSON file specification 8/57.



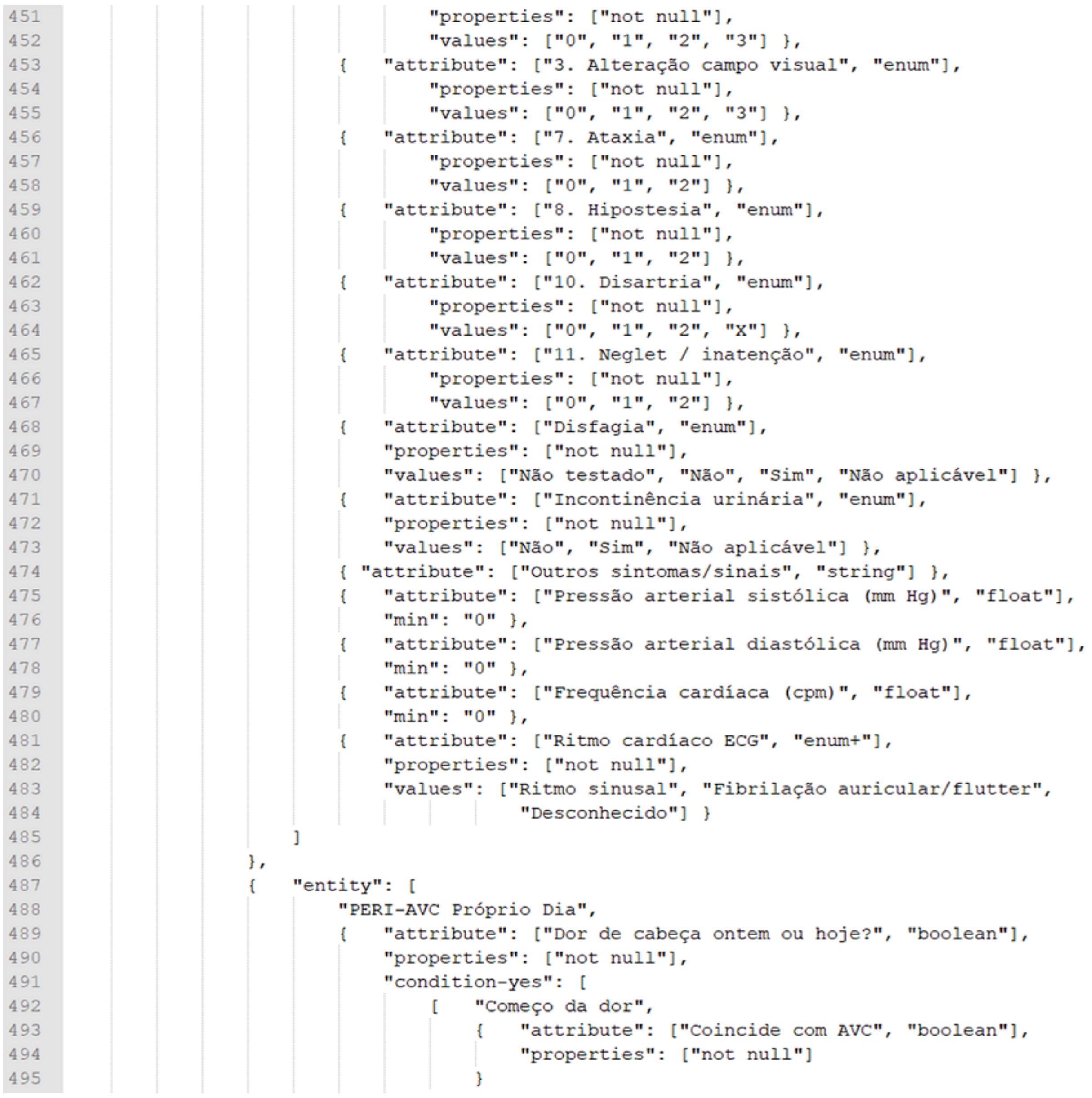
```
        },
        "diagnosis": {
            "label": "Diagnóstico",
            "entities": [
                {   "entity": [
                        "Caracterização AVC",
                        {   "attribute": ["AVC ao acordar", "enum"],
                            "values": ["Não", "Sim", "Desconhecido"],
                            "properties": ["not null"] },
                        {   "attribute": ["Instalação AVC presenciada", "enum"],
                            "values": ["Não", "Sim", "Desconhecido"],
                            "properties": ["not null"] },
                        {   "attribute": ["Defeito máximo desde o início", "enum"],
                            "values": ["Não", "Sim", "Desconhecido"],
                            "properties": ["not null"] },
                        {   "attribute": ["Melhoria de sintomas espontânea desde o início", "enum"],
                            "values": ["Não", "Sim", "Desconhecido"],
                            "properties": ["not null"] },
                        {   "attribute": ["Cefaleia", "enum"],
                            "values": ["Não", "Sim", "Desconhecido"],
                            "properties": ["not null"] },
                        {   "attribute": ["Crise Epilética", "enum"],
                            "values": ["Não", "Sim", "Desconhecido"],
                            "properties": ["not null"] },
                        [   "Resultados de Laboratório",
                            { "attribute": ["Data", "date"] },
                            {   "attribute": ["Hemoglobina admissão (g/dL)", "float"],
                                "min": "0" },
                            {   "attribute": ["Hematócrito (%)", "float"],
                                "min": "0" },
                            {   "attribute": ["INR admissão", "float"],
                                "min": "0" },
                            {   "attribute": ["Glicémia admissão (mg/dL)", "float"],
                                "min": "0" },
                            {   "attribute": ["Creatinina (mg/dL)", "float"],
                                "min": "0" },
                            {   "attribute": ["Colestrol HDL (mg/dL)", "float"],
                                "min": "0" },
                            {   "attribute": ["Colestrol LDL (mg/dL)", "float"],
                                "min": "0" }
                        ]
                    ]
                },
                {   "entity": [
                        "Caracterização AVC - NIHSS",
```

Figure C.9: PRECISE Stroke - JSON file specification 9/57.



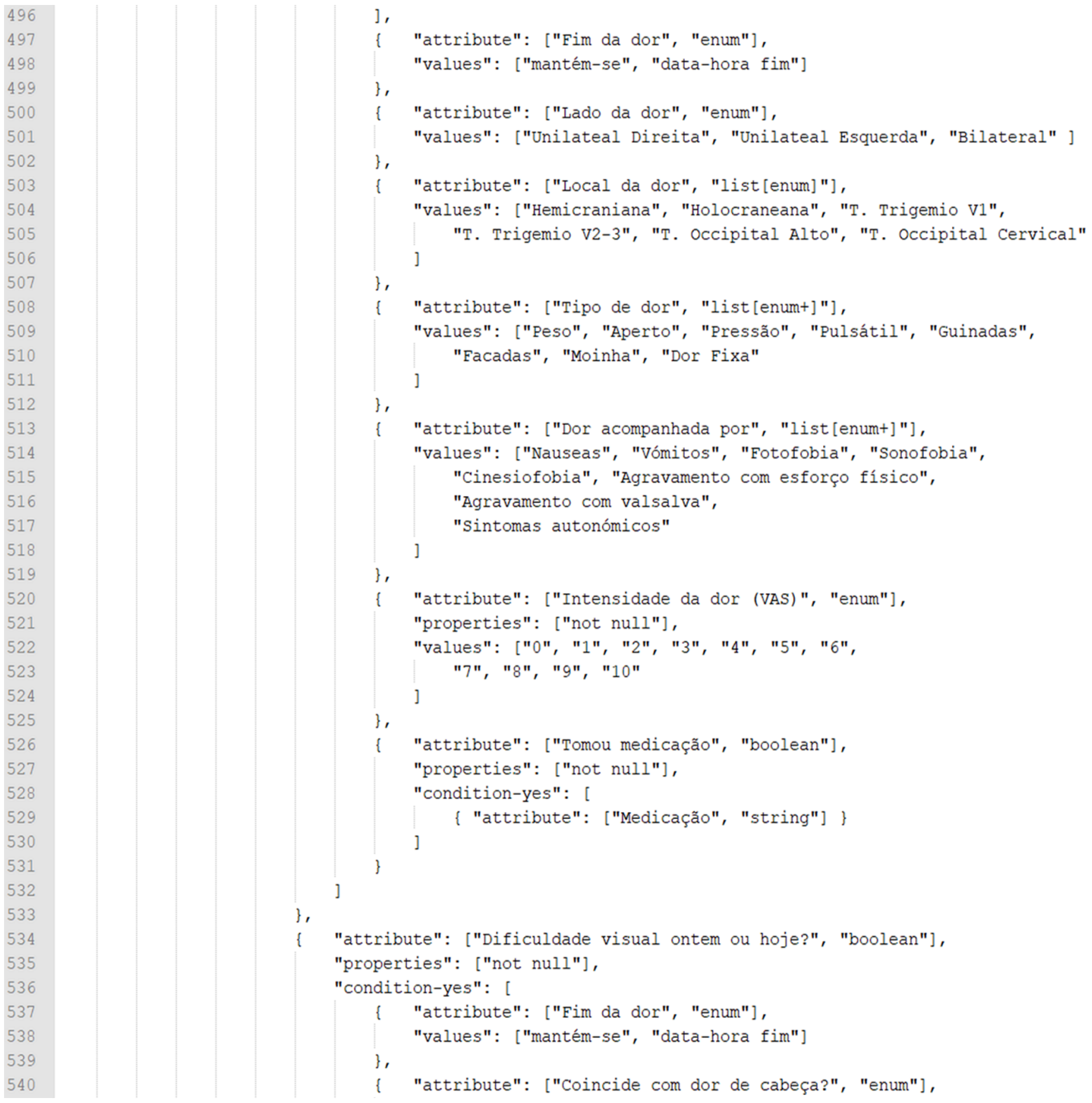
```
{   "attribute": ["Data", "date"],
    "properties": ["not null"] },
{   "attribute": ["Período", "enum"],
    "values": ["Manhã", "Tarde", "Noite"],
    "properties": ["not null"] },
{   "attribute": ["Afasia", "boolean"],
    "properties": ["not null"],
    "condition-yes": [
        {   "attribute": ["1b. LOC", "enum"],
            "values": ["0", "1", "2"],
            "properties": ["not null"] },
        {   "attribute": ["1c. Comandos", "enum"],
            "values": ["0", "1", "2"],
            "properties": ["not null"] },
        {   "attribute": ["9. Linguagem", "enum"],
            "values": ["0", "1", "2", "3"],
            "properties": ["not null"] }
    ]
},
{   "attribute": ["Parésia", "boolean"],
    "properties": ["not null"],
    "condition-yes": [
        [   "Membros",
            {   "attribute": ["4. Face", "enum"],
                "properties": ["not null"],
                "values": ["0", "1", "2", "3"] },
            {   "attribute": ["5a. Superior Esquerdo", "enum"],
                "properties": ["not null"],
                "values": ["0", "1", "2", "3", "4", "X"] },
            {   "attribute": ["5b. Superior Direito", "enum"],
                "properties": ["not null"],
                "values": ["0", "1", "2", "3", "4", "X"] },
            {   "attribute": ["6a. Inferior Esquerdo", "enum"],
                "properties": ["not null"],
                "values": ["0", "1", "2", "3", "4", "X"] },
            {   "attribute": ["6b. Inferior Direito", "enum"],
                "properties": ["not null"],
                "values": ["0", "1", "2", "3", "4", "X"] }
        ]
    ]
},
{   "attribute": ["1a. Alteração estado consciência", "enum"],
                "properties": ["not null"],
                "values": ["0", "1", "2", "3"] },
{   "attribute": ["2. Alteração movimentos oculares", "enum"],
```

Figure C.10: PRECISE Stroke - JSON file specification 10/57.



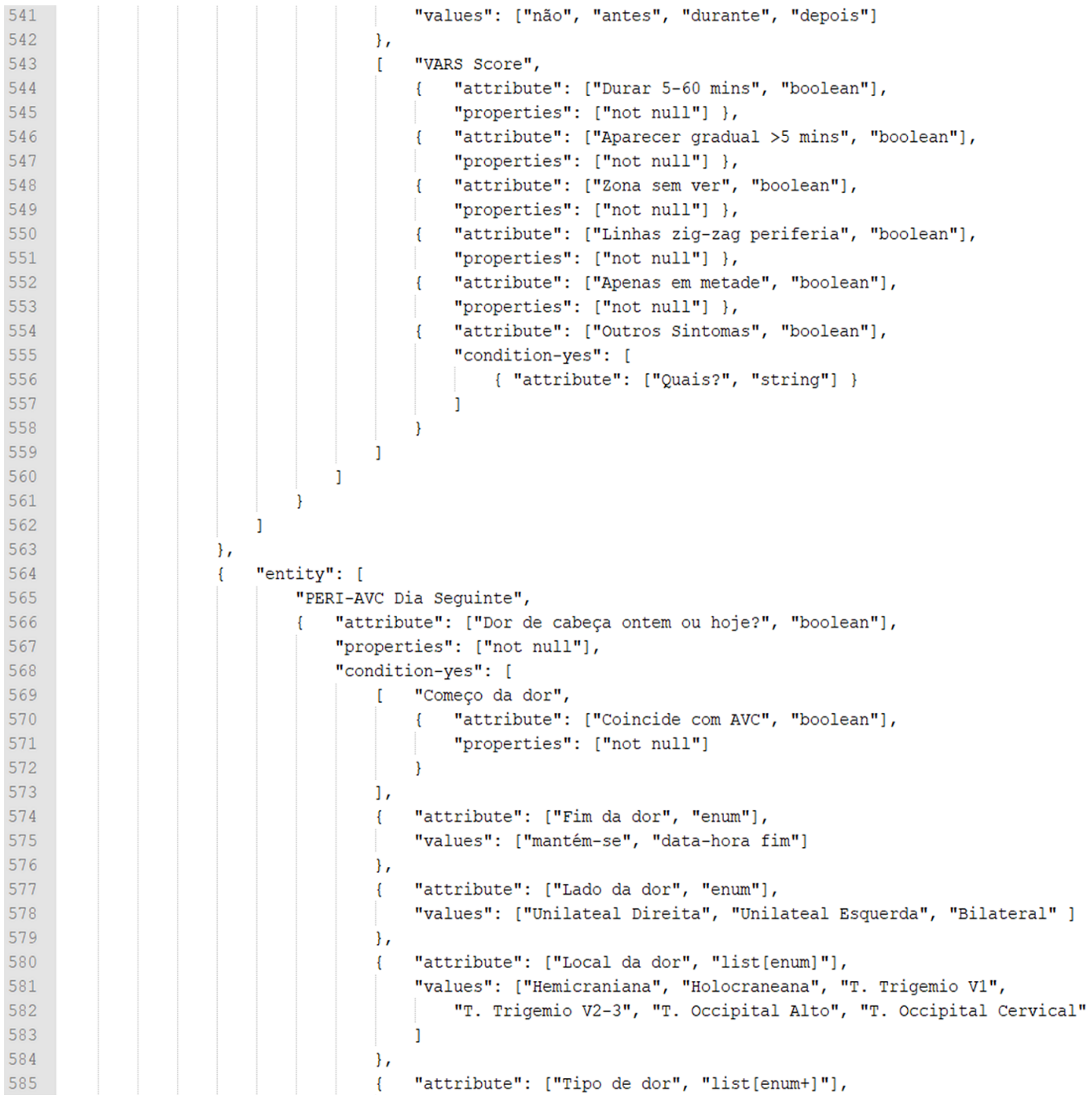
```
                "properties": ["not null"],
                "values": ["0", "1", "2", "3"] },
        {   "attribute": ["3. Alteração campo visual", "enum"],
                "properties": ["not null"],
                "values": ["0", "1", "2", "3"] },
        {   "attribute": ["7. Ataxia", "enum"],
                "properties": ["not null"],
                "values": ["0", "1", "2"] },
        {   "attribute": ["8. Hipostesia", "enum"],
                "properties": ["not null"],
                "values": ["0", "1", "2"] },
        {   "attribute": ["10. Disartria", "enum"],
                "properties": ["not null"],
                "values": ["0", "1", "2", "X"] },
        {   "attribute": ["11. Neglet / inatenção", "enum"],
                "properties": ["not null"],
                "values": ["0", "1", "2"] },
        {   "attribute": ["Disfagia", "enum"],
            "properties": ["not null"],
            "values": ["Não testado", "Não", "Sim", "Não aplicável"] },
        {   "attribute": ["Incontinência urinária", "enum"],
            "properties": ["not null"],
            "values": ["Não", "Sim", "Não aplicável"] },
        { "attribute": ["Outros sintomas/sinais", "string"] },
        {   "attribute": ["Pressão arterial sistólica (mm Hg)", "float"],
            "min": "0" },
        {   "attribute": ["Pressão arterial diastólica (mm Hg)", "float"],
            "min": "0" },
        {   "attribute": ["Frequência cardíaca (cpm)", "float"],
            "min": "0" },
        {   "attribute": ["Ritmo cardíaco ECG", "enum+"],
            "properties": ["not null"],
            "values": ["Ritmo sinusal", "Fibrilação auricular/flutter",
                        "Desconhecido"] }
    ]
},
{   "entity": [
        "PERI-AVC Próprio Dia",
        {   "attribute": ["Dor de cabeça ontem ou hoje?", "boolean"],
            "properties": ["not null"],
            "condition-yes": [
                [   "Começo da dor",
                    {   "attribute": ["Coincide com AVC", "boolean"],
                        "properties": ["not null"]
                    }
```

Figure C.11: PRECISE Stroke - JSON file specification 11/57.



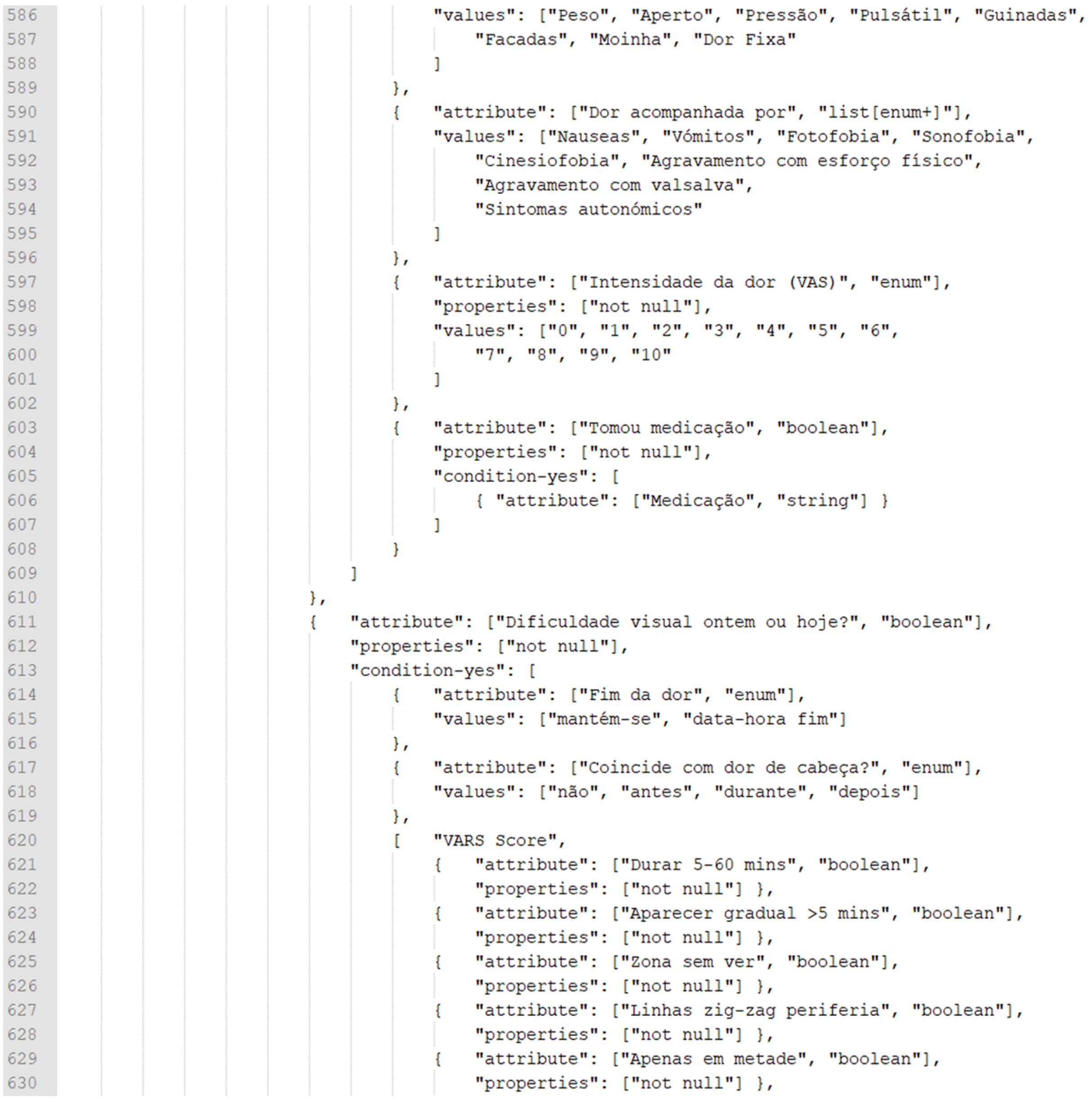
```
        ],
        {   "attribute": ["Fim da dor", "enum"],
            "values": ["mantém-se", "data-hora fim"]
        },
        {   "attribute": ["Lado da dor", "enum"],
            "values": ["Unilateal Direita", "Unilateal Esquerda", "Bilateral" ]
        },
        {   "attribute": ["Local da dor", "list[enum]"],
            "values": ["Hemicraniana", "Holocraneana", "T. Trigemio V1",
                "T. Trigemio V2-3", "T. Occipital Alto", "T. Occipital Cervical"
            ]
        },
        {   "attribute": ["Tipo de dor", "list[enum+]"],
            "values": ["Peso", "Aperto", "Pressão", "Pulsátil", "Guinadas",
                "Facadas", "Moinha", "Dor Fixa"
            ]
        },
        {   "attribute": ["Dor acompanhada por", "list[enum+]"],
            "values": ["Nauseas", "Vómitos", "Fotofobia", "Sonofobia",
                "Cinesiofobia", "Agravamento com esforço físico",
                "Agravamento com valsalva",
                "Sintomas autonómicos"
            ]
        },
        {   "attribute": ["Intensidade da dor (VAS)", "enum"],
            "properties": ["not null"],
            "values": ["0", "1", "2", "3", "4", "5", "6",
                "7", "8", "9", "10"
            ]
        },
        {   "attribute": ["Tomou medicação", "boolean"],
            "properties": ["not null"],
            "condition-yes": [
                { "attribute": ["Medicação", "string"] }
            ]
        }
    ]
},
{   "attribute": ["Dificuldade visual ontem ou hoje?", "boolean"],
    "properties": ["not null"],
    "condition-yes": [
        {   "attribute": ["Fim da dor", "enum"],
            "values": ["mantém-se", "data-hora fim"]
        },
        {   "attribute": ["Coincide com dor de cabeça?", "enum"],
```

Figure C.12: PRECISE Stroke - JSON file specification 12/57.



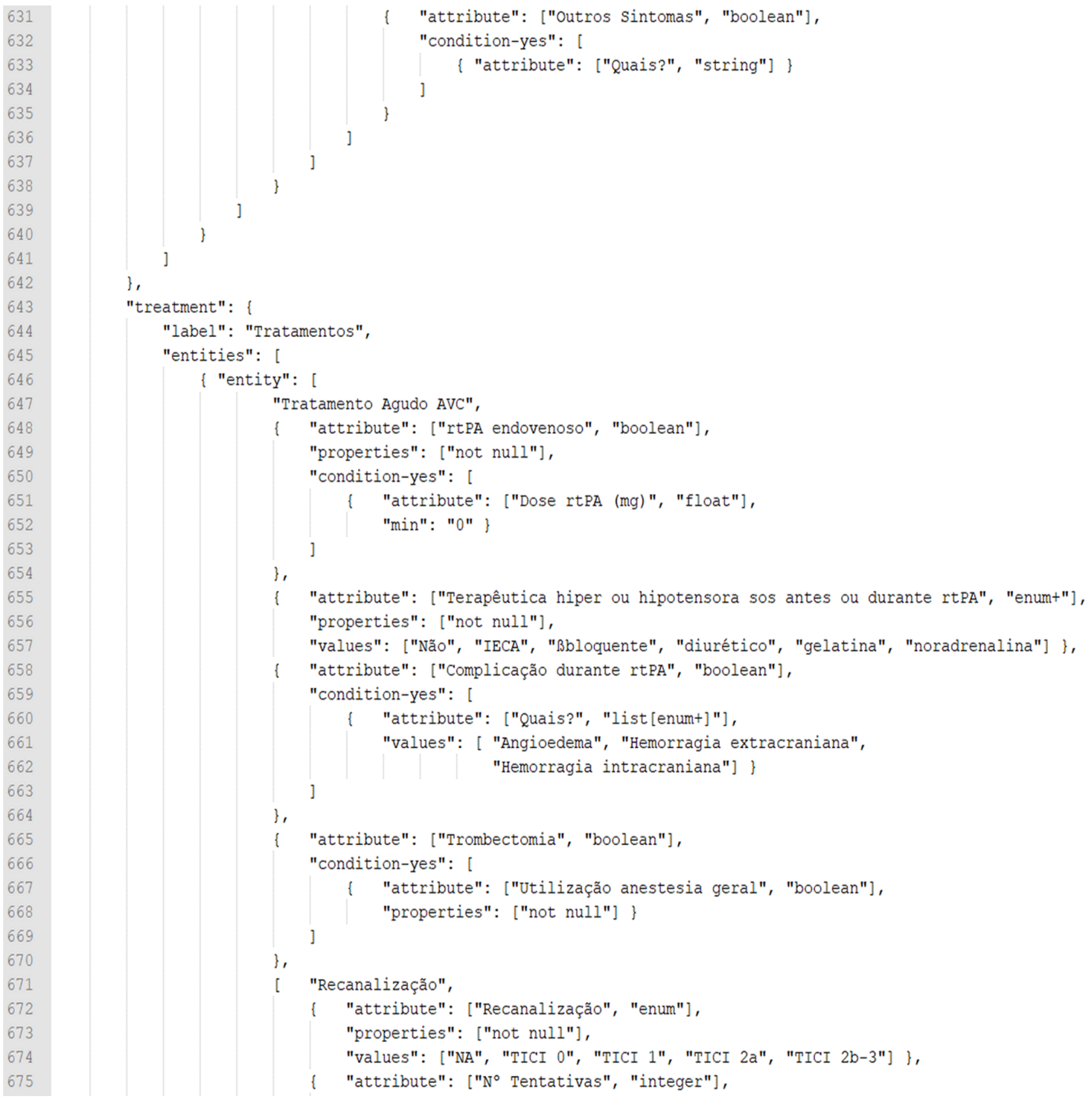
```
                                    "values": ["não", "antes", "durante", "depois"]
                                },
                                [   "VARS Score",
                                    {   "attribute": ["Durar 5-60 mins", "boolean"],
                                        "properties": ["not null"] },
                                    {   "attribute": ["Aparecer gradual >5 mins", "boolean"],
                                        "properties": ["not null"] },
                                    {   "attribute": ["Zona sem ver", "boolean"],
                                        "properties": ["not null"] },
                                    {   "attribute": ["Linhas zig-zag periferia", "boolean"],
                                        "properties": ["not null"] },
                                    {   "attribute": ["Apenas em metade", "boolean"],
                                        "properties": ["not null"] },
                                    {   "attribute": ["Outros Sintomas", "boolean"],
                                        "condition-yes": [
                                            { "attribute": ["Quais?", "string"] }
                                        ]
                                    }
                                ]
                            ]
                        }
                    ]
                },
                {   "entity": [
                        "PERI-AVC Dia Seguinte",
                        {   "attribute": ["Dor de cabeça ontem ou hoje?", "boolean"],
                            "properties": ["not null"],
                            "condition-yes": [
                                [   "Começo da dor",
                                    {   "attribute": ["Coincide com AVC", "boolean"],
                                        "properties": ["not null"]
                                    }
                                ],
                                {   "attribute": ["Fim da dor", "enum"],
                                    "values": ["mantém-se", "data-hora fim"]
                                },
                                {   "attribute": ["Lado da dor", "enum"],
                                    "values": ["Unilateal Direita", "Unilateal Esquerda", "Bilateral" ]
                                },
                                {   "attribute": ["Local da dor", "list[enum]"],
                                    "values": ["Hemicraniana", "Holocraneana", "T. Trigemio V1",
                                        "T. Trigemio V2-3", "T. Occipital Alto", "T. Occipital Cervical"
                                    ]
                                },
                                {   "attribute": ["Tipo de dor", "list[enum+]"],
```

Figure C.13: PRECISE Stroke - JSON file specification 13/57.



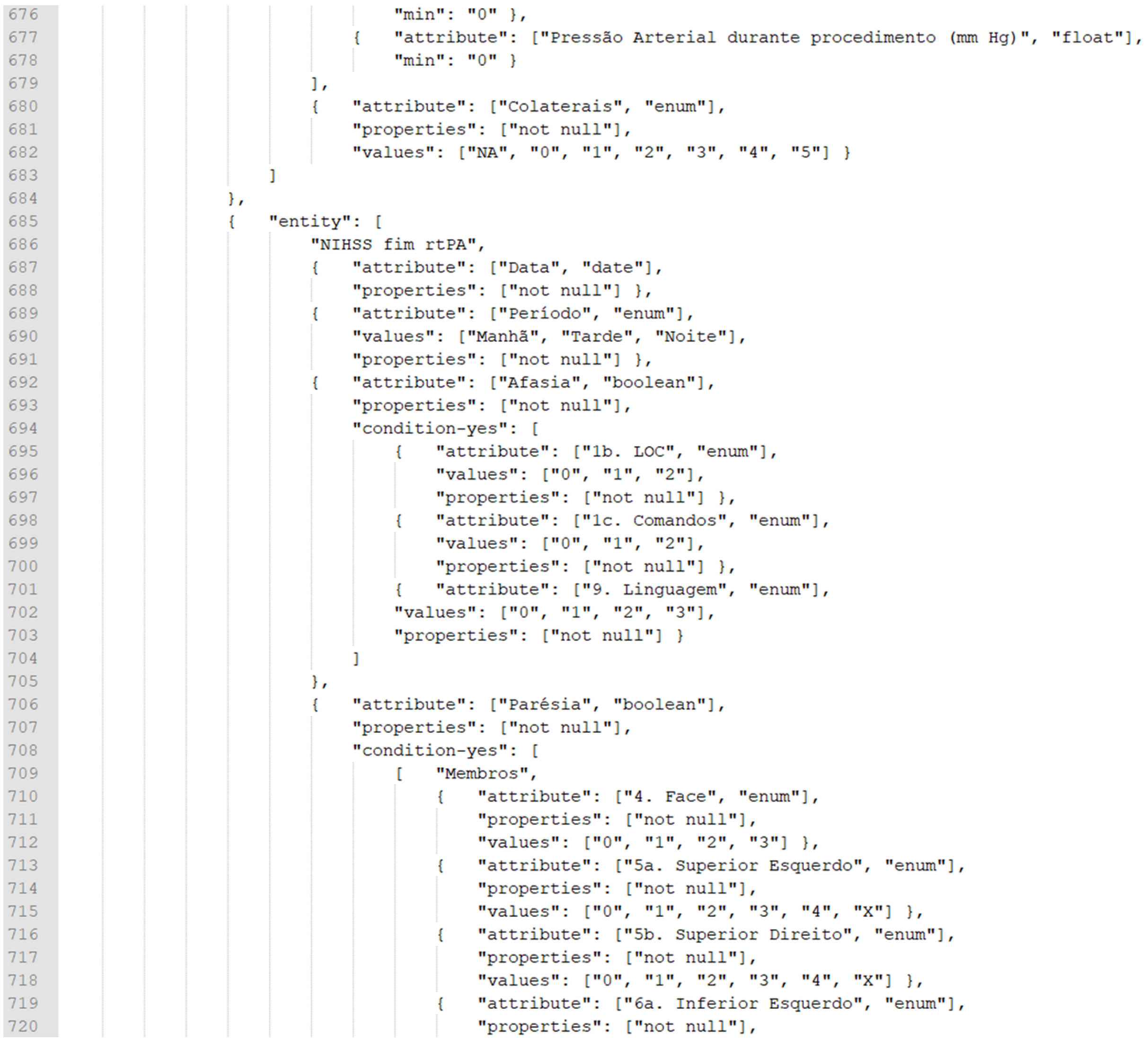
```
                                    "values": ["Peso", "Aperto", "Pressão", "Pulsátil", "Guinadas",
                                        "Facadas", "Moinha", "Dor Fixa"
                                    ]
                                },
                                {   "attribute": ["Dor acompanhada por", "list[enum+]"],
                                    "values": ["Nauseas", "Vómitos", "Fotofobia", "Sonofobia",
                                        "Cinesiofobia", "Agravamento com esforço físico",
                                        "Agravamento com valsalva",
                                        "Sintomas autonómicos"
                                    ]
                                },
                                {   "attribute": ["Intensidade da dor (VAS)", "enum"],
                                    "properties": ["not null"],
                                    "values": ["0", "1", "2", "3", "4", "5", "6",
                                        "7", "8", "9", "10"
                                    ]
                                },
                                {   "attribute": ["Tomou medicação", "boolean"],
                                    "properties": ["not null"],
                                    "condition-yes": [
                                        { "attribute": ["Medicação", "string"] }
                                    ]
                                }
                            ]
                        },
                        {   "attribute": ["Dificuldade visual ontem ou hoje?", "boolean"],
                            "properties": ["not null"],
                            "condition-yes": [
                                {   "attribute": ["Fim da dor", "enum"],
                                    "values": ["mantém-se", "data-hora fim"]
                                },
                                {   "attribute": ["Coincide com dor de cabeça?", "enum"],
                                    "values": ["não", "antes", "durante", "depois"]
                                },
                                [   "VARS Score",
                                    {   "attribute": ["Durar 5-60 mins", "boolean"],
                                        "properties": ["not null"] },
                                    {   "attribute": ["Aparecer gradual >5 mins", "boolean"],
                                        "properties": ["not null"] },
                                    {   "attribute": ["Zona sem ver", "boolean"],
                                        "properties": ["not null"] },
                                    {   "attribute": ["Linhas zig-zag periferia", "boolean"],
                                        "properties": ["not null"] },
                                    {   "attribute": ["Apenas em metade", "boolean"],
                                        "properties": ["not null"] },
```

Figure C.14: PRECISE Stroke - JSON file specification 14/57.



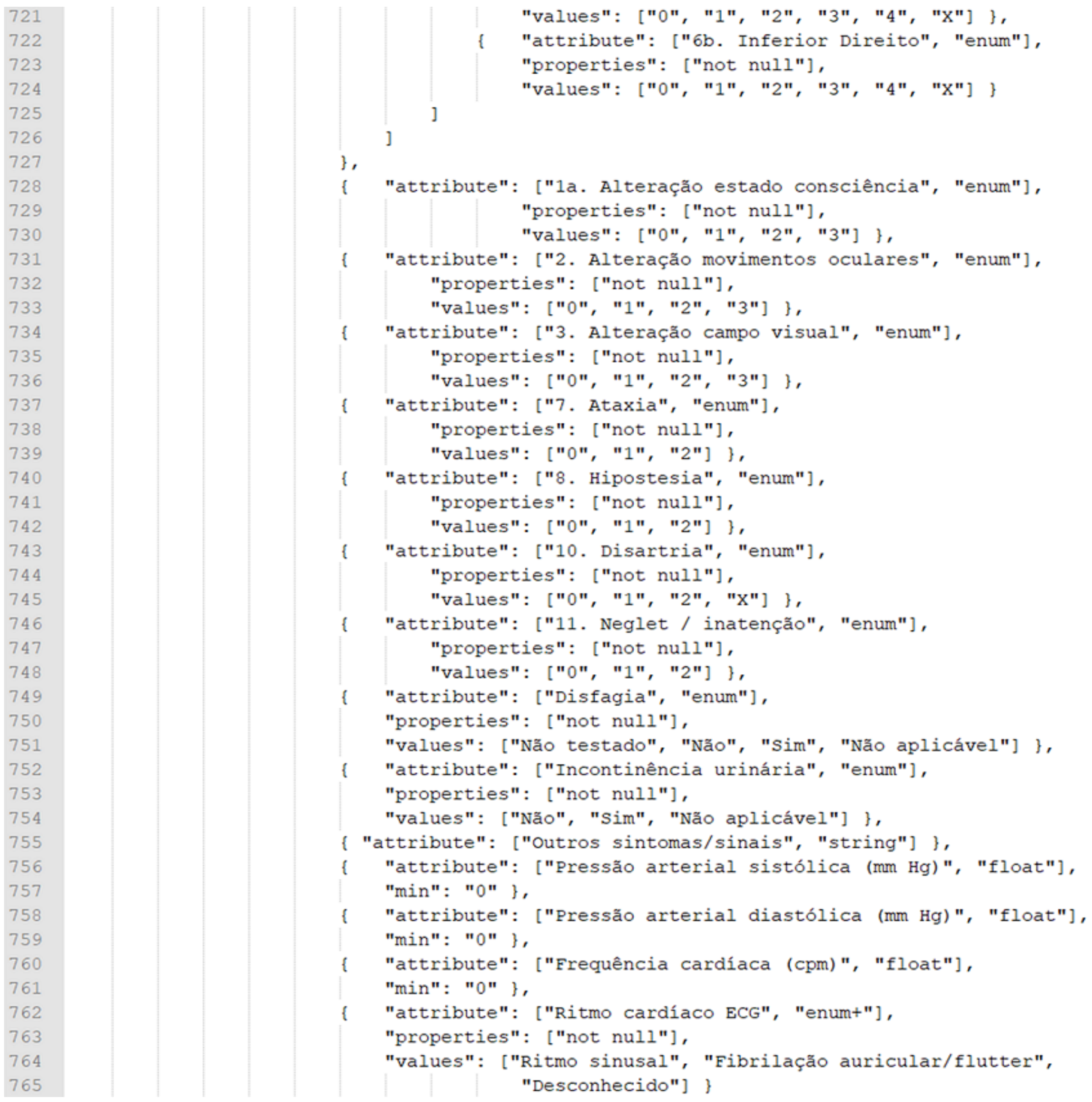
```
                                    {   "attribute": ["Outros Sintomas", "boolean"],
                                        "condition-yes": [
                                            { "attribute": ["Quais?", "string"] }
                                        ]
                                    }
                                ]
                            ]
                        }
                    ]
                }
            ]
        },
        "treatment": {
            "label": "Tratamentos",
            "entities": [
                { "entity": [
                        "Tratamento Agudo AVC",
                        {   "attribute": ["rtPA endovenoso", "boolean"],
                            "properties": ["not null"],
                            "condition-yes": [
                                {   "attribute": ["Dose rtPA (mg)", "float"],
                                    "min": "0" }
                            ]
                        },
                        {   "attribute": ["Terapêutica hiper ou hipotensora sos antes ou durante rtPA", "enum+"],
                            "properties": ["not null"],
                            "values": ["Não", "IECA", "ßbloquente", "diurético", "gelatina", "noradrenalina"] },
                        {   "attribute": ["Complicação durante rtPA", "boolean"],
                            "condition-yes": [
                                {   "attribute": ["Quais?", "list[enum+]"],
                                    "values": [ "Angioedema", "Hemorragia extracraniana",
                                                "Hemorragia intracraniana"] }
                            ]
                        },
                        {   "attribute": ["Trombectomia", "boolean"],
                            "condition-yes": [
                                {   "attribute": ["Utilização anestesia geral", "boolean"],
                                    "properties": ["not null"] }
                            ]
                        },
                        [   "Recanalização",
                            {   "attribute": ["Recanalização", "enum"],
                                "properties": ["not null"],
                                "values": ["NA", "TICI 0", "TICI 1", "TICI 2a", "TICI 2b-3"] },
                            {   "attribute": ["N° Tentativas", "integer"],
```

Figure C.15: PRECISE Stroke - JSON file specification 15/57.



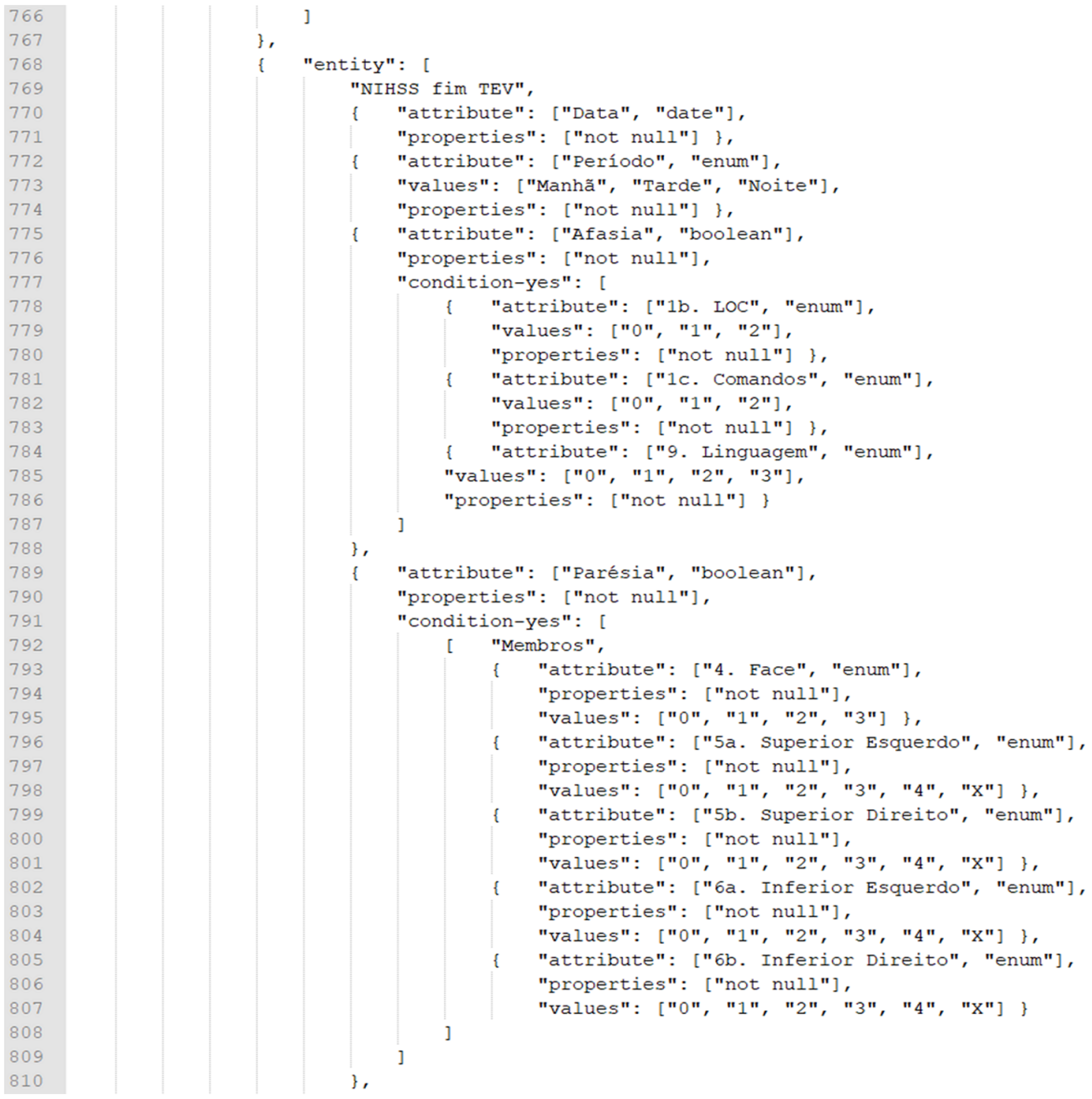
```
                                "min": "0" },
                            {   "attribute": ["Pressão Arterial durante procedimento (mm Hg)", "float"],
                                "min": "0" }
                        ],
                        {   "attribute": ["Colaterais", "enum"],
                            "properties": ["not null"],
                            "values": ["NA", "0", "1", "2", "3", "4", "5"] }
                    ]
                },
                {   "entity": [
                        "NIHSS fim rtPA",
                        {   "attribute": ["Data", "date"],
                            "properties": ["not null"] },
                        {   "attribute": ["Período", "enum"],
                            "values": ["Manhã", "Tarde", "Noite"],
                            "properties": ["not null"] },
                        {   "attribute": ["Afasia", "boolean"],
                            "properties": ["not null"],
                            "condition-yes": [
                                {   "attribute": ["1b. LOC", "enum"],
                                    "values": ["0", "1", "2"],
                                    "properties": ["not null"] },
                                {   "attribute": ["1c. Comandos", "enum"],
                                    "values": ["0", "1", "2"],
                                    "properties": ["not null"] },
                                {   "attribute": ["9. Linguagem", "enum"],
                                "values": ["0", "1", "2", "3"],
                                "properties": ["not null"] }
                            ]
                        },
                        {   "attribute": ["Parésia", "boolean"],
                            "properties": ["not null"],
                            "condition-yes": [
                                [   "Membros",
                                    {   "attribute": ["4. Face", "enum"],
                                        "properties": ["not null"],
                                        "values": ["0", "1", "2", "3"] },
                                    {   "attribute": ["5a. Superior Esquerdo", "enum"],
                                        "properties": ["not null"],
                                        "values": ["0", "1", "2", "3", "4", "X"] },
                                    {   "attribute": ["5b. Superior Direito", "enum"],
                                        "properties": ["not null"],
                                        "values": ["0", "1", "2", "3", "4", "X"] },
                                    {   "attribute": ["6a. Inferior Esquerdo", "enum"],
                                        "properties": ["not null"],
```

Figure C.16: PRECISE Stroke - JSON file specification 16/57.



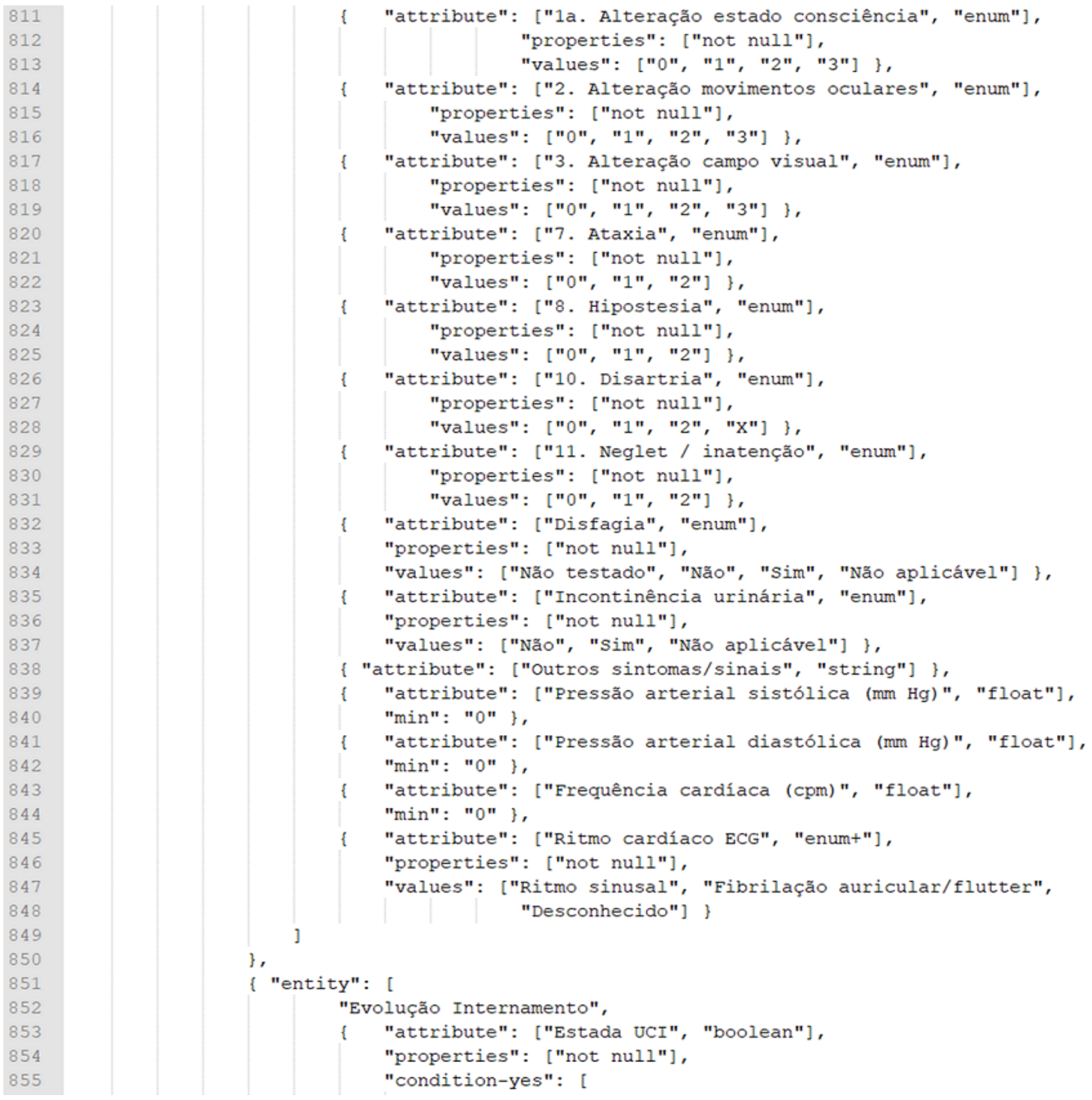
```
                                        "values": ["0", "1", "2", "3", "4", "X"] },
                                    {   "attribute": ["6b. Inferior Direito", "enum"],
                                        "properties": ["not null"],
                                        "values": ["0", "1", "2", "3", "4", "X"] }
                                ]
                            ]
                        },
                        {   "attribute": ["1a. Alteração estado consciência", "enum"],
                                        "properties": ["not null"],
                                        "values": ["0", "1", "2", "3"] },
                        {   "attribute": ["2. Alteração movimentos oculares", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2", "3"] },
                        {   "attribute": ["3. Alteração campo visual", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2", "3"] },
                        {   "attribute": ["7. Ataxia", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2"] },
                        {   "attribute": ["8. Hipostesia", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2"] },
                        {   "attribute": ["10. Disartria", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2", "X"] },
                        {   "attribute": ["11. Neglet / inatenção", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2"] },
                        {   "attribute": ["Disfagia", "enum"],
                            "properties": ["not null"],
                            "values": ["Não testado", "Não", "Sim", "Não aplicável"] },
                        {   "attribute": ["Incontinência urinária", "enum"],
                            "properties": ["not null"],
                            "values": ["Não", "Sim", "Não aplicável"] },
                        { "attribute": ["Outros sintomas/sinais", "string"] },
                        {   "attribute": ["Pressão arterial sistólica (mm Hg)", "float"],
                            "min": "0" },
                        {   "attribute": ["Pressão arterial diastólica (mm Hg)", "float"],
                            "min": "0" },
                        {   "attribute": ["Frequência cardíaca (cpm)", "float"],
                            "min": "0" },
                        {   "attribute": ["Ritmo cardíaco ECG", "enum+"],
                            "properties": ["not null"],
                            "values": ["Ritmo sinusal", "Fibrilação auricular/flutter",
                                        "Desconhecido"] }
```

Figure C.17: PRECISE Stroke - JSON file specification 17/57.



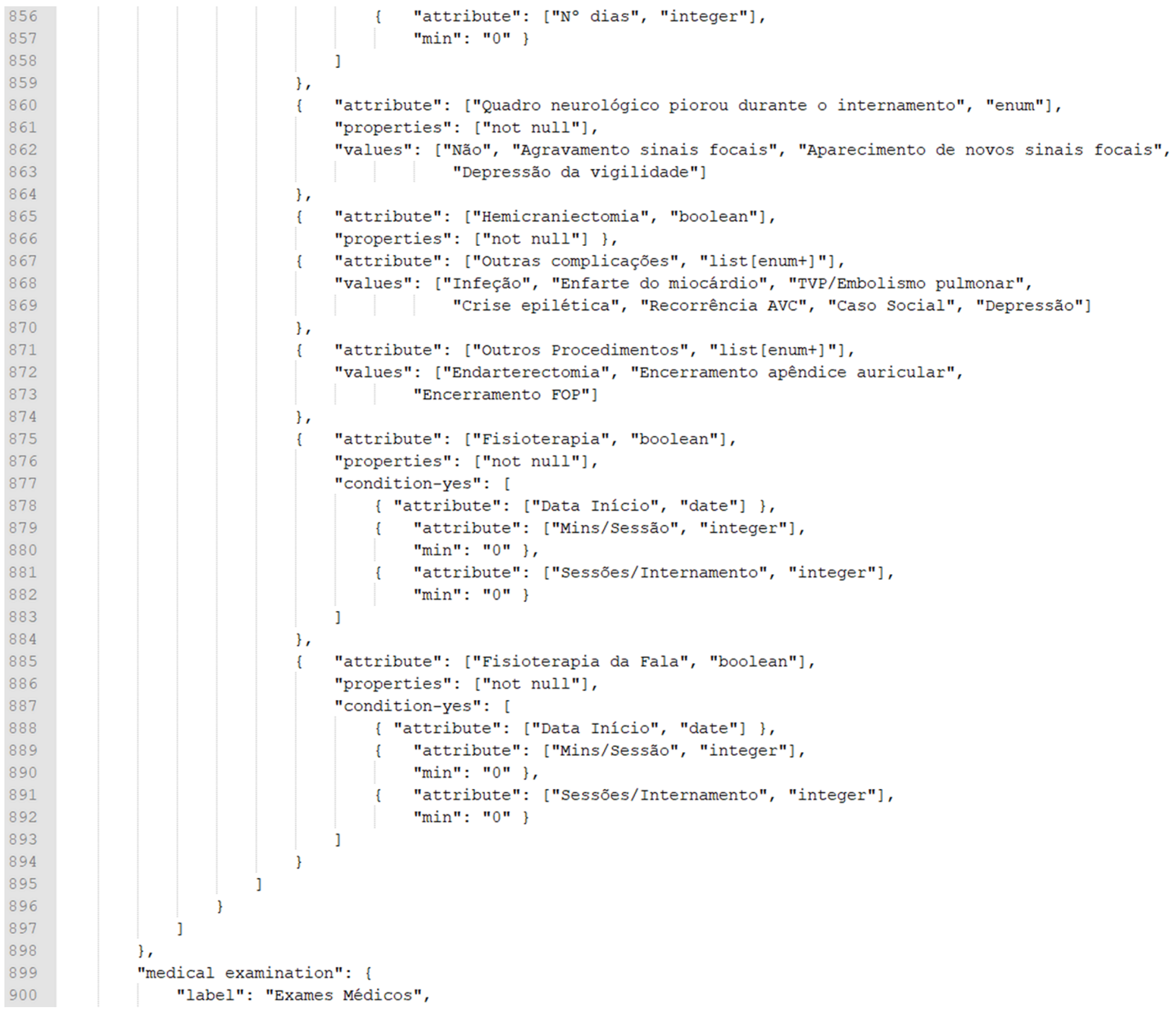
```
    ]
},
{   "entity": [
        "NIHSS fim TEV",
        {   "attribute": ["Data", "date"],
            "properties": ["not null"] },
        {   "attribute": ["Período", "enum"],
            "values": ["Manhã", "Tarde", "Noite"],
            "properties": ["not null"] },
        {   "attribute": ["Afasia", "boolean"],
            "properties": ["not null"],
            "condition-yes": [
                {   "attribute": ["1b. LOC", "enum"],
                    "values": ["0", "1", "2"],
                    "properties": ["not null"] },
                {   "attribute": ["1c. Comandos", "enum"],
                    "values": ["0", "1", "2"],
                    "properties": ["not null"] },
                {   "attribute": ["9. Linguagem", "enum"],
                "values": ["0", "1", "2", "3"],
                "properties": ["not null"] }
            ]
        },
        {   "attribute": ["Parésia", "boolean"],
            "properties": ["not null"],
            "condition-yes": [
                [   "Membros",
                    {   "attribute": ["4. Face", "enum"],
                        "properties": ["not null"],
                        "values": ["0", "1", "2", "3"] },
                    {   "attribute": ["5a. Superior Esquerdo", "enum"],
                        "properties": ["not null"],
                        "values": ["0", "1", "2", "3", "4", "X"] },
                    {   "attribute": ["5b. Superior Direito", "enum"],
                        "properties": ["not null"],
                        "values": ["0", "1", "2", "3", "4", "X"] },
                    {   "attribute": ["6a. Inferior Esquerdo", "enum"],
                        "properties": ["not null"],
                        "values": ["0", "1", "2", "3", "4", "X"] },
                    {   "attribute": ["6b. Inferior Direito", "enum"],
                        "properties": ["not null"],
                        "values": ["0", "1", "2", "3", "4", "X"] }
                ]
            ]
        },
```

Figure C.18: PRECISE Stroke - JSON file specification 18/57.



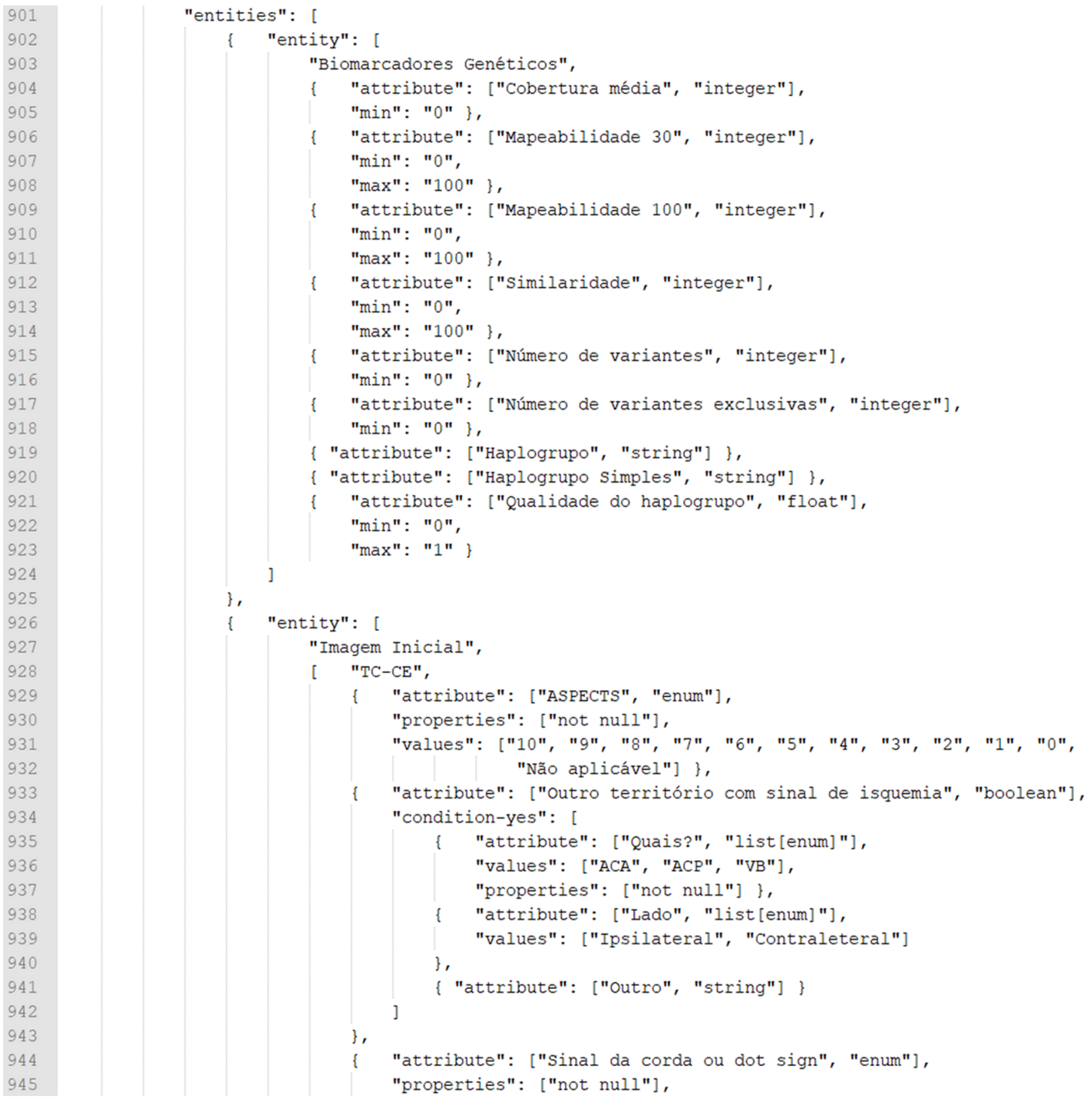
```
                        {   "attribute": ["1a. Alteração estado consciência", "enum"],
                                        "properties": ["not null"],
                                        "values": ["0", "1", "2", "3"] },
                        {   "attribute": ["2. Alteração movimentos oculares", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2", "3"] },
                        {   "attribute": ["3. Alteração campo visual", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2", "3"] },
                        {   "attribute": ["7. Ataxia", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2"] },
                        {   "attribute": ["8. Hipostesia", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2"] },
                        {   "attribute": ["10. Disartria", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2", "X"] },
                        {   "attribute": ["11. Neglet / inatenção", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2"] },
                        {   "attribute": ["Disfagia", "enum"],
                            "properties": ["not null"],
                            "values": ["Não testado", "Não", "Sim", "Não aplicável"] },
                        {   "attribute": ["Incontinência urinária", "enum"],
                            "properties": ["not null"],
                            "values": ["Não", "Sim", "Não aplicável"] },
                        { "attribute": ["Outros sintomas/sinais", "string"] },
                        {   "attribute": ["Pressão arterial sistólica (mm Hg)", "float"],
                            "min": "0" },
                        {   "attribute": ["Pressão arterial diastólica (mm Hg)", "float"],
                            "min": "0" },
                        {   "attribute": ["Frequência cardíaca (cpm)", "float"],
                            "min": "0" },
                        {   "attribute": ["Ritmo cardíaco ECG", "enum+"],
                            "properties": ["not null"],
                            "values": ["Ritmo sinusal", "Fibrilação auricular/flutter",
                                        "Desconhecido"] }
                    ]
                },
                { "entity": [
                        "Evolução Internamento",
                        {   "attribute": ["Estada UCI", "boolean"],
                            "properties": ["not null"],
                            "condition-yes": [
```

Figure C.19: PRECISE Stroke - JSON file specification 19/57.



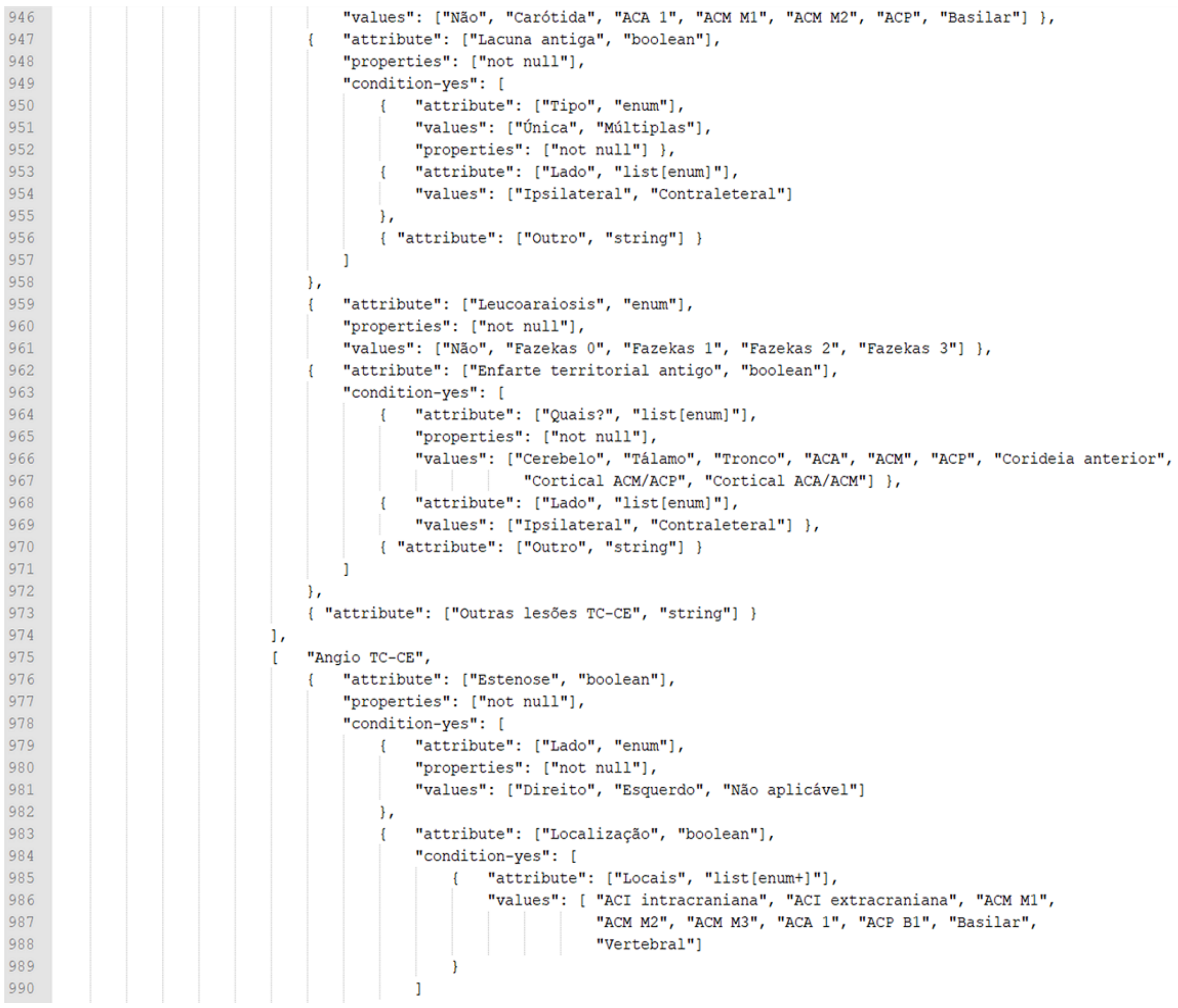
```
                            {   "attribute": ["Nº dias", "integer"],
                                "min": "0" }
                        ]
                    },
                    {   "attribute": ["Quadro neurológico piorou durante o internamento", "enum"],
                        "properties": ["not null"],
                        "values": ["Não", "Agravamento sinais focais", "Aparecimento de novos sinais focais",
                                    "Depressão da vigilidade"]
                    },
                    {   "attribute": ["Hemicraniectomia", "boolean"],
                        "properties": ["not null"] },
                    {   "attribute": ["Outras complicações", "list[enum+]"],
                        "values": ["Infeção", "Enfarte do miocárdio", "TVP/Embolismo pulmonar",
                                    "Crise epilética", "Recorrência AVC", "Caso Social", "Depressão"]
                    },
                    {   "attribute": ["Outros Procedimentos", "list[enum+]"],
                        "values": ["Endarterectomia", "Encerramento apêndice auricular",
                                "Encerramento FOP"]
                    },
                    {   "attribute": ["Fisioterapia", "boolean"],
                        "properties": ["not null"],
                        "condition-yes": [
                            { "attribute": ["Data Início", "date"] },
                            {   "attribute": ["Mins/Sessão", "integer"],
                                "min": "0" },
                            {   "attribute": ["Sessões/Internamento", "integer"],
                                "min": "0" }
                        ]
                    },
                    {   "attribute": ["Fisioterapia da Fala", "boolean"],
                        "properties": ["not null"],
                        "condition-yes": [
                            { "attribute": ["Data Início", "date"] },
                            {   "attribute": ["Mins/Sessão", "integer"],
                                "min": "0" },
                            {   "attribute": ["Sessões/Internamento", "integer"],
                                "min": "0" }
                        ]
                    }
                ]
            }
        ]
    },
    "medical examination": {
        "label": "Exames Médicos",
```

Figure C.20: PRECISE Stroke - JSON file specification 20/57.



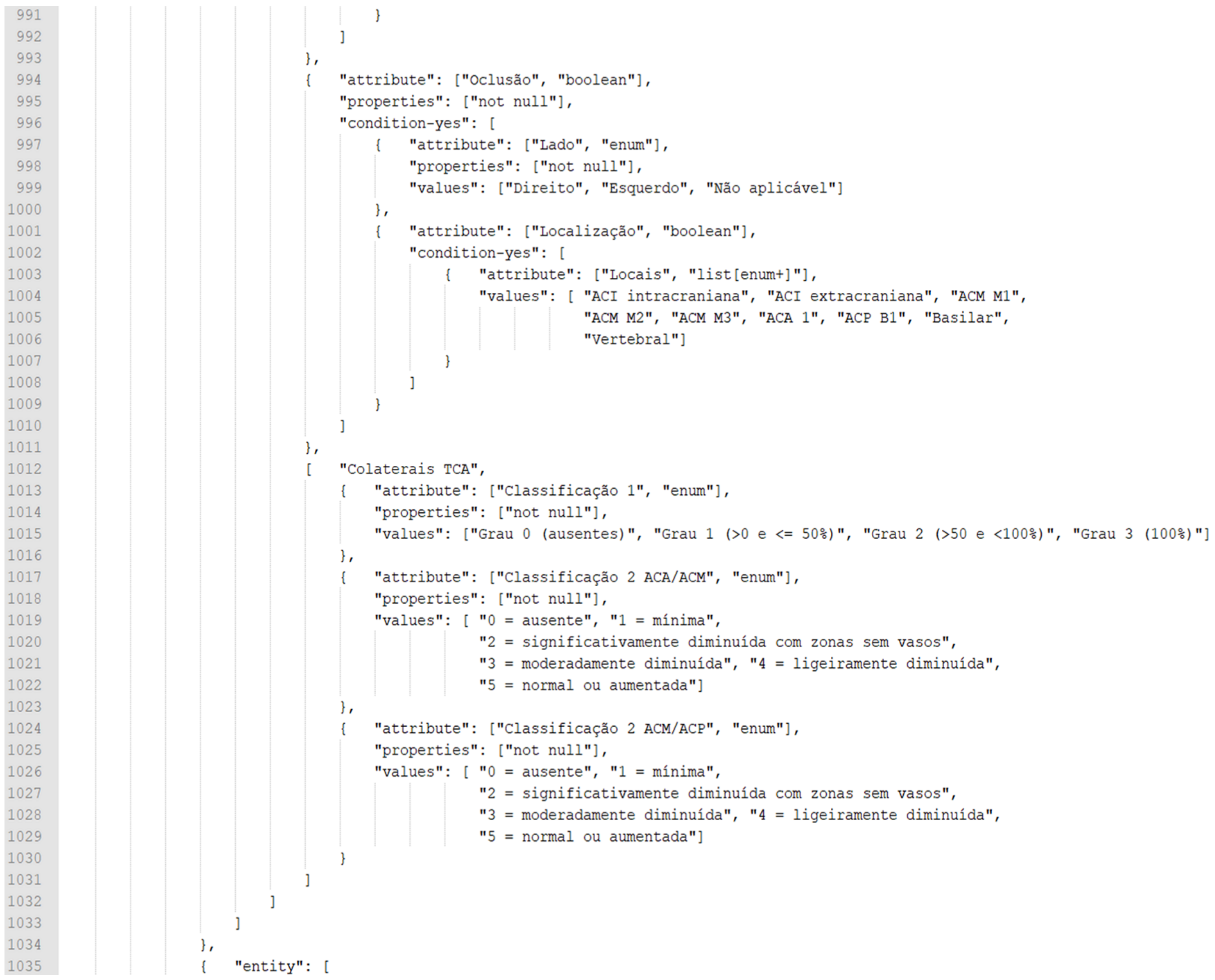
```
"entities": [
    {   "entity": [
            "Biomarcadores Genéticos",
            {   "attribute": ["Cobertura média", "integer"],
                "min": "0" },
            {   "attribute": ["Mapeabilidade 30", "integer"],
                "min": "0",
                "max": "100" },
            {   "attribute": ["Mapeabilidade 100", "integer"],
                "min": "0",
                "max": "100" },
            {   "attribute": ["Similaridade", "integer"],
                "min": "0",
                "max": "100" },
            {   "attribute": ["Número de variantes", "integer"],
                "min": "0" },
            {   "attribute": ["Número de variantes exclusivas", "integer"],
                "min": "0" },
            { "attribute": ["Haplogrupo", "string"] },
            { "attribute": ["Haplogrupo Simples", "string"] },
            {   "attribute": ["Qualidade do haplogrupo", "float"],
                "min": "0",
                "max": "1" }
        ]
    },
    {   "entity": [
            "Imagem Inicial",
            [   "TC-CE",
                {   "attribute": ["ASPECTS", "enum"],
                    "properties": ["not null"],
                    "values": ["10", "9", "8", "7", "6", "5", "4", "3", "2", "1", "0",
                                "Não aplicável"] },
                {   "attribute": ["Outro território com sinal de isquemia", "boolean"],
                    "condition-yes": [
                        {   "attribute": ["Quais?", "list[enum]"],
                            "values": ["ACA", "ACP", "VB"],
                            "properties": ["not null"] },
                        {   "attribute": ["Lado", "list[enum]"],
                            "values": ["Ipsilateral", "Contraleteral"]
                        },
                        { "attribute": ["Outro", "string"] }
                    ]
                },
                {   "attribute": ["Sinal da corda ou dot sign", "enum"],
                    "properties": ["not null"],
```

Figure C.21: PRECISE Stroke - JSON file specification 21/57.



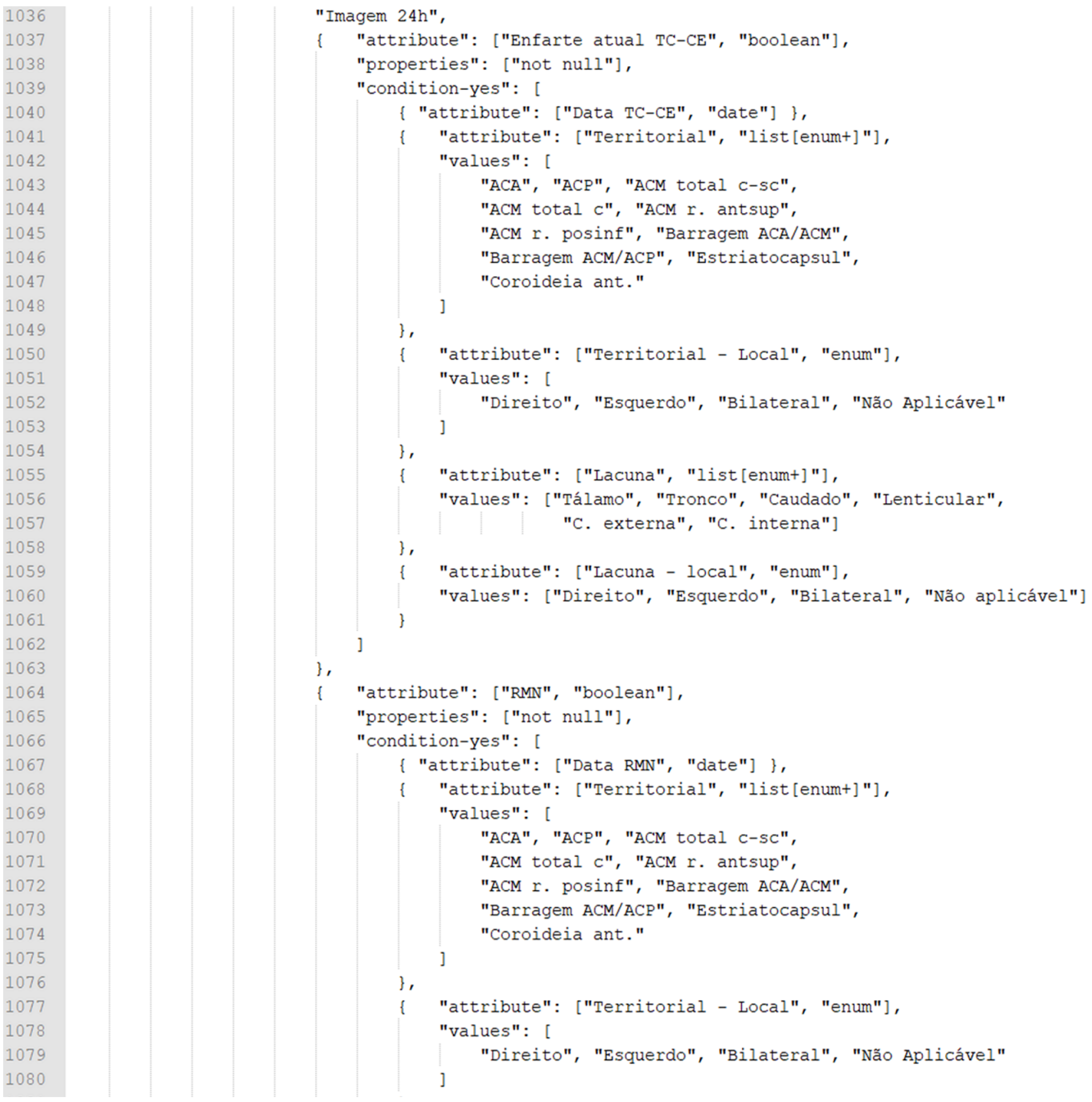
```
        "values": ["Não", "Carótida", "ACA 1", "ACM M1", "ACM M2", "ACP", "Basilar"] },
    {   "attribute": ["Lacuna antiga", "boolean"],
        "properties": ["not null"],
        "condition-yes": [
            {   "attribute": ["Tipo", "enum"],
                "values": ["Única", "Múltiplas"],
                "properties": ["not null"] },
            {   "attribute": ["Lado", "list[enum]"],
                "values": ["Ipsilateral", "Contraleteral"]
            },
            { "attribute": ["Outro", "string"] }
        ]
    },
    {   "attribute": ["Leucoaraiosis", "enum"],
        "properties": ["not null"],
        "values": ["Não", "Fazekas 0", "Fazekas 1", "Fazekas 2", "Fazekas 3"] },
    {   "attribute": ["Enfarte territorial antigo", "boolean"],
        "condition-yes": [
            {   "attribute": ["Quais?", "list[enum]"],
                "properties": ["not null"],
                "values": ["Cerebelo", "Tálamo", "Tronco", "ACA", "ACM", "ACP", "Corideia anterior",
                            "Cortical ACM/ACP", "Cortical ACA/ACM"] },
            {   "attribute": ["Lado", "list[enum]"],
                "values": ["Ipsilateral", "Contraleteral"] },
            { "attribute": ["Outro", "string"] }
        ]
    },
    { "attribute": ["Outras lesões TC-CE", "string"] }
],
[   "Angio TC-CE",
    {   "attribute": ["Estenose", "boolean"],
        "properties": ["not null"],
        "condition-yes": [
            {   "attribute": ["Lado", "enum"],
                "properties": ["not null"],
                "values": ["Direito", "Esquerdo", "Não aplicável"]
            },
            {   "attribute": ["Localização", "boolean"],
                "condition-yes": [
                    {   "attribute": ["Locais", "list[enum+]"],
                        "values": [ "ACI intracraniana", "ACI extracraniana", "ACM M1",
                                    "ACM M2", "ACM M3", "ACA 1", "ACP B1", "Basilar",
                                    "Vertebral"]
                    }
                ]
```

Figure C.22: PRECISE Stroke - JSON file specification 22/57.



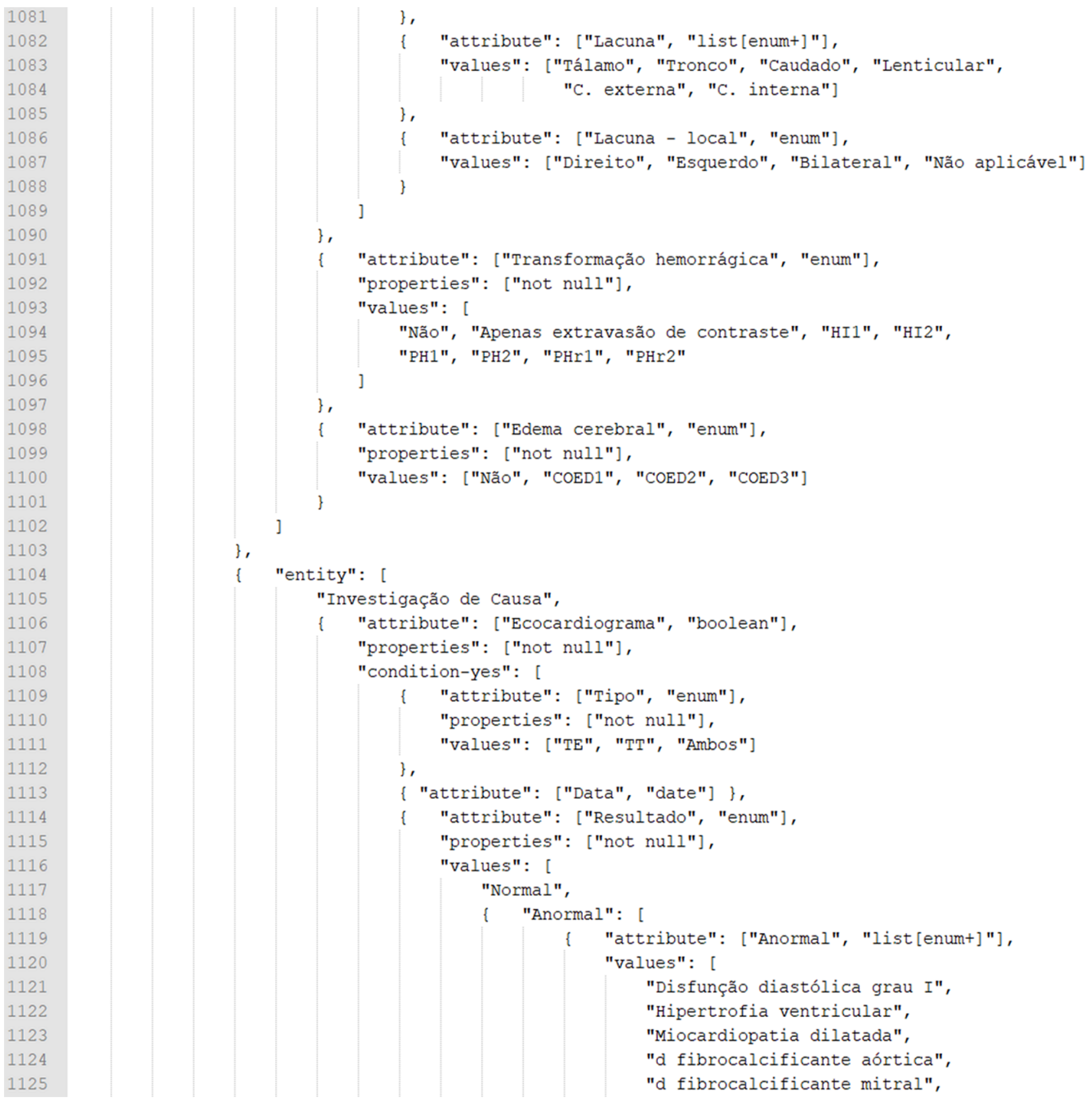
```
                                    }
                                ]
                            },
                            {   "attribute": ["Oclusão", "boolean"],
                                "properties": ["not null"],
                                "condition-yes": [
                                    {   "attribute": ["Lado", "enum"],
                                        "properties": ["not null"],
                                        "values": ["Direito", "Esquerdo", "Não aplicável"]
                                    },
                                    {   "attribute": ["Localização", "boolean"],
                                        "condition-yes": [
                                            {   "attribute": ["Locais", "list[enum+]"],
                                                "values": [ "ACI intracraniana", "ACI extracraniana", "ACM M1",
                                                            "ACM M2", "ACM M3", "ACA 1", "ACP B1", "Basilar",
                                                            "Vertebral"]
                                            }
                                        ]
                                    }
                                ]
                            },
                            [   "Colaterais TCA",
                                {   "attribute": ["Classificação 1", "enum"],
                                    "properties": ["not null"],
                                    "values": ["Grau 0 (ausentes)", "Grau 1 (>0 e <= 50%)", "Grau 2 (>50 e <100%)", "Grau 3 (100%)"]
                                },
                                {   "attribute": ["Classificação 2 ACA/ACM", "enum"],
                                    "properties": ["not null"],
                                    "values": [ "0 = ausente", "1 = mínima",
                                                "2 = significativamente diminuída com zonas sem vasos",
                                                "3 = moderadamente diminuída", "4 = ligeiramente diminuída",
                                                "5 = normal ou aumentada"]
                                },
                                {   "attribute": ["Classificação 2 ACM/ACP", "enum"],
                                    "properties": ["not null"],
                                    "values": [ "0 = ausente", "1 = mínima",
                                                "2 = significativamente diminuída com zonas sem vasos",
                                                "3 = moderadamente diminuída", "4 = ligeiramente diminuída",
                                                "5 = normal ou aumentada"]
                                }
                            ]
                        ]
                    ]
                },
                {   "entity": [
```

Figure C.23: PRECISE Stroke - JSON file specification 23/57.



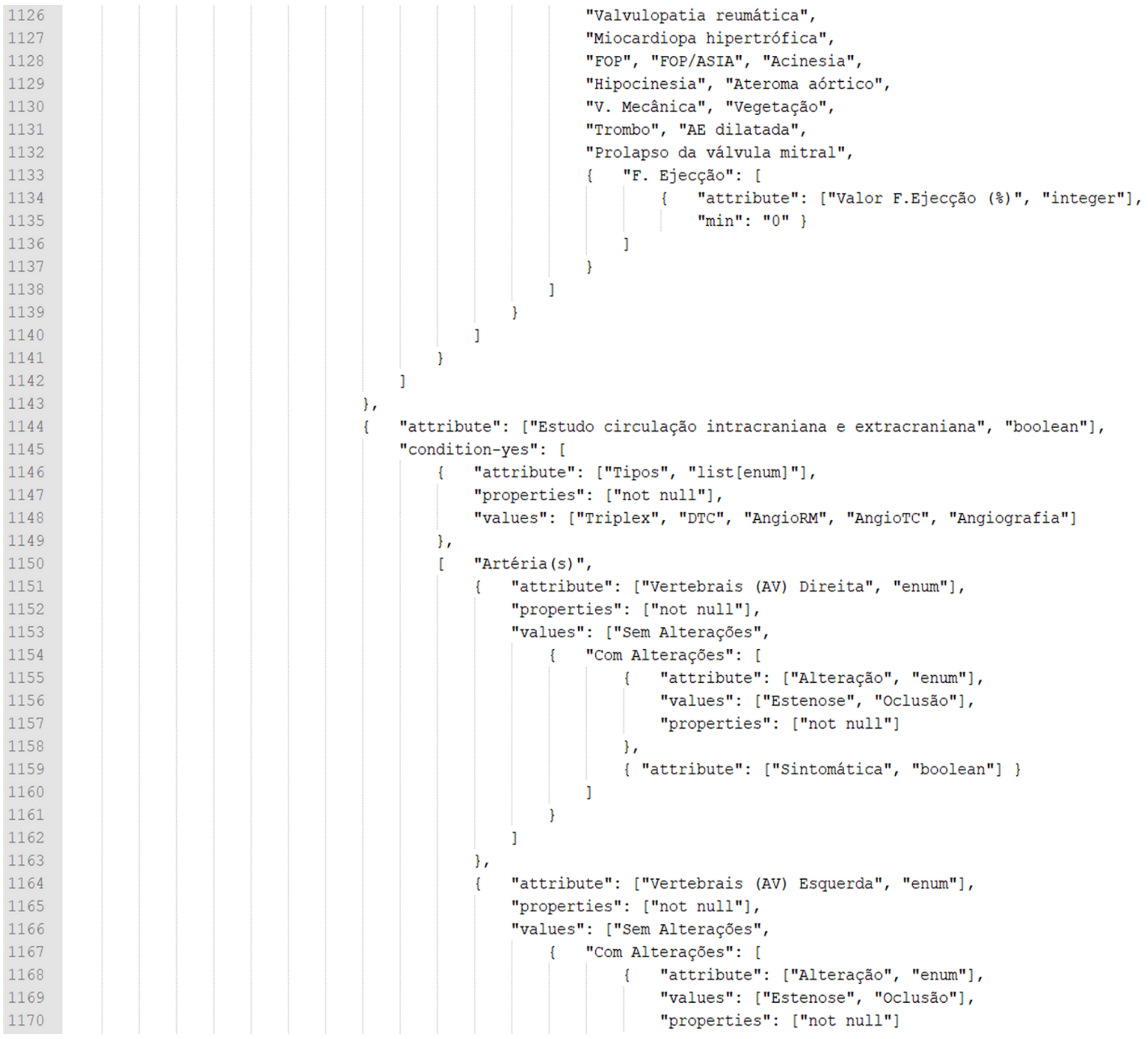
```
                        "Imagem 24h",
                        {   "attribute": ["Enfarte atual TC-CE", "boolean"],
                            "properties": ["not null"],
                            "condition-yes": [
                                { "attribute": ["Data TC-CE", "date"] },
                                {   "attribute": ["Territorial", "list[enum+]"],
                                    "values": [
                                        "ACA", "ACP", "ACM total c-sc",
                                        "ACM total c", "ACM r. antsup",
                                        "ACM r. posinf", "Barragem ACA/ACM",
                                        "Barragem ACM/ACP", "Estriatocapsul",
                                        "Coroideia ant."
                                    ]
                                },
                                {   "attribute": ["Territorial - Local", "enum"],
                                    "values": [
                                        "Direito", "Esquerdo", "Bilateral", "Não Aplicável"
                                    ]
                                },
                                {   "attribute": ["Lacuna", "list[enum+]"],
                                    "values": ["Tálamo", "Tronco", "Caudado", "Lenticular",
                                                "C. externa", "C. interna"]
                                },
                                {   "attribute": ["Lacuna - local", "enum"],
                                    "values": ["Direito", "Esquerdo", "Bilateral", "Não aplicável"]
                                }
                            ]
                        },
                        {   "attribute": ["RMN", "boolean"],
                            "properties": ["not null"],
                            "condition-yes": [
                                { "attribute": ["Data RMN", "date"] },
                                {   "attribute": ["Territorial", "list[enum+]"],
                                    "values": [
                                        "ACA", "ACP", "ACM total c-sc",
                                        "ACM total c", "ACM r. antsup",
                                        "ACM r. posinf", "Barragem ACA/ACM",
                                        "Barragem ACM/ACP", "Estriatocapsul",
                                        "Coroideia ant."
                                    ]
                                },
                                {   "attribute": ["Territorial - Local", "enum"],
                                    "values": [
                                        "Direito", "Esquerdo", "Bilateral", "Não Aplicável"
                                    ]
```

Figure C.24: PRECISE Stroke - JSON file specification 24/57.



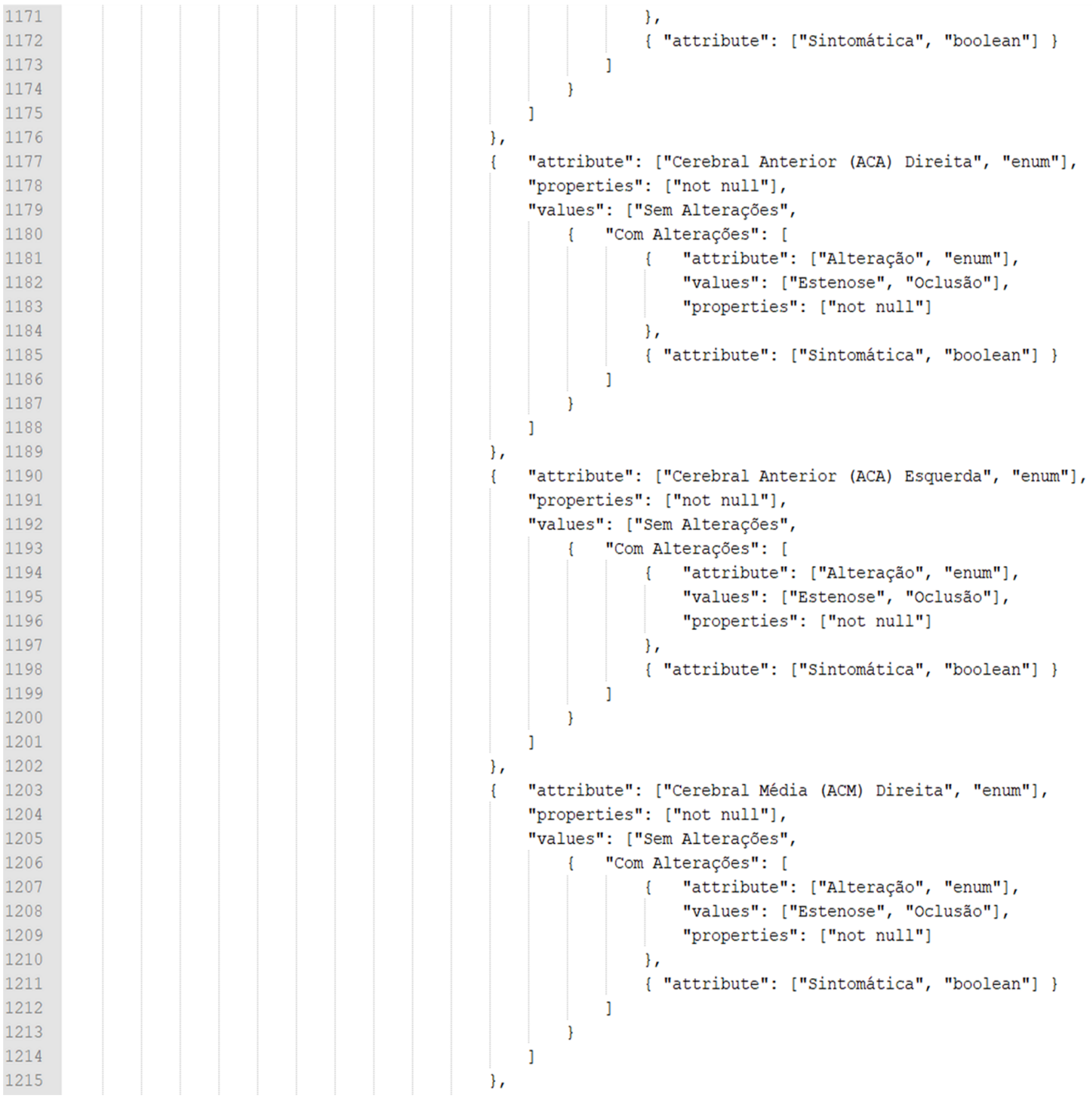
```
                },
                {   "attribute": ["Lacuna", "list[enum+]"],
                    "values": ["Tálamo", "Tronco", "Caudado", "Lenticular",
                                "C. externa", "C. interna"]
                },
                {   "attribute": ["Lacuna - local", "enum"],
                    "values": ["Direito", "Esquerdo", "Bilateral", "Não aplicável"]
                }
            ]
        },
        {   "attribute": ["Transformação hemorrágica", "enum"],
            "properties": ["not null"],
            "values": [
                "Não", "Apenas extravasão de contraste", "HI1", "HI2",
                "PH1", "PH2", "PHr1", "PHr2"
            ]
        },
        {   "attribute": ["Edema cerebral", "enum"],
            "properties": ["not null"],
            "values": ["Não", "COED1", "COED2", "COED3"]
        }
    ]
},
{   "entity": [
        "Investigação de Causa",
        {   "attribute": ["Ecocardiograma", "boolean"],
            "properties": ["not null"],
            "condition-yes": [
                {   "attribute": ["Tipo", "enum"],
                    "properties": ["not null"],
                    "values": ["TE", "TT", "Ambos"]
                },
                { "attribute": ["Data", "date"] },
                {   "attribute": ["Resultado", "enum"],
                    "properties": ["not null"],
                    "values": [
                        "Normal",
                        {   "Anormal": [
                                {   "attribute": ["Anormal", "list[enum+]"],
                                    "values": [
                                        "Disfunção diastólica grau I",
                                        "Hipertrofia ventricular",
                                        "Miocardiopatia dilatada",
                                        "d fibrocalcificante aórtica",
                                        "d fibrocalcificante mitral",
```

Figure C.25: PRECISE Stroke - JSON file specification 25/57.



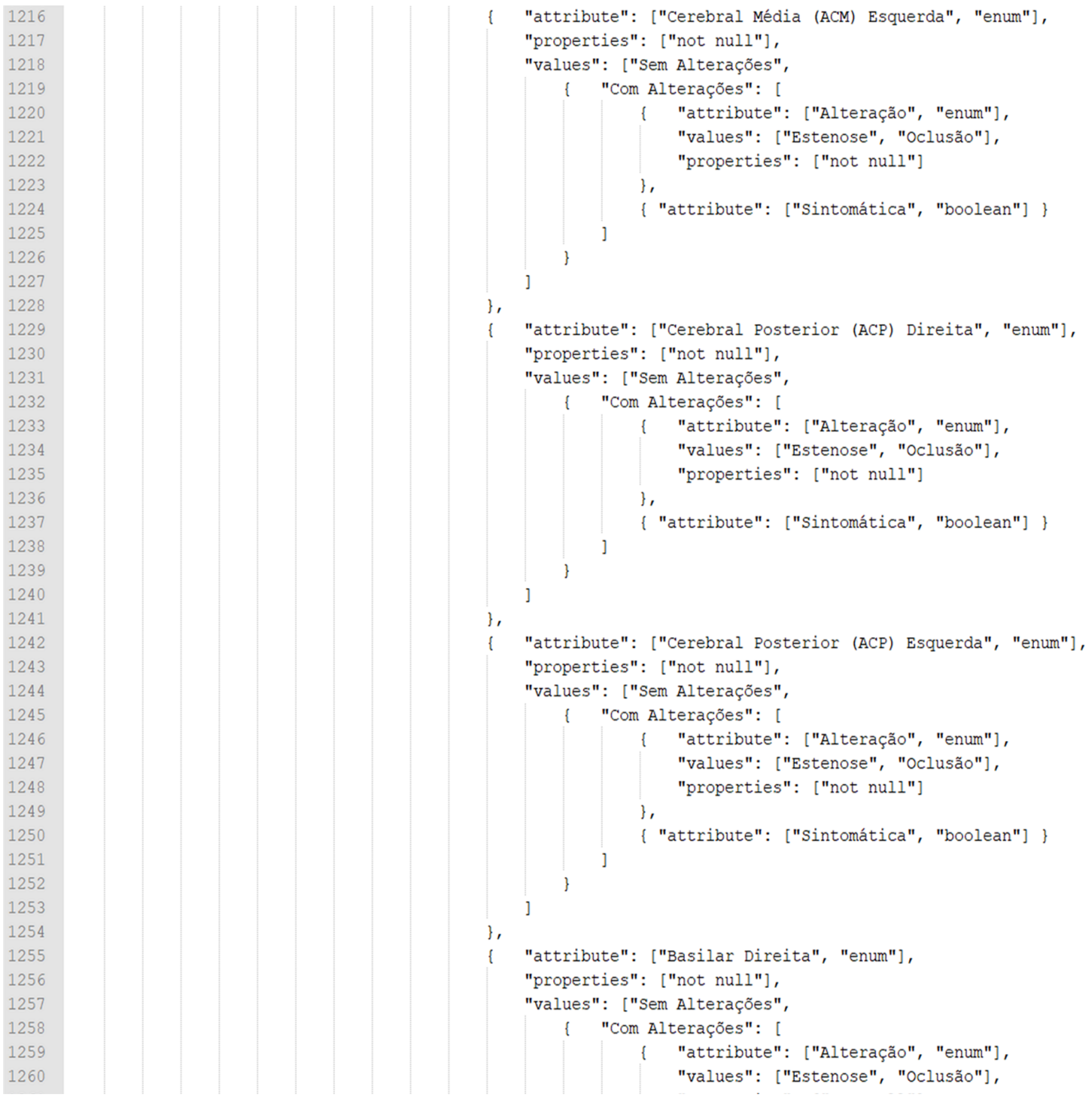
```
                                                        "Valvulopatia reumática",
                                                        "Miocardiopa hipertrófica",
                                                        "FOP", "FOP/ASIA", "Acinesia",
                                                        "Hipocinesia", "Ateroma aórtico",
                                                        "V. Mecânica", "Vegetação",
                                                        "Trombo", "AE dilatada",
                                                        "Prolapso da válvula mitral",
                                                        {   "F. Ejecção": [
                                                                {   "attribute": ["Valor F.Ejecção (%)", "integer"],
                                                                    "min": "0" }
                                                            ]
                                                        }
                                                    ]
                                                }
                                            ]
                                        }
                                    ]
                                },
                                {   "attribute": ["Estudo circulação intracraniana e extracraniana", "boolean"],
                                    "condition-yes": [
                                        {   "attribute": ["Tipos", "list[enum]"],
                                            "properties": ["not null"],
                                            "values": ["Triplex", "DTC", "AngioRM", "AngioTC", "Angiografia"]
                                        },
                                        [   "Artéria(s)",
                                            {   "attribute": ["Vertebrais (AV) Direita", "enum"],
                                                "properties": ["not null"],
                                                "values": ["Sem Alterações",
                                                    {   "Com Alterações": [
                                                            {   "attribute": ["Alteração", "enum"],
                                                                "values": ["Estenose", "Oclusão"],
                                                                "properties": ["not null"]
                                                            },
                                                            { "attribute": ["Sintomática", "boolean"] }
                                                        ]
                                                    }
                                                ]
                                            },
                                            {   "attribute": ["Vertebrais (AV) Esquerda", "enum"],
                                                "properties": ["not null"],
                                                "values": ["Sem Alterações",
                                                    {   "Com Alterações": [
                                                            {   "attribute": ["Alteração", "enum"],
                                                                "values": ["Estenose", "Oclusão"],
                                                                "properties": ["not null"]
```

Figure C.26: PRECISE Stroke - JSON file specification 26/57.



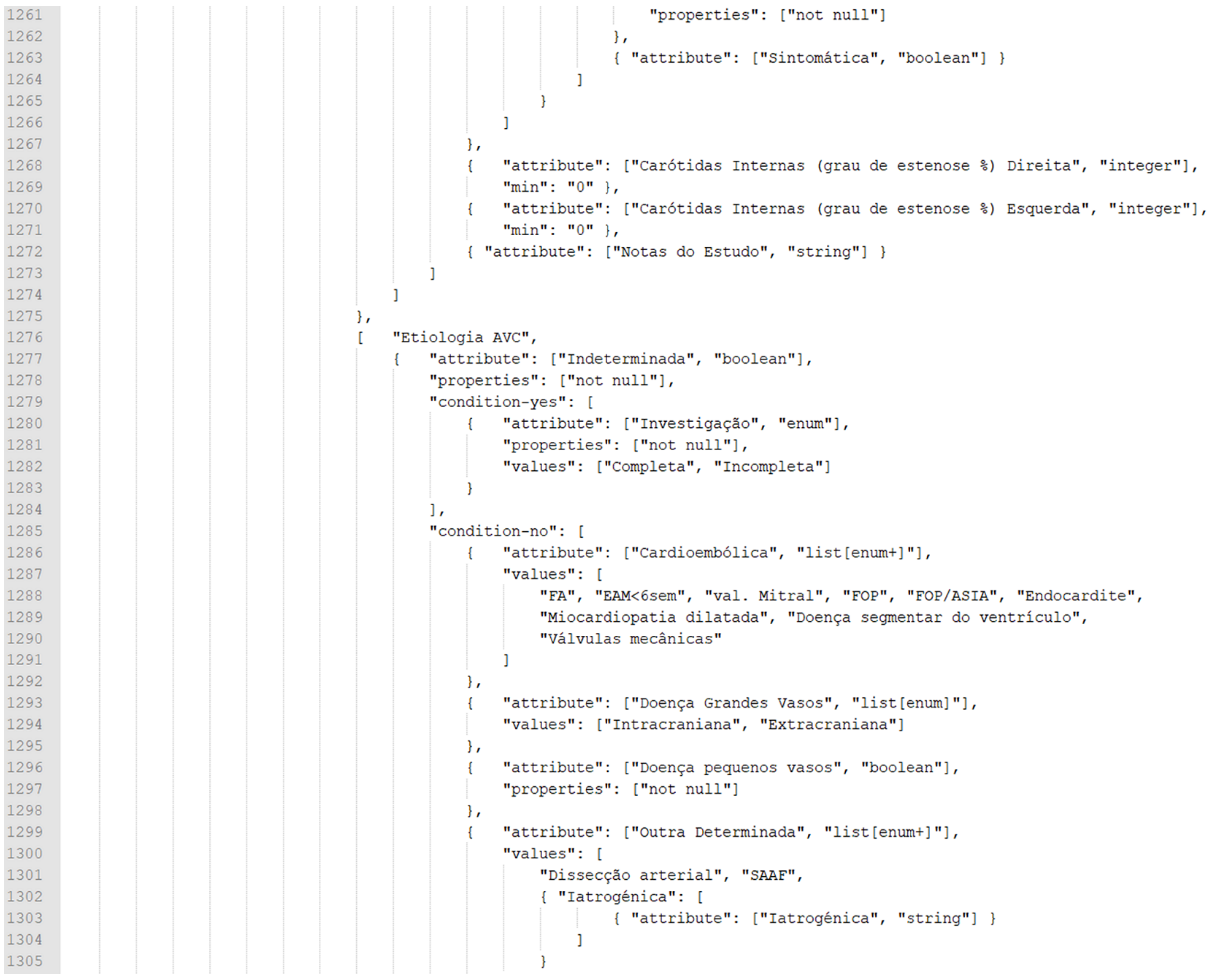
```
                                                            },
                                                            { "attribute": ["Sintomática", "boolean"] }
                                                        ]
                                                    }
                                                ]
                                            },
                                            {   "attribute": ["Cerebral Anterior (ACA) Direita", "enum"],
                                                "properties": ["not null"],
                                                "values": ["Sem Alterações",
                                                    {   "Com Alterações": [
                                                            {   "attribute": ["Alteração", "enum"],
                                                                "values": ["Estenose", "Oclusão"],
                                                                "properties": ["not null"]
                                                            },
                                                            { "attribute": ["Sintomática", "boolean"] }
                                                        ]
                                                    }
                                                ]
                                            },
                                            {   "attribute": ["Cerebral Anterior (ACA) Esquerda", "enum"],
                                                "properties": ["not null"],
                                                "values": ["Sem Alterações",
                                                    {   "Com Alterações": [
                                                            {   "attribute": ["Alteração", "enum"],
                                                                "values": ["Estenose", "Oclusão"],
                                                                "properties": ["not null"]
                                                            },
                                                            { "attribute": ["Sintomática", "boolean"] }
                                                        ]
                                                    }
                                                ]
                                            },
                                            {   "attribute": ["Cerebral Média (ACM) Direita", "enum"],
                                                "properties": ["not null"],
                                                "values": ["Sem Alterações",
                                                    {   "Com Alterações": [
                                                            {   "attribute": ["Alteração", "enum"],
                                                                "values": ["Estenose", "Oclusão"],
                                                                "properties": ["not null"]
                                                            },
                                                            { "attribute": ["Sintomática", "boolean"] }
                                                        ]
                                                    }
                                                ]
                                            },
```

Figure C.27: PRECISE Stroke - JSON file specification 27/57.



```
                                            {   "attribute": ["Cerebral Média (ACM) Esquerda", "enum"],
                                                "properties": ["not null"],
                                                "values": ["Sem Alterações",
                                                    {   "Com Alterações": [
                                                            {   "attribute": ["Alteração", "enum"],
                                                                "values": ["Estenose", "Oclusão"],
                                                                "properties": ["not null"]
                                                            },
                                                            { "attribute": ["Sintomática", "boolean"] }
                                                        ]
                                                    }
                                                ]
                                            },
                                            {   "attribute": ["Cerebral Posterior (ACP) Direita", "enum"],
                                                "properties": ["not null"],
                                                "values": ["Sem Alterações",
                                                    {   "Com Alterações": [
                                                            {   "attribute": ["Alteração", "enum"],
                                                                "values": ["Estenose", "Oclusão"],
                                                                "properties": ["not null"]
                                                            },
                                                            { "attribute": ["Sintomática", "boolean"] }
                                                        ]
                                                    }
                                                ]
                                            },
                                            {   "attribute": ["Cerebral Posterior (ACP) Esquerda", "enum"],
                                                "properties": ["not null"],
                                                "values": ["Sem Alterações",
                                                    {   "Com Alterações": [
                                                            {   "attribute": ["Alteração", "enum"],
                                                                "values": ["Estenose", "Oclusão"],
                                                                "properties": ["not null"]
                                                            },
                                                            { "attribute": ["Sintomática", "boolean"] }
                                                        ]
                                                    }
                                                ]
                                            },
                                            {   "attribute": ["Basilar Direita", "enum"],
                                                "properties": ["not null"],
                                                "values": ["Sem Alterações",
                                                    {   "Com Alterações": [
                                                            {   "attribute": ["Alteração", "enum"],
                                                                "values": ["Estenose", "Oclusão"],
```

Figure C.28: PRECISE Stroke - JSON file specification 28/57.



```
                                                                "properties": ["not null"]
                                                            },
                                                            { "attribute": ["Sintomática", "boolean"] }
                                                        ]
                                                    }
                                                ]
                                            },
                                            {   "attribute": ["Carótidas Internas (grau de estenose %) Direita", "integer"],
                                                "min": "0" },
                                            {   "attribute": ["Carótidas Internas (grau de estenose %) Esquerda", "integer"],
                                                "min": "0" },
                                            { "attribute": ["Notas do Estudo", "string"] }
                                        ]
                                    ]
                                },
                                [   "Etiologia AVC",
                                    {   "attribute": ["Indeterminada", "boolean"],
                                        "properties": ["not null"],
                                        "condition-yes": [
                                            {   "attribute": ["Investigação", "enum"],
                                                "properties": ["not null"],
                                                "values": ["Completa", "Incompleta"]
                                            }
                                        ],
                                        "condition-no": [
                                            {   "attribute": ["Cardioembólica", "list[enum+]"],
                                                "values": [
                                                    "FA", "EAM<6sem", "val. Mitral", "FOP", "FOP/ASIA", "Endocardite",
                                                    "Miocardiopatia dilatada", "Doença segmentar do ventrículo",
                                                    "Válvulas mecânicas"
                                                ]
                                            },
                                            {   "attribute": ["Doença Grandes Vasos", "list[enum]"],
                                                "values": ["Intracraniana", "Extracraniana"]
                                            },
                                            {   "attribute": ["Doença pequenos vasos", "boolean"],
                                                "properties": ["not null"]
                                            },
                                            {   "attribute": ["Outra Determinada", "list[enum+]"],
                                                "values": [
                                                    "Dissecção arterial", "SAAF",
                                                    { "Iatrogénica": [
                                                            { "attribute": ["Iatrogénica", "string"] }
                                                        ]
                                                    }
```

Figure C.29: PRECISE Stroke - JSON file specification 29/57.

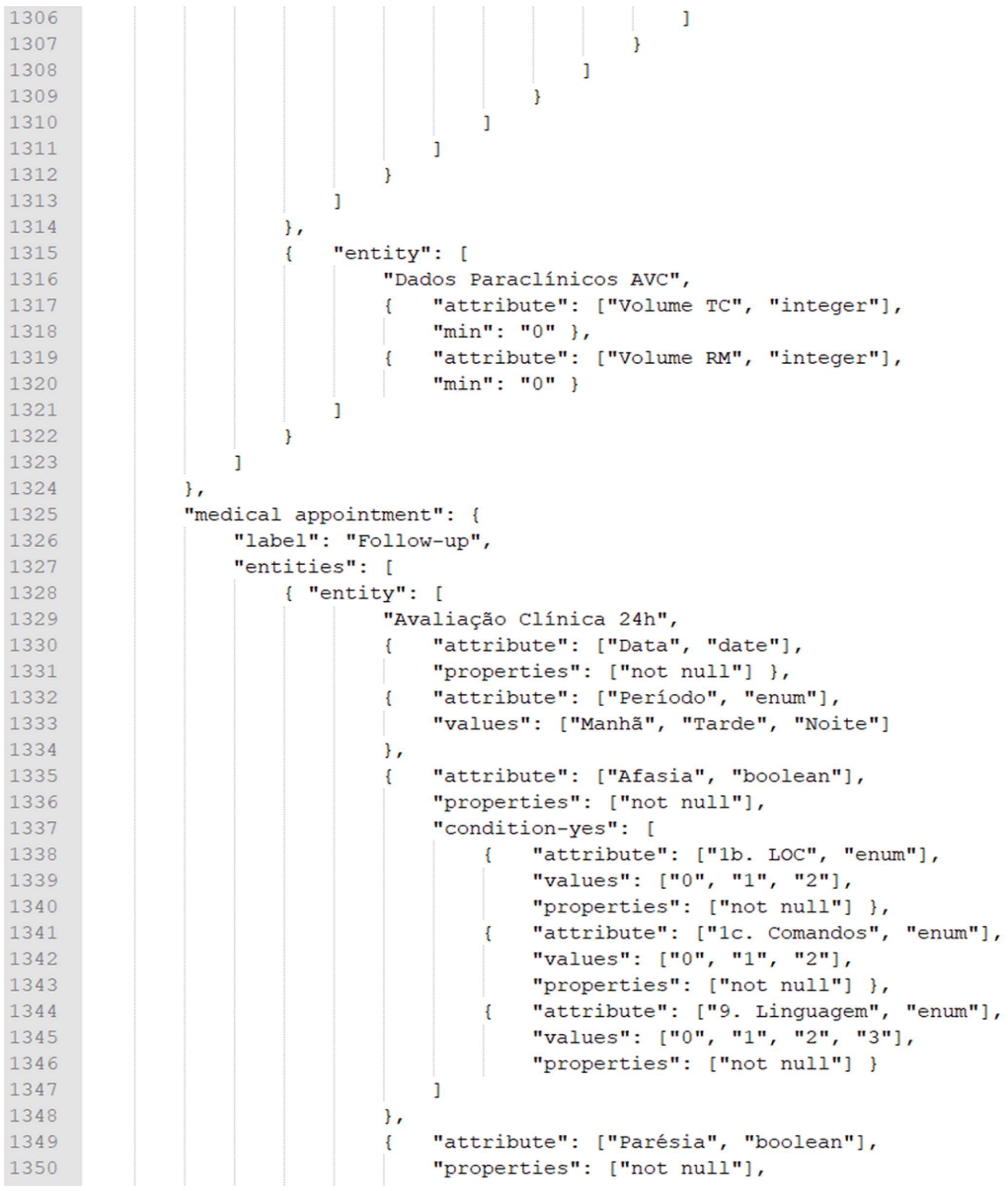
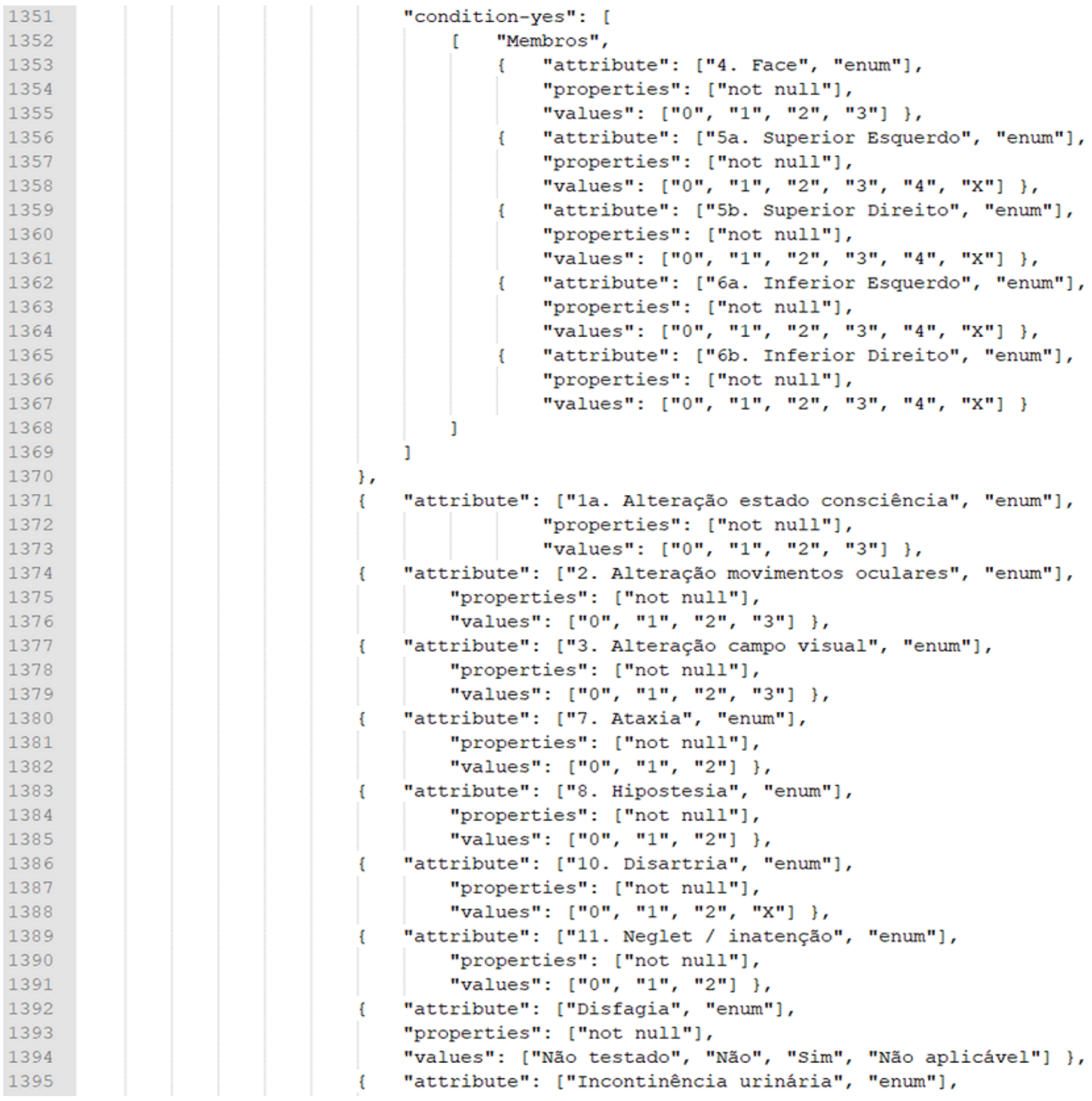
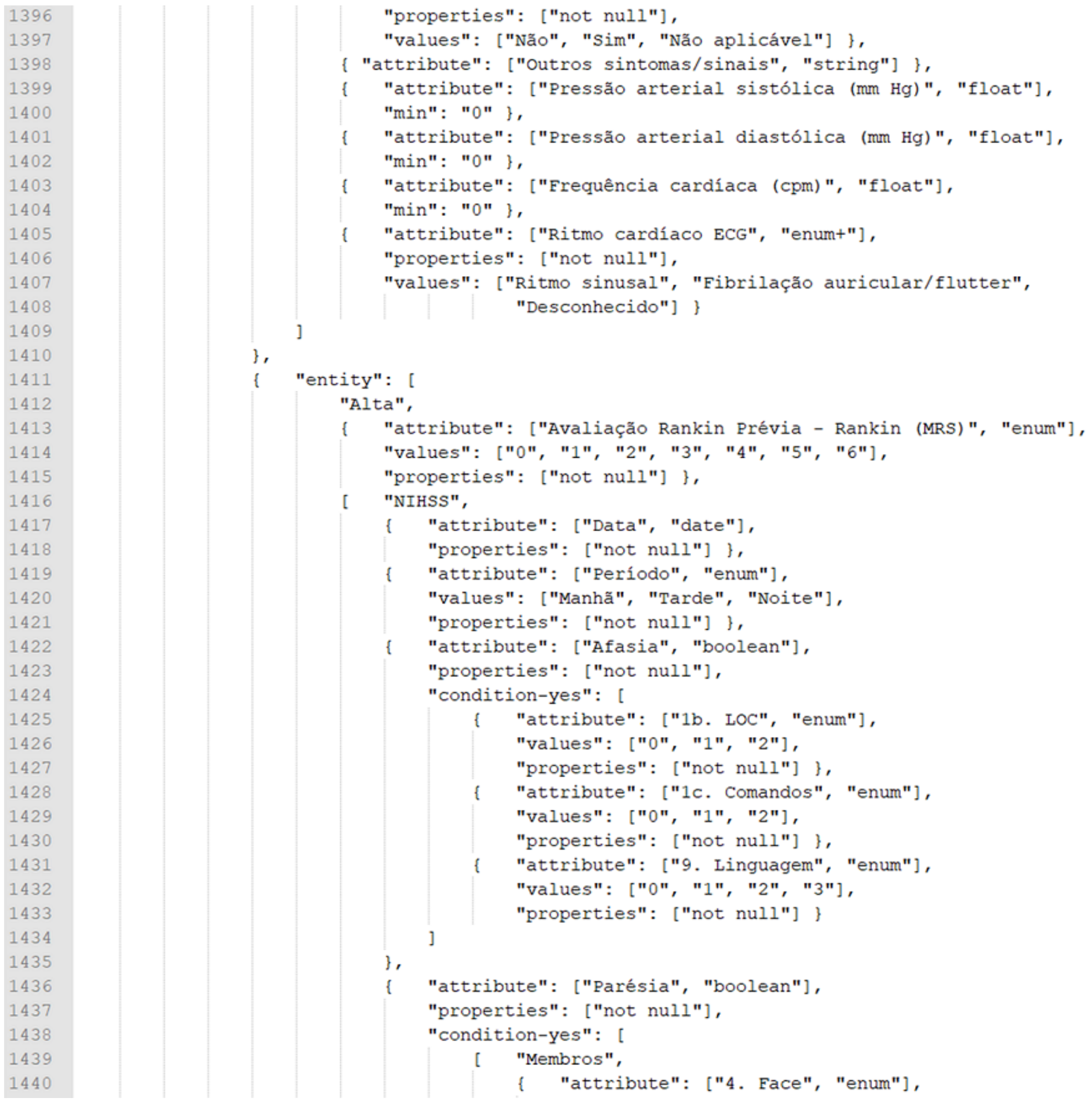

```
                                            ]
                                        }
                                    ]
                                }
                            ]
                        ]
                    }
                ]
            },
            {   "entity": [
                    "Dados Paraclínicos AVC",
                    {   "attribute": ["Volume TC", "integer"],
                        "min": "0" },
                    {   "attribute": ["Volume RM", "integer"],
                        "min": "0" }
                ]
            }
        ]
    },
    "medical appointment": {
        "label": "Follow-up",
        "entities": [
            { "entity": [
                    "Avaliação Clínica 24h",
                    {   "attribute": ["Data", "date"],
                        "properties": ["not null"] },
                    {   "attribute": ["Período", "enum"],
                        "values": ["Manhã", "Tarde", "Noite"]
                    },
                    {   "attribute": ["Afasia", "boolean"],
                        "properties": ["not null"],
                        "condition-yes": [
                            {   "attribute": ["1b. LOC", "enum"],
                                "values": ["0", "1", "2"],
                                "properties": ["not null"] },
                            {   "attribute": ["1c. Comandos", "enum"],
                                "values": ["0", "1", "2"],
                                "properties": ["not null"] },
                            {   "attribute": ["9. Linguagem", "enum"],
                                "values": ["0", "1", "2", "3"],
                                "properties": ["not null"] }
                        ]
                    },
                    {   "attribute": ["Parésia", "boolean"],
                        "properties": ["not null"],
```

Figure C.30: PRECISE Stroke - JSON file specification 30/57.



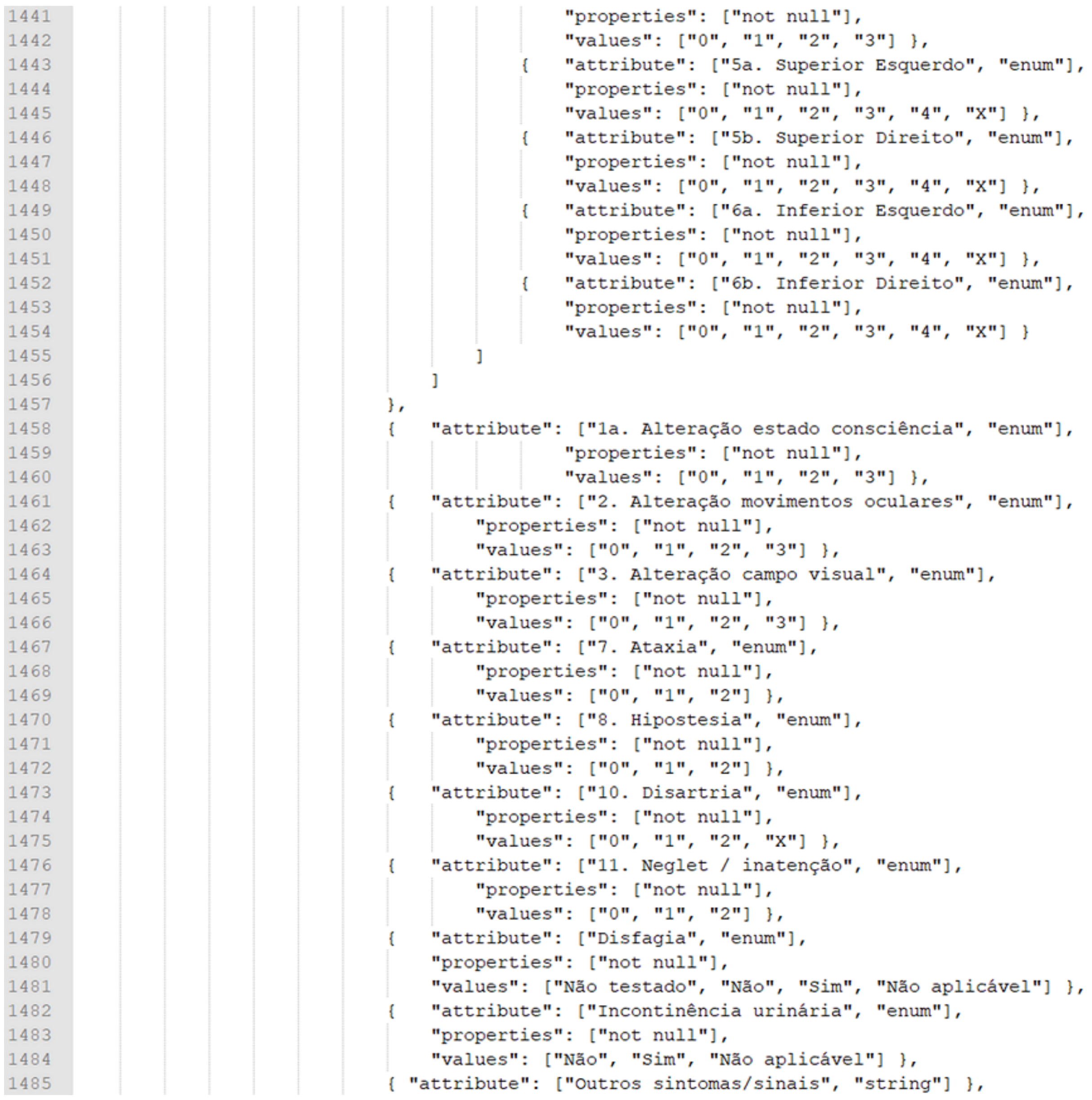
```
                    "condition-yes": [
                        [   "Membros",
                            {   "attribute": ["4. Face", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2", "3"] },
                            {   "attribute": ["5a. Superior Esquerdo", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2", "3", "4", "X"] },
                            {   "attribute": ["5b. Superior Direito", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2", "3", "4", "X"] },
                            {   "attribute": ["6a. Inferior Esquerdo", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2", "3", "4", "X"] },
                            {   "attribute": ["6b. Inferior Direito", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2", "3", "4", "X"] }
                        ]
                    ]
                },
                {   "attribute": ["1a. Alteração estado consciência", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2", "3"] },
                {   "attribute": ["2. Alteração movimentos oculares", "enum"],
                        "properties": ["not null"],
                        "values": ["0", "1", "2", "3"] },
                {   "attribute": ["3. Alteração campo visual", "enum"],
                        "properties": ["not null"],
                        "values": ["0", "1", "2", "3"] },
                {   "attribute": ["7. Ataxia", "enum"],
                        "properties": ["not null"],
                        "values": ["0", "1", "2"] },
                {   "attribute": ["8. Hipostesia", "enum"],
                        "properties": ["not null"],
                        "values": ["0", "1", "2"] },
                {   "attribute": ["10. Disartria", "enum"],
                        "properties": ["not null"],
                        "values": ["0", "1", "2", "X"] },
                {   "attribute": ["11. Neglet / inatenção", "enum"],
                        "properties": ["not null"],
                        "values": ["0", "1", "2"] },
                {   "attribute": ["Disfagia", "enum"],
                    "properties": ["not null"],
                    "values": ["Não testado", "Não", "Sim", "Não aplicável"] },
                {   "attribute": ["Incontinência urinária", "enum"],
```

Figure C.31: PRECISE Stroke - JSON file specification 31/57.



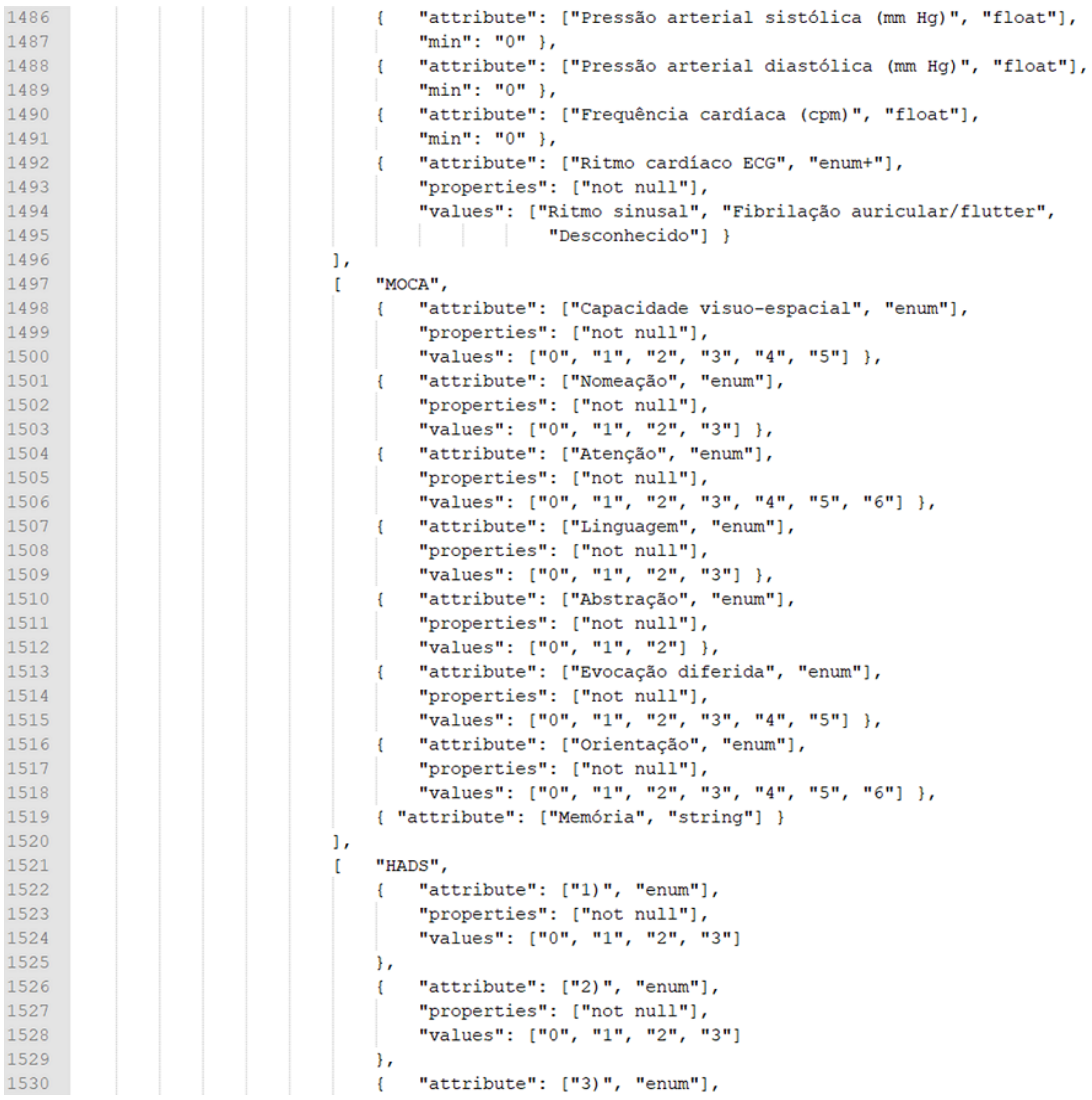
```
                            "properties": ["not null"],
                            "values": ["Não", "Sim", "Não aplicável"] },
                        { "attribute": ["Outros sintomas/sinais", "string"] },
                        {   "attribute": ["Pressão arterial sistólica (mm Hg)", "float"],
                            "min": "0" },
                        {   "attribute": ["Pressão arterial diastólica (mm Hg)", "float"],
                            "min": "0" },
                        {   "attribute": ["Frequência cardíaca (cpm)", "float"],
                            "min": "0" },
                        {   "attribute": ["Ritmo cardíaco ECG", "enum+"],
                            "properties": ["not null"],
                            "values": ["Ritmo sinusal", "Fibrilação auricular/flutter",
                                        "Desconhecido"] }
                    ]
                },
                {   "entity": [
                        "Alta",
                        {   "attribute": ["Avaliação Rankin Prévia - Rankin (MRS)", "enum"],
                            "values": ["0", "1", "2", "3", "4", "5", "6"],
                            "properties": ["not null"] },
                        [   "NIHSS",
                            {   "attribute": ["Data", "date"],
                                "properties": ["not null"] },
                            {   "attribute": ["Período", "enum"],
                                "values": ["Manhã", "Tarde", "Noite"],
                                "properties": ["not null"] },
                            {   "attribute": ["Afasia", "boolean"],
                                "properties": ["not null"],
                                "condition-yes": [
                                    {   "attribute": ["1b. LOC", "enum"],
                                        "values": ["0", "1", "2"],
                                        "properties": ["not null"] },
                                    {   "attribute": ["1c. Comandos", "enum"],
                                        "values": ["0", "1", "2"],
                                        "properties": ["not null"] },
                                    {   "attribute": ["9. Linguagem", "enum"],
                                        "values": ["0", "1", "2", "3"],
                                        "properties": ["not null"] }
                                ]
                            },
                            {   "attribute": ["Parésia", "boolean"],
                                "properties": ["not null"],
                                "condition-yes": [
                                    [   "Membros",
                                        {   "attribute": ["4. Face", "enum"],
```

Figure C.32: PRECISE Stroke - JSON file specification 32/57.



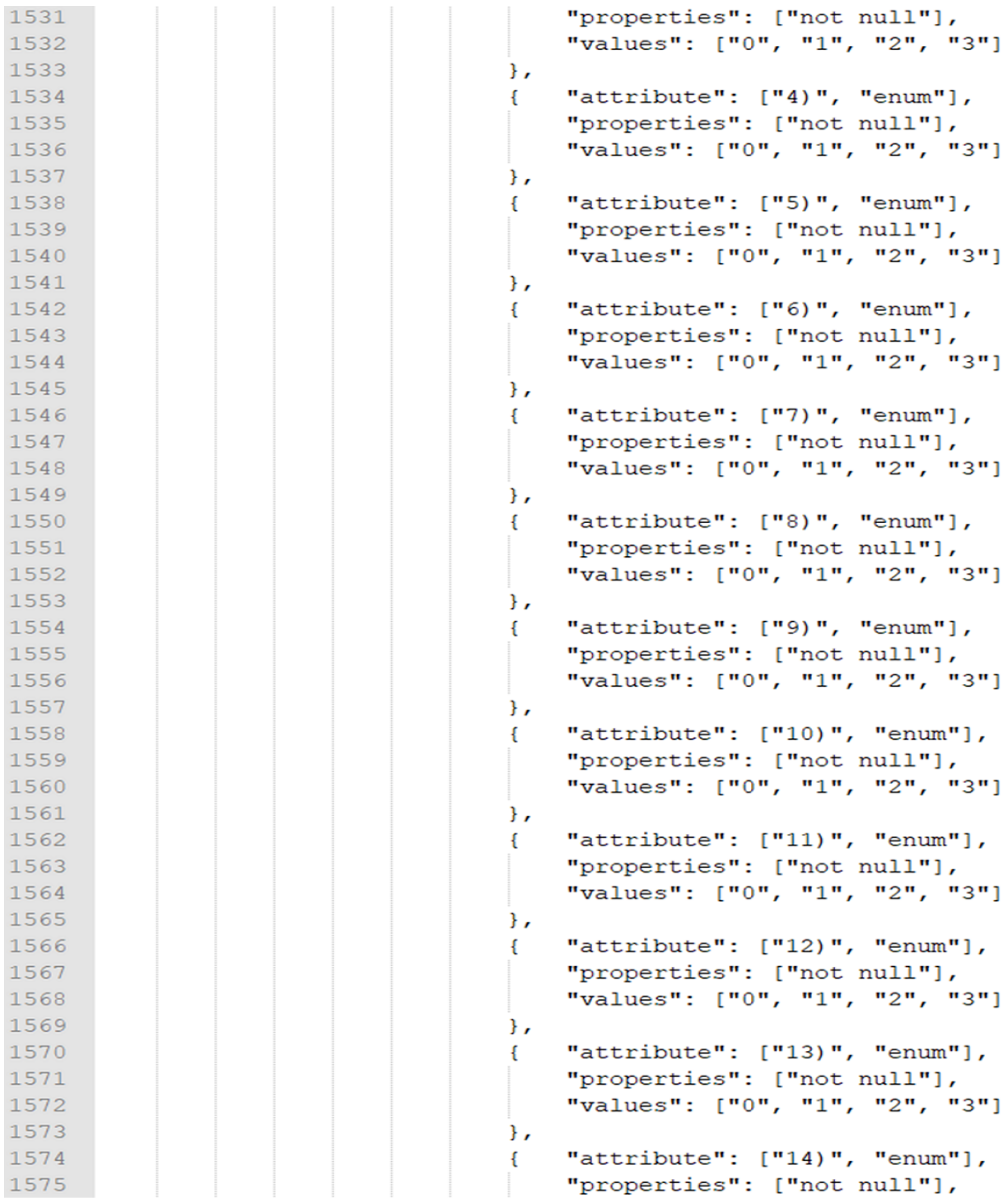
```
                                        "properties": ["not null"],
                                        "values": ["0", "1", "2", "3"] },
                                    {   "attribute": ["5a. Superior Esquerdo", "enum"],
                                        "properties": ["not null"],
                                        "values": ["0", "1", "2", "3", "4", "X"] },
                                    {   "attribute": ["5b. Superior Direito", "enum"],
                                        "properties": ["not null"],
                                        "values": ["0", "1", "2", "3", "4", "X"] },
                                    {   "attribute": ["6a. Inferior Esquerdo", "enum"],
                                        "properties": ["not null"],
                                        "values": ["0", "1", "2", "3", "4", "X"] },
                                    {   "attribute": ["6b. Inferior Direito", "enum"],
                                        "properties": ["not null"],
                                        "values": ["0", "1", "2", "3", "4", "X"] }
                                ]
                            ]
                        },
                        {   "attribute": ["1a. Alteração estado consciência", "enum"],
                                        "properties": ["not null"],
                                        "values": ["0", "1", "2", "3"] },
                        {   "attribute": ["2. Alteração movimentos oculares", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2", "3"] },
                        {   "attribute": ["3. Alteração campo visual", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2", "3"] },
                        {   "attribute": ["7. Ataxia", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2"] },
                        {   "attribute": ["8. Hipostesia", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2"] },
                        {   "attribute": ["10. Disartria", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2", "X"] },
                        {   "attribute": ["11. Neglet / inatenção", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2"] },
                        {   "attribute": ["Disfagia", "enum"],
                            "properties": ["not null"],
                            "values": ["Não testado", "Não", "Sim", "Não aplicável"] },
                        {   "attribute": ["Incontinência urinária", "enum"],
                            "properties": ["not null"],
                            "values": ["Não", "Sim", "Não aplicável"] },
                        { "attribute": ["Outros sintomas/sinais", "string"] },
```

Figure C.33: PRECISE Stroke - JSON file specification 33/57.



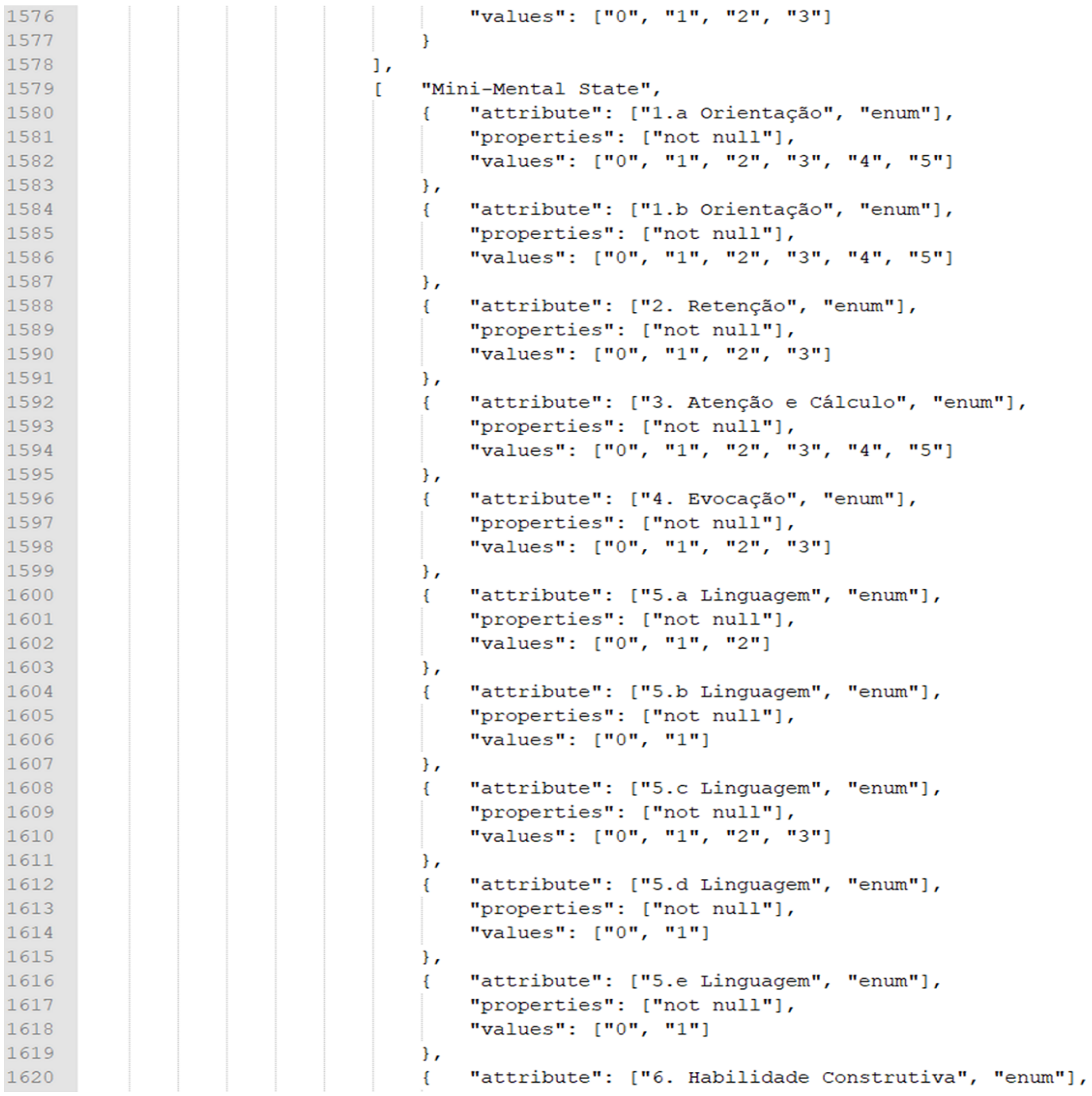
```
                            {   "attribute": ["Pressão arterial sistólica (mm Hg)", "float"],
                                "min": "0" },
                            {   "attribute": ["Pressão arterial diastólica (mm Hg)", "float"],
                                "min": "0" },
                            {   "attribute": ["Frequência cardíaca (cpm)", "float"],
                                "min": "0" },
                            {   "attribute": ["Ritmo cardíaco ECG", "enum+"],
                                "properties": ["not null"],
                                "values": ["Ritmo sinusal", "Fibrilação auricular/flutter",
                                            "Desconhecido"] }
                        ],
                        [   "MOCA",
                            {   "attribute": ["Capacidade visuo-espacial", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2", "3", "4", "5"] },
                            {   "attribute": ["Nomeação", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2", "3"] },
                            {   "attribute": ["Atenção", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2", "3", "4", "5", "6"] },
                            {   "attribute": ["Linguagem", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2", "3"] },
                            {   "attribute": ["Abstração", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2"] },
                            {   "attribute": ["Evocação diferida", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2", "3", "4", "5"] },
                            {   "attribute": ["Orientação", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2", "3", "4", "5", "6"] },
                            { "attribute": ["Memória", "string"] }
                        ],
                        [   "HADS",
                            {   "attribute": ["1)", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2", "3"]
                            },
                            {   "attribute": ["2)", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2", "3"]
                            },
                            {   "attribute": ["3)", "enum"],
```

Figure C.34: PRECISE Stroke - JSON file specification 34/57.



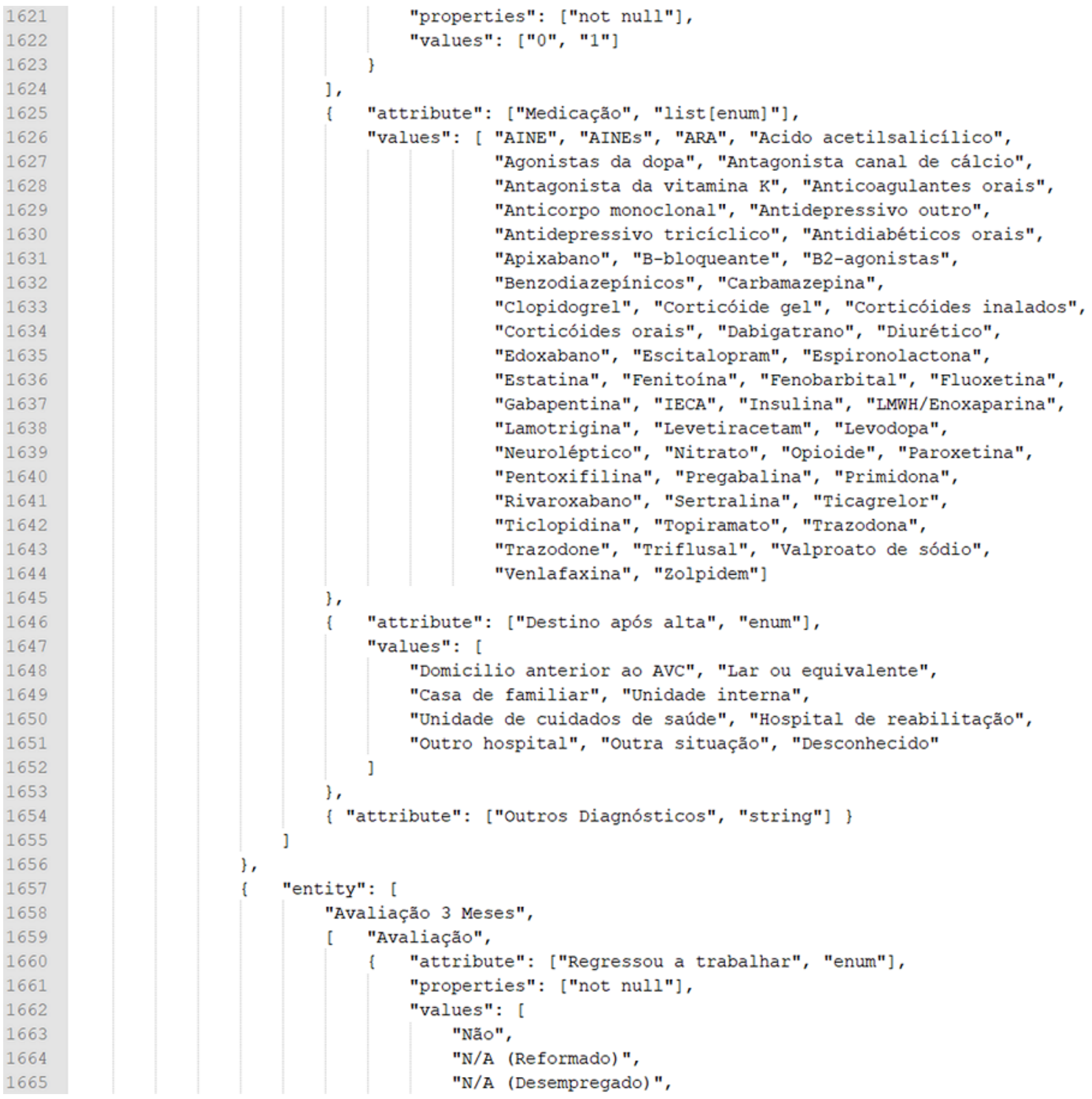
```
                                "properties": ["not null"],
                                "values": ["0", "1", "2", "3"]
                            },
                            {   "attribute": ["4)", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2", "3"]
                            },
                            {   "attribute": ["5)", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2", "3"]
                            },
                            {   "attribute": ["6)", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2", "3"]
                            },
                            {   "attribute": ["7)", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2", "3"]
                            },
                            {   "attribute": ["8)", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2", "3"]
                            },
                            {   "attribute": ["9)", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2", "3"]
                            },
                            {   "attribute": ["10)", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2", "3"]
                            },
                            {   "attribute": ["11)", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2", "3"]
                            },
                            {   "attribute": ["12)", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2", "3"]
                            },
                            {   "attribute": ["13)", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2", "3"]
                            },
                            {   "attribute": ["14)", "enum"],
                                "properties": ["not null"],
```

Figure C.35: PRECISE Stroke - JSON file specification 35/57.



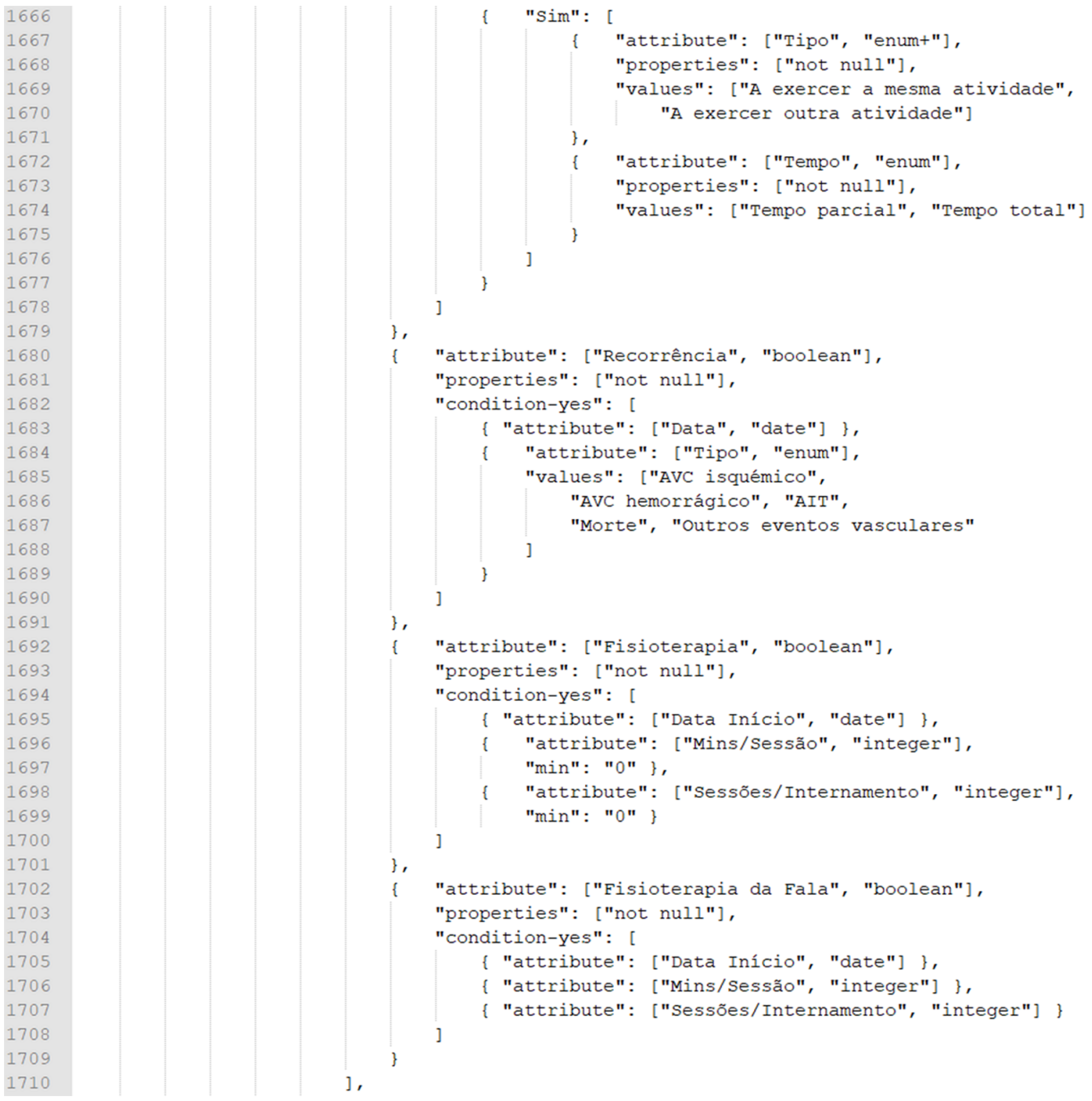
```
                                "values": ["0", "1", "2", "3"]
                            }
                        ],
                        [   "Mini-Mental State",
                            {   "attribute": ["1.a Orientação", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2", "3", "4", "5"]
                            },
                            {   "attribute": ["1.b Orientação", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2", "3", "4", "5"]
                            },
                            {   "attribute": ["2. Retenção", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2", "3"]
                            },
                            {   "attribute": ["3. Atenção e Cálculo", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2", "3", "4", "5"]
                            },
                            {   "attribute": ["4. Evocação", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2", "3"]
                            },
                            {   "attribute": ["5.a Linguagem", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2"]
                            },
                            {   "attribute": ["5.b Linguagem", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1"]
                            },
                            {   "attribute": ["5.c Linguagem", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2", "3"]
                            },
                            {   "attribute": ["5.d Linguagem", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1"]
                            },
                            {   "attribute": ["5.e Linguagem", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1"]
                            },
                            {   "attribute": ["6. Habilidade Construtiva", "enum"],
```

Figure C.36: PRECISE Stroke - JSON file specification 36/57.



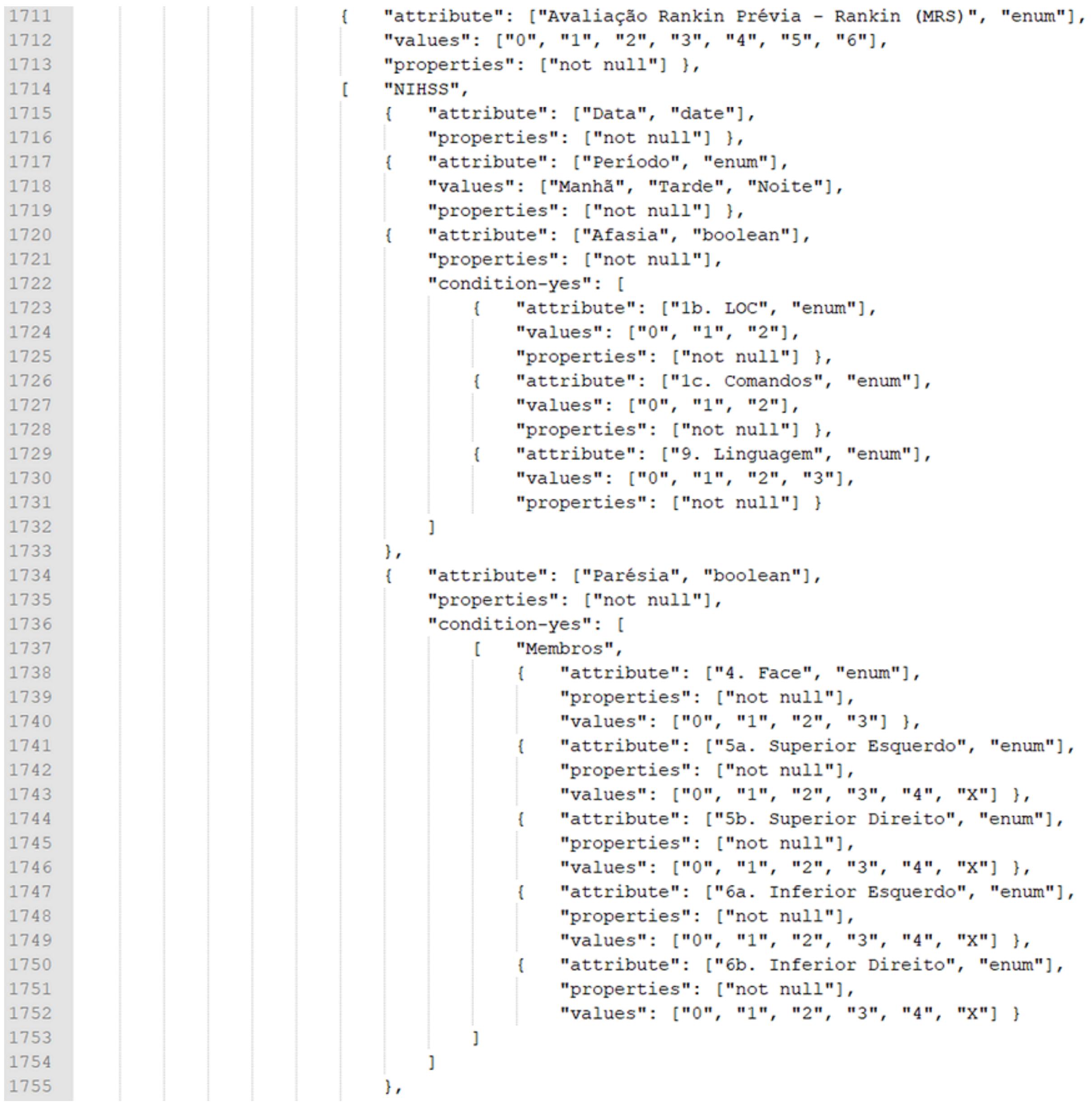
```
                                "properties": ["not null"],
                                "values": ["0", "1"]
                            }
                        ],
                        {   "attribute": ["Medicação", "list[enum]"],
                            "values": [ "AINE", "AINEs", "ARA", "Acido acetilsalicílico",
                                        "Agonistas da dopa", "Antagonista canal de cálcio",
                                        "Antagonista da vitamina K", "Anticoagulantes orais",
                                        "Anticorpo monoclonal", "Antidepressivo outro",
                                        "Antidepressivo tricíclico", "Antidiabéticos orais",
                                        "Apixabano", "B-bloqueante", "B2-agonistas",
                                        "Benzodiazepínicos", "Carbamazepina",
                                        "Clopidogrel", "Corticóide gel", "Corticóides inalados",
                                        "Corticóides orais", "Dabigatrano", "Diurético",
                                        "Edoxabano", "Escitalopram", "Espironolactona",
                                        "Estatina", "Fenitoína", "Fenobarbital", "Fluoxetina",
                                        "Gabapentina", "IECA", "Insulina", "LMWH/Enoxaparina",
                                        "Lamotrigina", "Levetiracetam", "Levodopa",
                                        "Neuroléptico", "Nitrato", "Opioide", "Paroxetina",
                                        "Pentoxifilina", "Pregabalina", "Primidona",
                                        "Rivaroxabano", "Sertralina", "Ticagrelor",
                                        "Ticlopidina", "Topiramato", "Trazodona",
                                        "Trazodone", "Triflusal", "Valproato de sódio",
                                        "Venlafaxina", "Zolpidem"]
                        },
                        {   "attribute": ["Destino após alta", "enum"],
                            "values": [
                                "Domicilio anterior ao AVC", "Lar ou equivalente",
                                "Casa de familiar", "Unidade interna",
                                "Unidade de cuidados de saúde", "Hospital de reabilitação",
                                "Outro hospital", "Outra situação", "Desconhecido"
                            ]
                        },
                        { "attribute": ["Outros Diagnósticos", "string"] }
                    ]
                },
                {   "entity": [
                        "Avaliação 3 Meses",
                        [   "Avaliação",
                            {   "attribute": ["Regressou a trabalhar", "enum"],
                                "properties": ["not null"],
                                "values": [
                                    "Não",
                                    "N/A (Reformado)",
                                    "N/A (Desempregado)",
```

Figure C.37: PRECISE Stroke - JSON file specification 37/57.



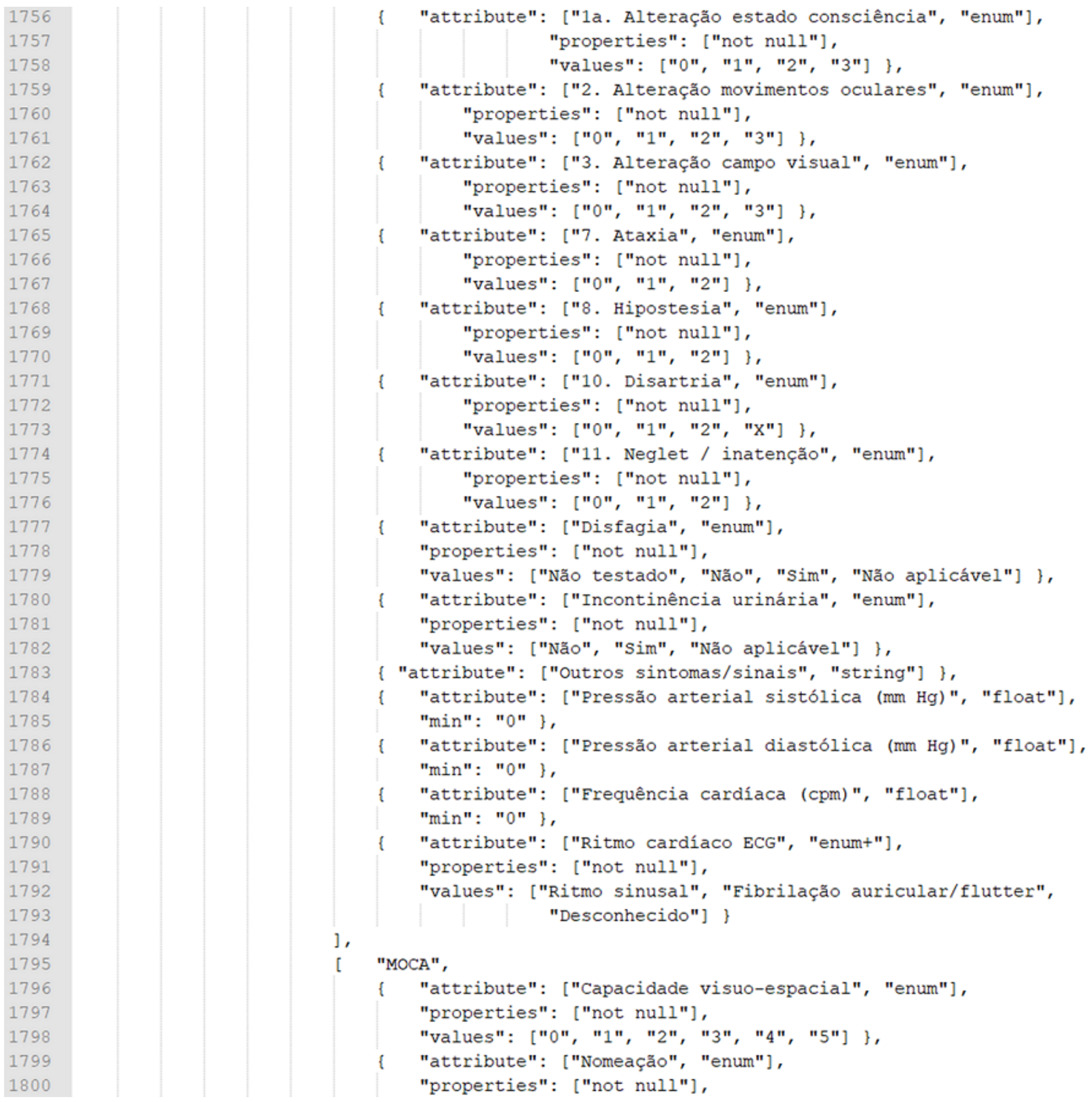
```
                                {   "Sim": [
                                        {   "attribute": ["Tipo", "enum+"],
                                            "properties": ["not null"],
                                            "values": ["A exercer a mesma atividade",
                                                "A exercer outra atividade"]
                                        },
                                        {   "attribute": ["Tempo", "enum"],
                                            "properties": ["not null"],
                                            "values": ["Tempo parcial", "Tempo total"]
                                        }
                                    ]
                                }
                            ]
                        },
                        {   "attribute": ["Recorrência", "boolean"],
                            "properties": ["not null"],
                            "condition-yes": [
                                { "attribute": ["Data", "date"] },
                                {   "attribute": ["Tipo", "enum"],
                                    "values": ["AVC isquémico",
                                        "AVC hemorrágico", "AIT",
                                        "Morte", "Outros eventos vasculares"
                                    ]
                                }
                            ]
                        },
                        {   "attribute": ["Fisioterapia", "boolean"],
                            "properties": ["not null"],
                            "condition-yes": [
                                { "attribute": ["Data Início", "date"] },
                                {   "attribute": ["Mins/Sessão", "integer"],
                                    "min": "0" },
                                {   "attribute": ["Sessões/Internamento", "integer"],
                                    "min": "0" }
                            ]
                        },
                        {   "attribute": ["Fisioterapia da Fala", "boolean"],
                            "properties": ["not null"],
                            "condition-yes": [
                                { "attribute": ["Data Início", "date"] },
                                { "attribute": ["Mins/Sessão", "integer"] },
                                { "attribute": ["Sessões/Internamento", "integer"] }
                            ]
                        }
                    ],
```

Figure C.38: PRECISE Stroke - JSON file specification 38/57.



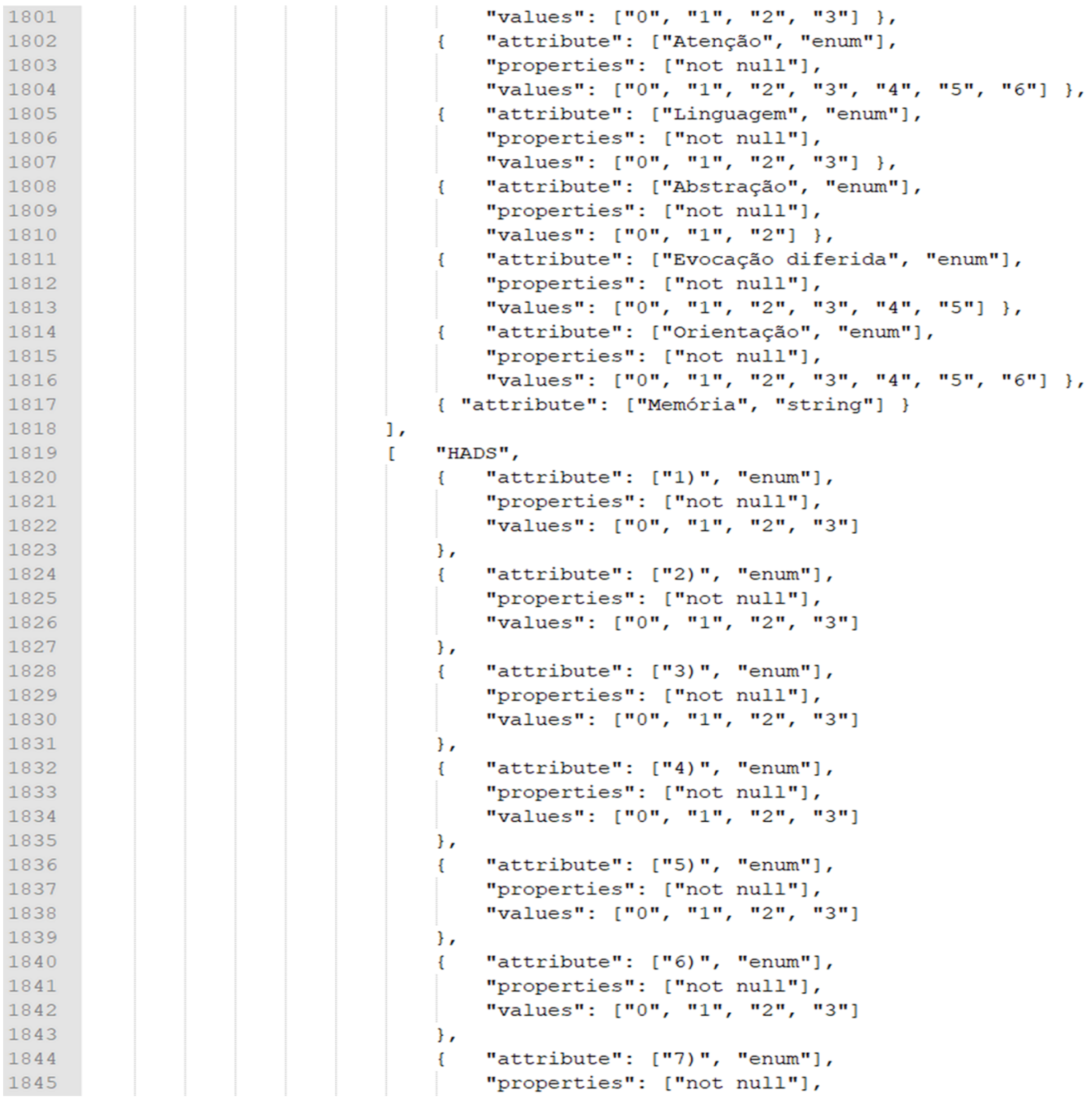
```
{   "attribute": ["Avaliação Rankin Prévia - Rankin (MRS)", "enum"],
    "values": ["0", "1", "2", "3", "4", "5", "6"],
    "properties": ["not null"] },
[   "NIHSS",
    {   "attribute": ["Data", "date"],
        "properties": ["not null"] },
    {   "attribute": ["Período", "enum"],
        "values": ["Manhã", "Tarde", "Noite"],
        "properties": ["not null"] },
    {   "attribute": ["Afasia", "boolean"],
        "properties": ["not null"],
        "condition-yes": [
            {   "attribute": ["1b. LOC", "enum"],
                "values": ["0", "1", "2"],
                "properties": ["not null"] },
            {   "attribute": ["1c. Comandos", "enum"],
                "values": ["0", "1", "2"],
                "properties": ["not null"] },
            {   "attribute": ["9. Linguagem", "enum"],
                "values": ["0", "1", "2", "3"],
                "properties": ["not null"] }
        ]
    },
    {   "attribute": ["Parésia", "boolean"],
        "properties": ["not null"],
        "condition-yes": [
            [   "Membros",
                {   "attribute": ["4. Face", "enum"],
                    "properties": ["not null"],
                    "values": ["0", "1", "2", "3"] },
                {   "attribute": ["5a. Superior Esquerdo", "enum"],
                    "properties": ["not null"],
                    "values": ["0", "1", "2", "3", "4", "X"] },
                {   "attribute": ["5b. Superior Direito", "enum"],
                    "properties": ["not null"],
                    "values": ["0", "1", "2", "3", "4", "X"] },
                {   "attribute": ["6a. Inferior Esquerdo", "enum"],
                    "properties": ["not null"],
                    "values": ["0", "1", "2", "3", "4", "X"] },
                {   "attribute": ["6b. Inferior Direito", "enum"],
                    "properties": ["not null"],
                    "values": ["0", "1", "2", "3", "4", "X"] }
            ]
        ]
    },
```

Figure C.39: PRECISE Stroke - JSON file specification 39/57.



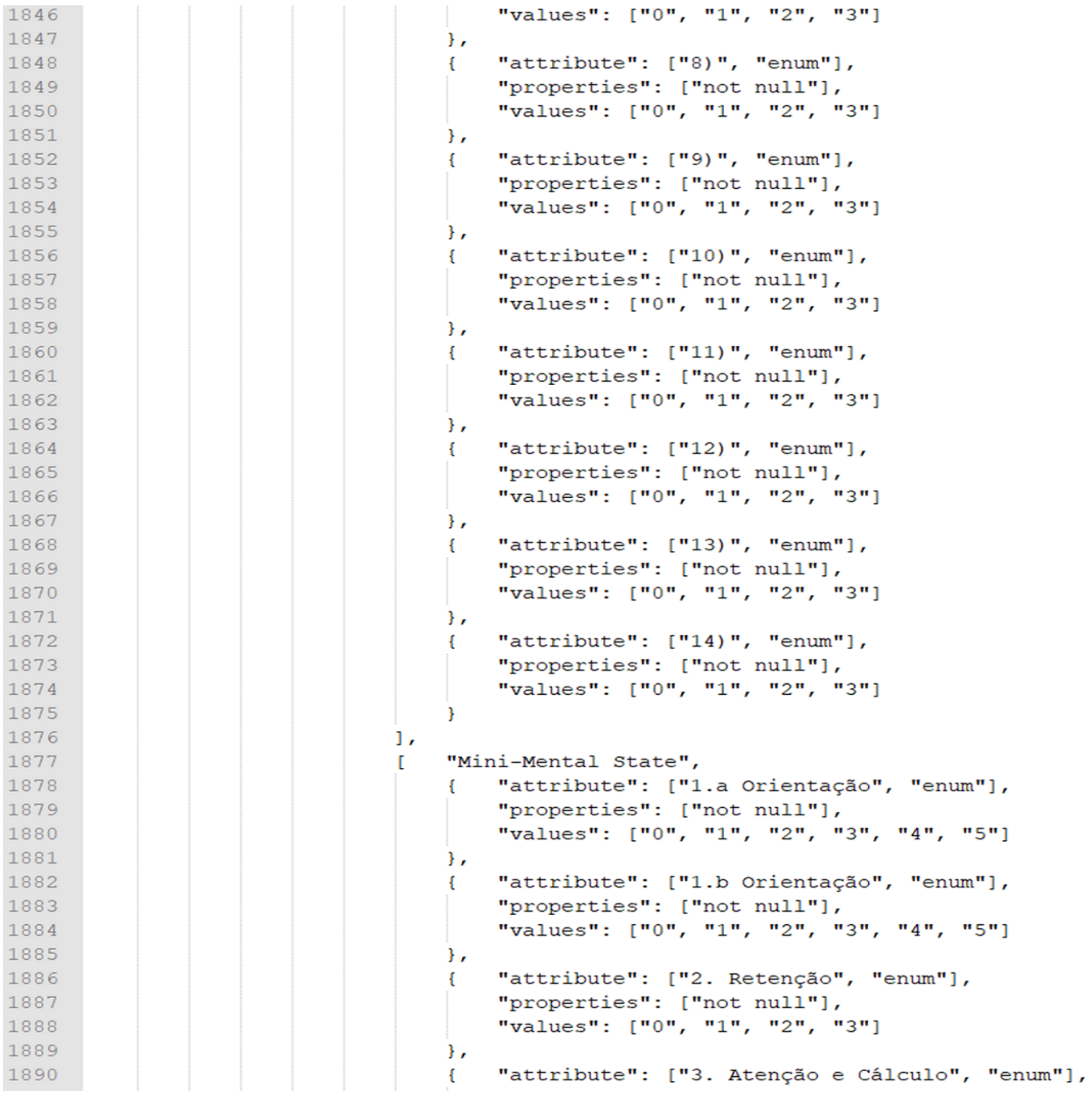
```
                            {   "attribute": ["1a. Alteração estado consciência", "enum"],
                                            "properties": ["not null"],
                                            "values": ["0", "1", "2", "3"] },
                            {   "attribute": ["2. Alteração movimentos oculares", "enum"],
                                    "properties": ["not null"],
                                    "values": ["0", "1", "2", "3"] },
                            {   "attribute": ["3. Alteração campo visual", "enum"],
                                    "properties": ["not null"],
                                    "values": ["0", "1", "2", "3"] },
                            {   "attribute": ["7. Ataxia", "enum"],
                                    "properties": ["not null"],
                                    "values": ["0", "1", "2"] },
                            {   "attribute": ["8. Hipostesia", "enum"],
                                    "properties": ["not null"],
                                    "values": ["0", "1", "2"] },
                            {   "attribute": ["10. Disartria", "enum"],
                                    "properties": ["not null"],
                                    "values": ["0", "1", "2", "X"] },
                            {   "attribute": ["11. Neglet / inatenção", "enum"],
                                    "properties": ["not null"],
                                    "values": ["0", "1", "2"] },
                            {   "attribute": ["Disfagia", "enum"],
                                "properties": ["not null"],
                                "values": ["Não testado", "Não", "Sim", "Não aplicável"] },
                            {   "attribute": ["Incontinência urinária", "enum"],
                                "properties": ["not null"],
                                "values": ["Não", "Sim", "Não aplicável"] },
                            { "attribute": ["Outros sintomas/sinais", "string"] },
                            {   "attribute": ["Pressão arterial sistólica (mm Hg)", "float"],
                                "min": "0" },
                            {   "attribute": ["Pressão arterial diastólica (mm Hg)", "float"],
                                "min": "0" },
                            {   "attribute": ["Frequência cardíaca (cpm)", "float"],
                                "min": "0" },
                            {   "attribute": ["Ritmo cardíaco ECG", "enum+"],
                                "properties": ["not null"],
                                "values": ["Ritmo sinusal", "Fibrilação auricular/flutter",
                                            "Desconhecido"] }
                        ],
                        [   "MOCA",
                            {   "attribute": ["Capacidade visuo-espacial", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2", "3", "4", "5"] },
                            {   "attribute": ["Nomeação", "enum"],
                                "properties": ["not null"],
```

Figure C.40: PRECISE Stroke - JSON file specification 40/57.



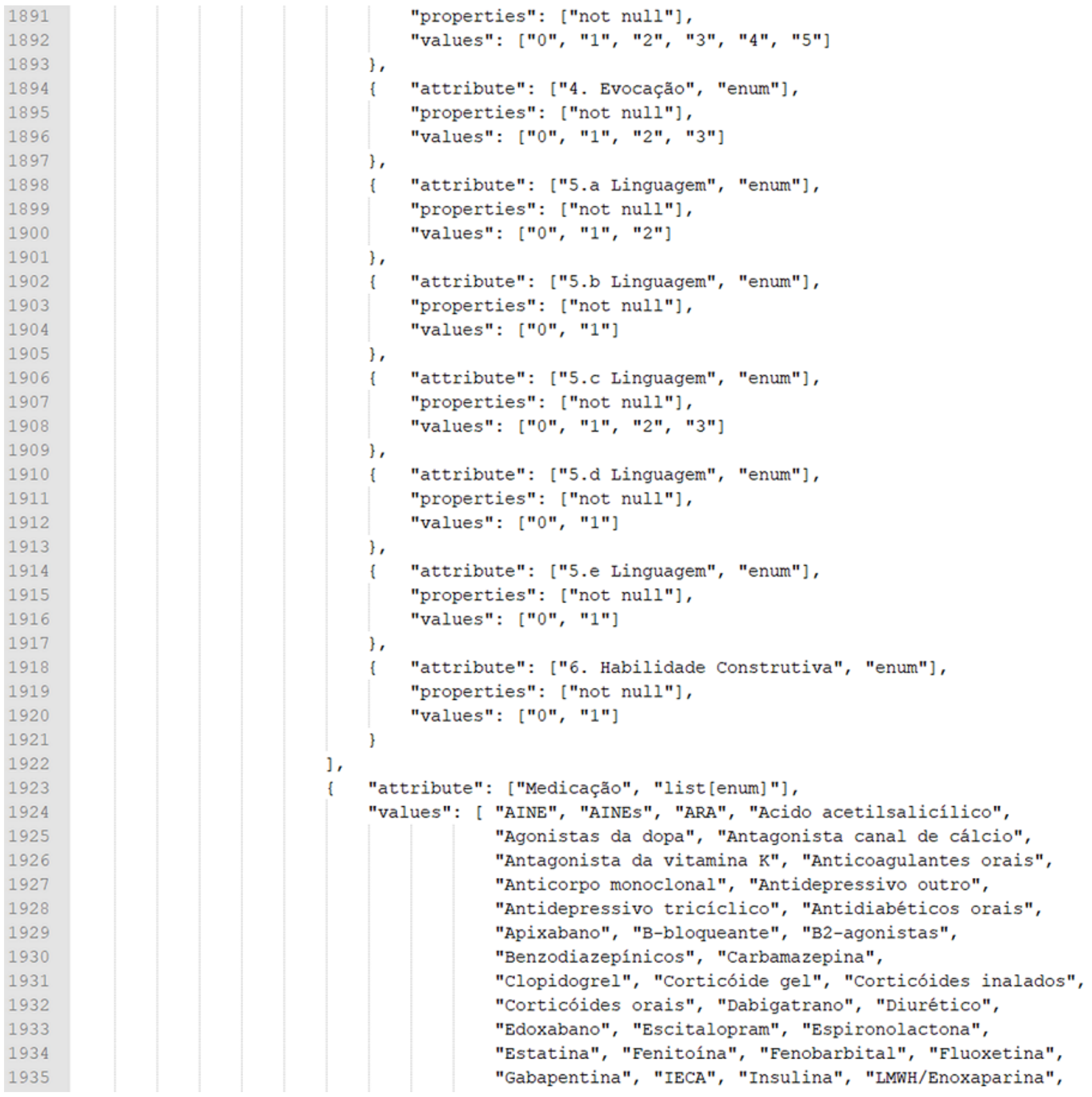
```
                                "values": ["0", "1", "2", "3"] },
                            {   "attribute": ["Atenção", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2", "3", "4", "5", "6"] },
                            {   "attribute": ["Linguagem", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2", "3"] },
                            {   "attribute": ["Abstração", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2"] },
                            {   "attribute": ["Evocação diferida", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2", "3", "4", "5"] },
                            {   "attribute": ["Orientação", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2", "3", "4", "5", "6"] },
                            { "attribute": ["Memória", "string"] }
                        ],
                        [   "HADS",
                            {   "attribute": ["1)", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2", "3"]
                            },
                            {   "attribute": ["2)", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2", "3"]
                            },
                            {   "attribute": ["3)", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2", "3"]
                            },
                            {   "attribute": ["4)", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2", "3"]
                            },
                            {   "attribute": ["5)", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2", "3"]
                            },
                            {   "attribute": ["6)", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1", "2", "3"]
                            },
                            {   "attribute": ["7)", "enum"],
                                "properties": ["not null"],
```

Figure C.41: PRECISE Stroke - JSON file specification 41/57.



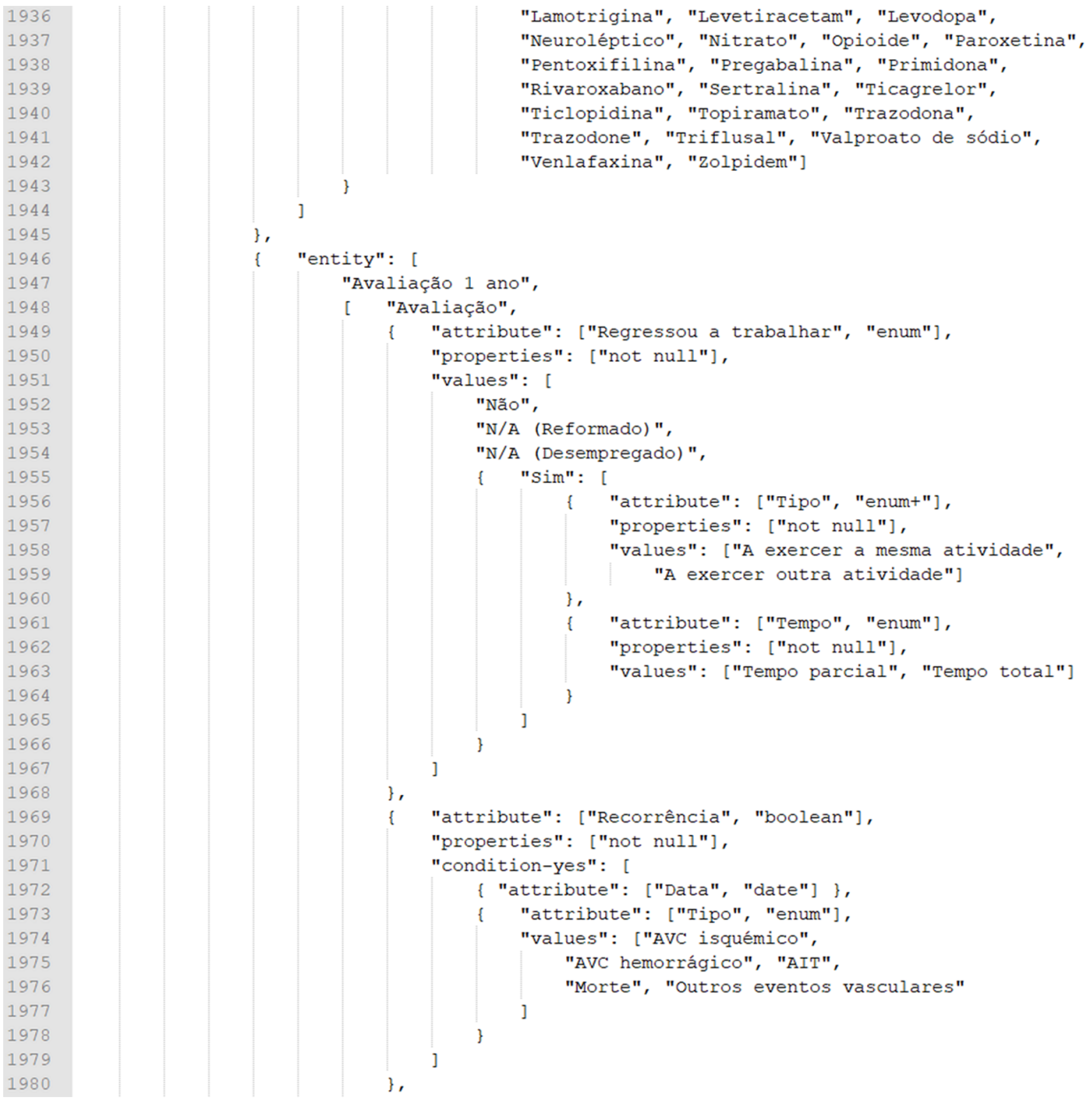
```
                                    "values": ["0", "1", "2", "3"]
                                },
                                {   "attribute": ["8)", "enum"],
                                    "properties": ["not null"],
                                    "values": ["0", "1", "2", "3"]
                                },
                                {   "attribute": ["9)", "enum"],
                                    "properties": ["not null"],
                                    "values": ["0", "1", "2", "3"]
                                },
                                {   "attribute": ["10)", "enum"],
                                    "properties": ["not null"],
                                    "values": ["0", "1", "2", "3"]
                                },
                                {   "attribute": ["11)", "enum"],
                                    "properties": ["not null"],
                                    "values": ["0", "1", "2", "3"]
                                },
                                {   "attribute": ["12)", "enum"],
                                    "properties": ["not null"],
                                    "values": ["0", "1", "2", "3"]
                                },
                                {   "attribute": ["13)", "enum"],
                                    "properties": ["not null"],
                                    "values": ["0", "1", "2", "3"]
                                },
                                {   "attribute": ["14)", "enum"],
                                    "properties": ["not null"],
                                    "values": ["0", "1", "2", "3"]
                                }
                            ],
                            [   "Mini-Mental State",
                                {   "attribute": ["1.a Orientação", "enum"],
                                    "properties": ["not null"],
                                    "values": ["0", "1", "2", "3", "4", "5"]
                                },
                                {   "attribute": ["1.b Orientação", "enum"],
                                    "properties": ["not null"],
                                    "values": ["0", "1", "2", "3", "4", "5"]
                                },
                                {   "attribute": ["2. Retenção", "enum"],
                                    "properties": ["not null"],
                                    "values": ["0", "1", "2", "3"]
                                },
                                {   "attribute": ["3. Atenção e Cálculo", "enum"],
```

Figure C.42: PRECISE Stroke - JSON file specification 42/57.



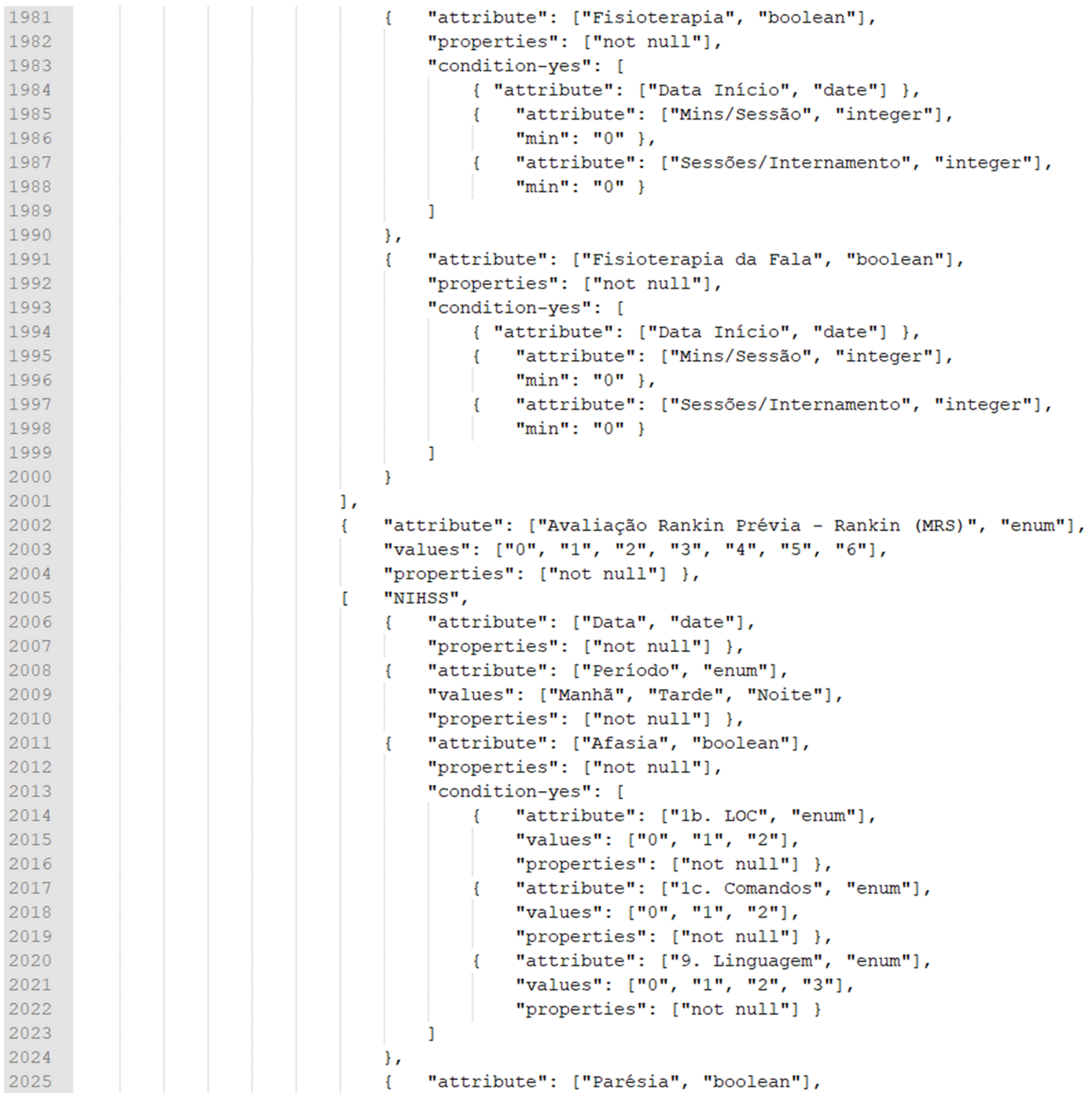
```
                "properties": ["not null"],
                "values": ["0", "1", "2", "3", "4", "5"]
            },
            {   "attribute": ["4. Evocação", "enum"],
                "properties": ["not null"],
                "values": ["0", "1", "2", "3"]
            },
            {   "attribute": ["5.a Linguagem", "enum"],
                "properties": ["not null"],
                "values": ["0", "1", "2"]
            },
            {   "attribute": ["5.b Linguagem", "enum"],
                "properties": ["not null"],
                "values": ["0", "1"]
            },
            {   "attribute": ["5.c Linguagem", "enum"],
                "properties": ["not null"],
                "values": ["0", "1", "2", "3"]
            },
            {   "attribute": ["5.d Linguagem", "enum"],
                "properties": ["not null"],
                "values": ["0", "1"]
            },
            {   "attribute": ["5.e Linguagem", "enum"],
                "properties": ["not null"],
                "values": ["0", "1"]
            },
            {   "attribute": ["6. Habilidade Construtiva", "enum"],
                "properties": ["not null"],
                "values": ["0", "1"]
            }
        ],
        {   "attribute": ["Medicação", "list[enum]"],
            "values": [ "AINE", "AINEs", "ARA", "Acido acetilsalicílico",
                        "Agonistas da dopa", "Antagonista canal de cálcio",
                        "Antagonista da vitamina K", "Anticoagulantes orais",
                        "Anticorpo monoclonal", "Antidepressivo outro",
                        "Antidepressivo tricíclico", "Antidiabéticos orais",
                        "Apixabano", "B-bloqueante", "B2-agonistas",
                        "Benzodiazepínicos", "Carbamazepina",
                        "Clopidogrel", "Corticóide gel", "Corticóides inalados",
                        "Corticóides orais", "Dabigatrano", "Diurético",
                        "Edoxabano", "Escitalopram", "Espironolactona",
                        "Estatina", "Fenitoína", "Fenobarbital", "Fluoxetina",
                        "Gabapentina", "IECA", "Insulina", "LMWH/Enoxaparina",
```

Figure C.43: PRECISE Stroke - JSON file specification 43/57.



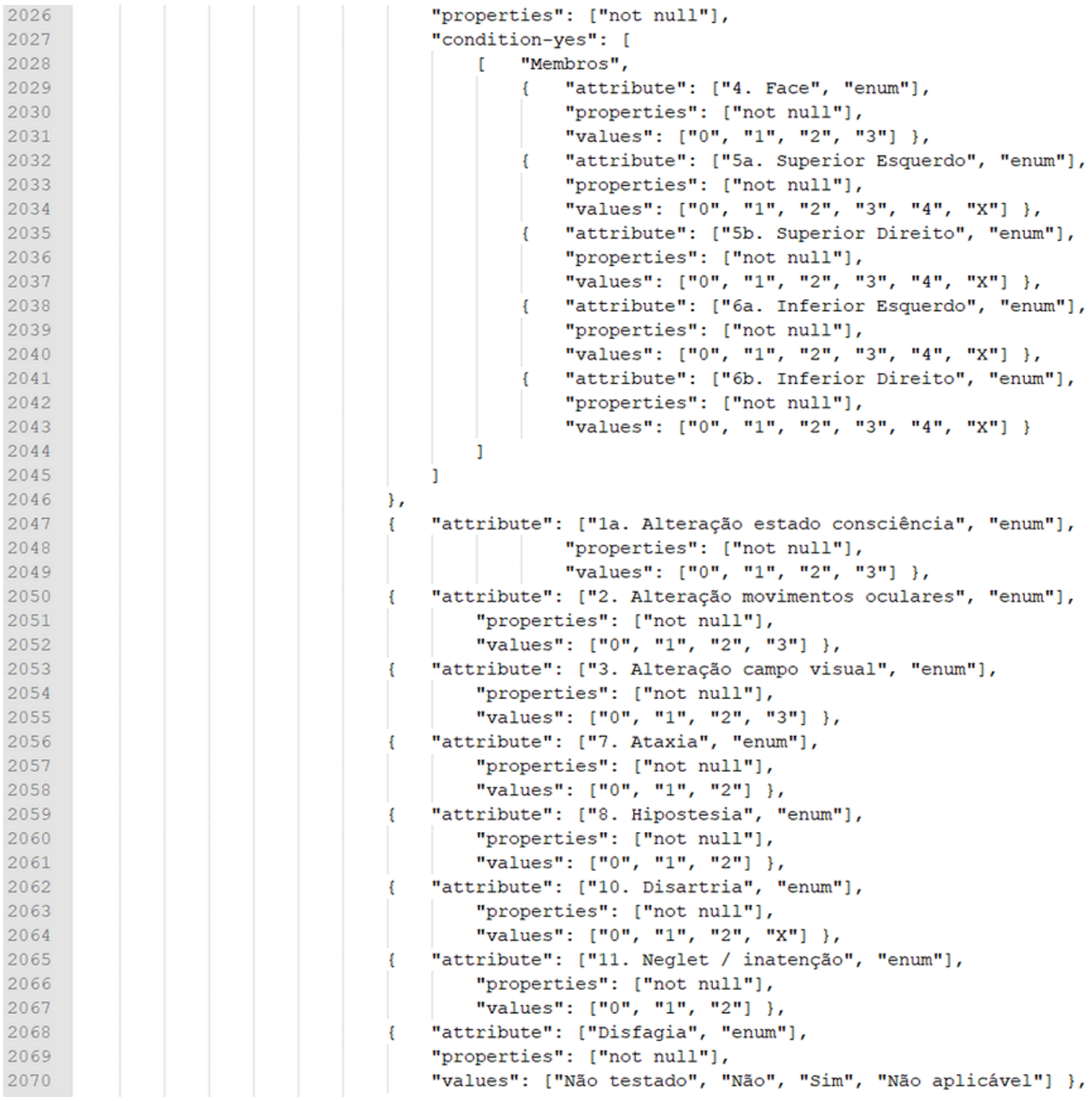
```
                                        "Lamotrigina", "Levetiracetam", "Levodopa",
                                        "Neuroléptico", "Nitrato", "Opioide", "Paroxetina",
                                        "Pentoxifilina", "Pregabalina", "Primidona",
                                        "Rivaroxabano", "Sertralina", "Ticagrelor",
                                        "Ticlopidina", "Topiramato", "Trazodona",
                                        "Trazodone", "Triflusal", "Valproato de sódio",
                                        "Venlafaxina", "Zolpidem"]
                        }
                    ]
                },
                {   "entity": [
                        "Avaliação 1 ano",
                        [   "Avaliação",
                            {   "attribute": ["Regressou a trabalhar", "enum"],
                                "properties": ["not null"],
                                "values": [
                                    "Não",
                                    "N/A (Reformado)",
                                    "N/A (Desempregado)",
                                    {   "Sim": [
                                            {   "attribute": ["Tipo", "enum+"],
                                                "properties": ["not null"],
                                                "values": ["A exercer a mesma atividade",
                                                    "A exercer outra atividade"]
                                            },
                                            {   "attribute": ["Tempo", "enum"],
                                                "properties": ["not null"],
                                                "values": ["Tempo parcial", "Tempo total"]
                                            }
                                        ]
                                    }
                                ]
                            },
                            {   "attribute": ["Recorrência", "boolean"],
                                "properties": ["not null"],
                                "condition-yes": [
                                    { "attribute": ["Data", "date"] },
                                    {   "attribute": ["Tipo", "enum"],
                                        "values": ["AVC isquémico",
                                            "AVC hemorrágico", "AIT",
                                            "Morte", "Outros eventos vasculares"
                                        ]
                                    }
                                ]
                            },
```

Figure C.44: PRECISE Stroke - JSON file specification 44/57.



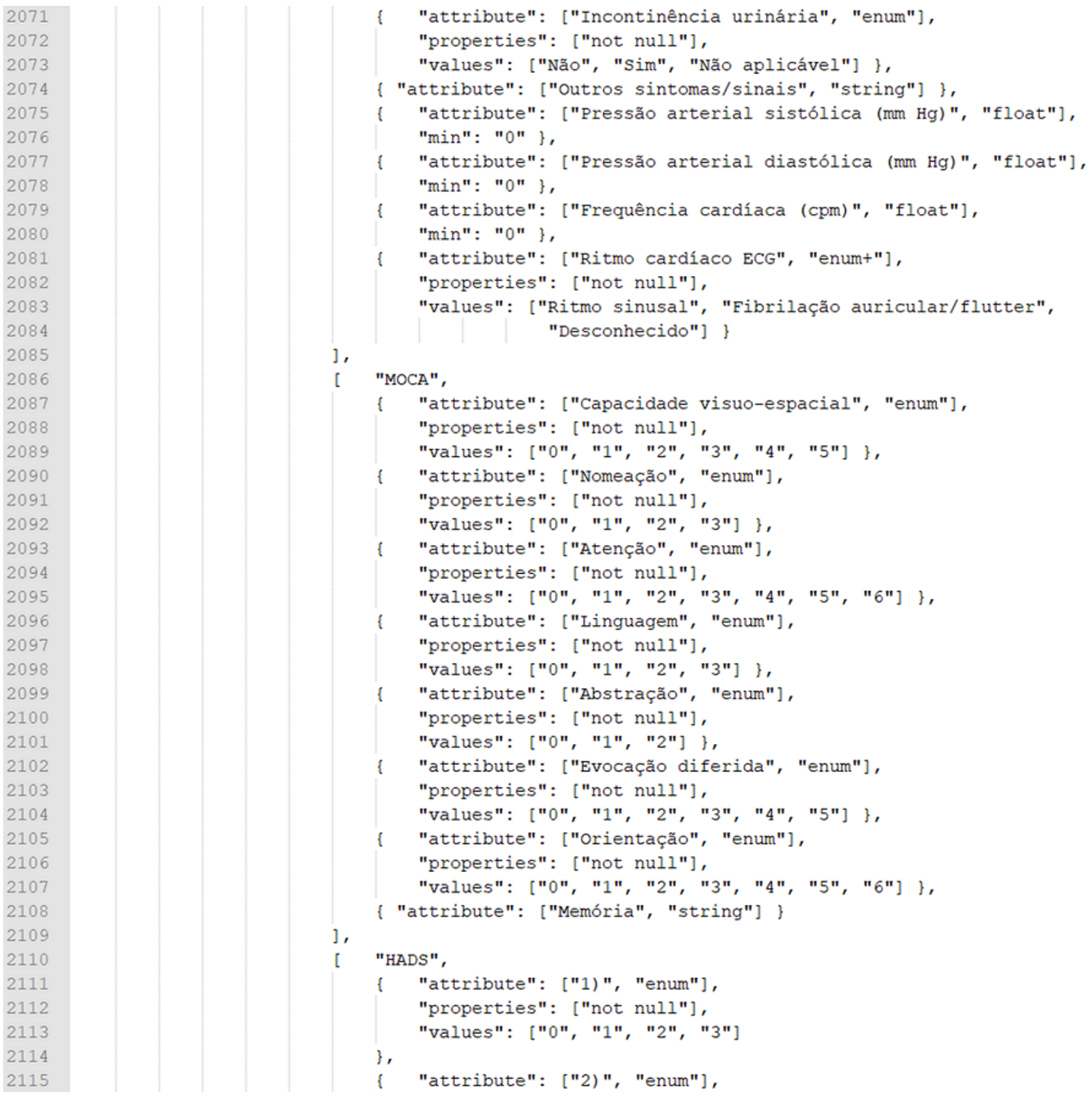
```
                            {   "attribute": ["Fisioterapia", "boolean"],
                                "properties": ["not null"],
                                "condition-yes": [
                                    { "attribute": ["Data Início", "date"] },
                                    {   "attribute": ["Mins/Sessão", "integer"],
                                        "min": "0" },
                                    {   "attribute": ["Sessões/Internamento", "integer"],
                                        "min": "0" }
                                ]
                            },
                            {   "attribute": ["Fisioterapia da Fala", "boolean"],
                                "properties": ["not null"],
                                "condition-yes": [
                                    { "attribute": ["Data Início", "date"] },
                                    {   "attribute": ["Mins/Sessão", "integer"],
                                        "min": "0" },
                                    {   "attribute": ["Sessões/Internamento", "integer"],
                                        "min": "0" }
                                ]
                            }
                        ],
                        {   "attribute": ["Avaliação Rankin Prévia - Rankin (MRS)", "enum"],
                            "values": ["0", "1", "2", "3", "4", "5", "6"],
                            "properties": ["not null"] },
                        [   "NIHSS",
                            {   "attribute": ["Data", "date"],
                                "properties": ["not null"] },
                            {   "attribute": ["Período", "enum"],
                                "values": ["Manhã", "Tarde", "Noite"],
                                "properties": ["not null"] },
                            {   "attribute": ["Afasia", "boolean"],
                                "properties": ["not null"],
                                "condition-yes": [
                                    {   "attribute": ["1b. LOC", "enum"],
                                        "values": ["0", "1", "2"],
                                        "properties": ["not null"] },
                                    {   "attribute": ["1c. Comandos", "enum"],
                                        "values": ["0", "1", "2"],
                                        "properties": ["not null"] },
                                    {   "attribute": ["9. Linguagem", "enum"],
                                        "values": ["0", "1", "2", "3"],
                                        "properties": ["not null"] }
                                ]
                            },
                            {   "attribute": ["Parésia", "boolean"],
```

Figure C.45: PRECISE Stroke - JSON file specification 45/57.



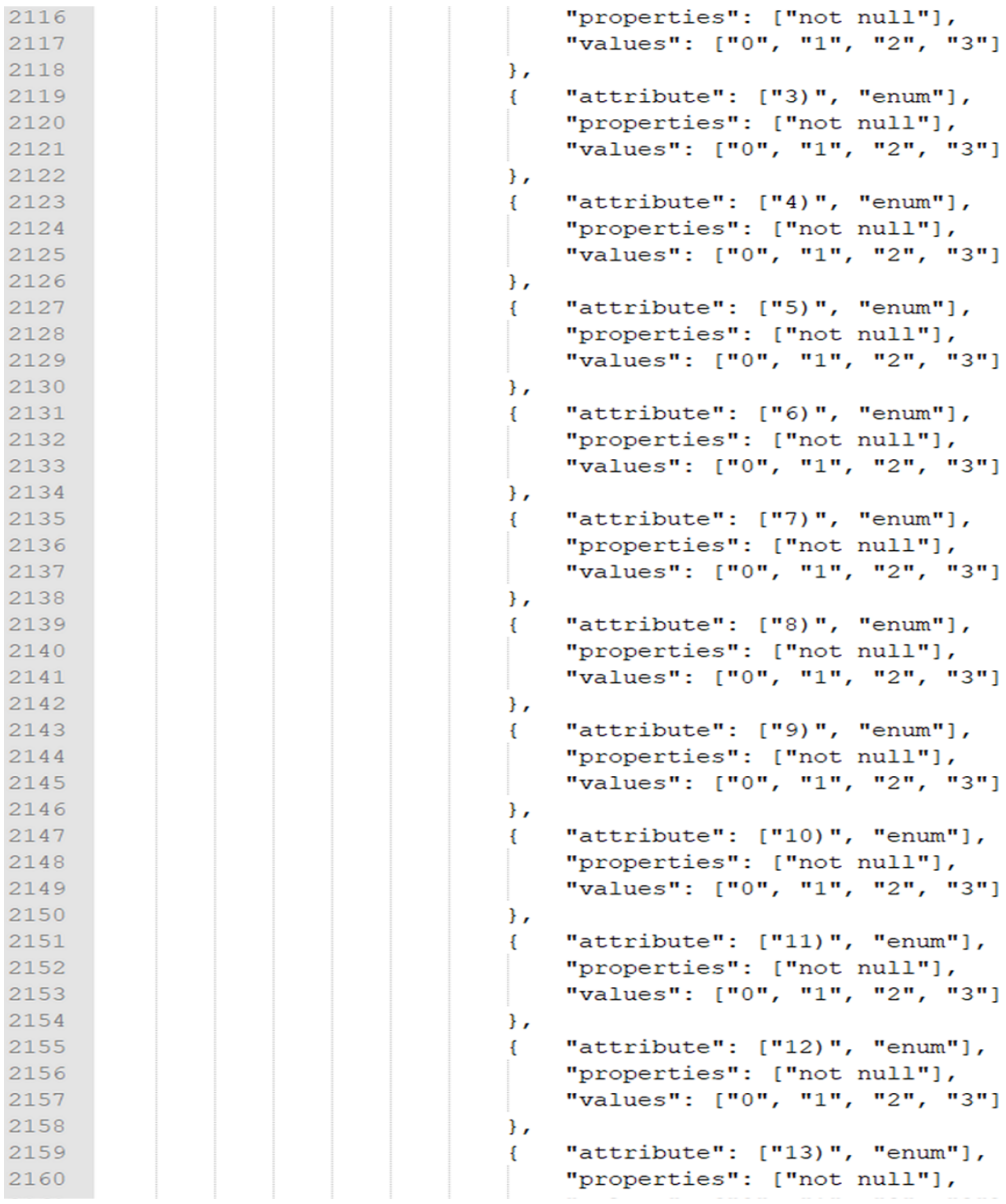
```
    "properties": ["not null"],
    "condition-yes": [
        [   "Membros",
            {   "attribute": ["4. Face", "enum"],
                "properties": ["not null"],
                "values": ["0", "1", "2", "3"] },
            {   "attribute": ["5a. Superior Esquerdo", "enum"],
                "properties": ["not null"],
                "values": ["0", "1", "2", "3", "4", "X"] },
            {   "attribute": ["5b. Superior Direito", "enum"],
                "properties": ["not null"],
                "values": ["0", "1", "2", "3", "4", "X"] },
            {   "attribute": ["6a. Inferior Esquerdo", "enum"],
                "properties": ["not null"],
                "values": ["0", "1", "2", "3", "4", "X"] },
            {   "attribute": ["6b. Inferior Direito", "enum"],
                "properties": ["not null"],
                "values": ["0", "1", "2", "3", "4", "X"] }
        ]
    ]
},
{   "attribute": ["1a. Alteração estado consciência", "enum"],
                "properties": ["not null"],
                "values": ["0", "1", "2", "3"] },
{   "attribute": ["2. Alteração movimentos oculares", "enum"],
        "properties": ["not null"],
        "values": ["0", "1", "2", "3"] },
{   "attribute": ["3. Alteração campo visual", "enum"],
        "properties": ["not null"],
        "values": ["0", "1", "2", "3"] },
{   "attribute": ["7. Ataxia", "enum"],
        "properties": ["not null"],
        "values": ["0", "1", "2"] },
{   "attribute": ["8. Hipostesia", "enum"],
        "properties": ["not null"],
        "values": ["0", "1", "2"] },
{   "attribute": ["10. Disartria", "enum"],
        "properties": ["not null"],
        "values": ["0", "1", "2", "X"] },
{   "attribute": ["11. Neglet / inatenção", "enum"],
        "properties": ["not null"],
        "values": ["0", "1", "2"] },
{   "attribute": ["Disfagia", "enum"],
    "properties": ["not null"],
    "values": ["Não testado", "Não", "Sim", "Não aplicável"] },
```

Figure C.46: PRECISE Stroke - JSON file specification 46/57.



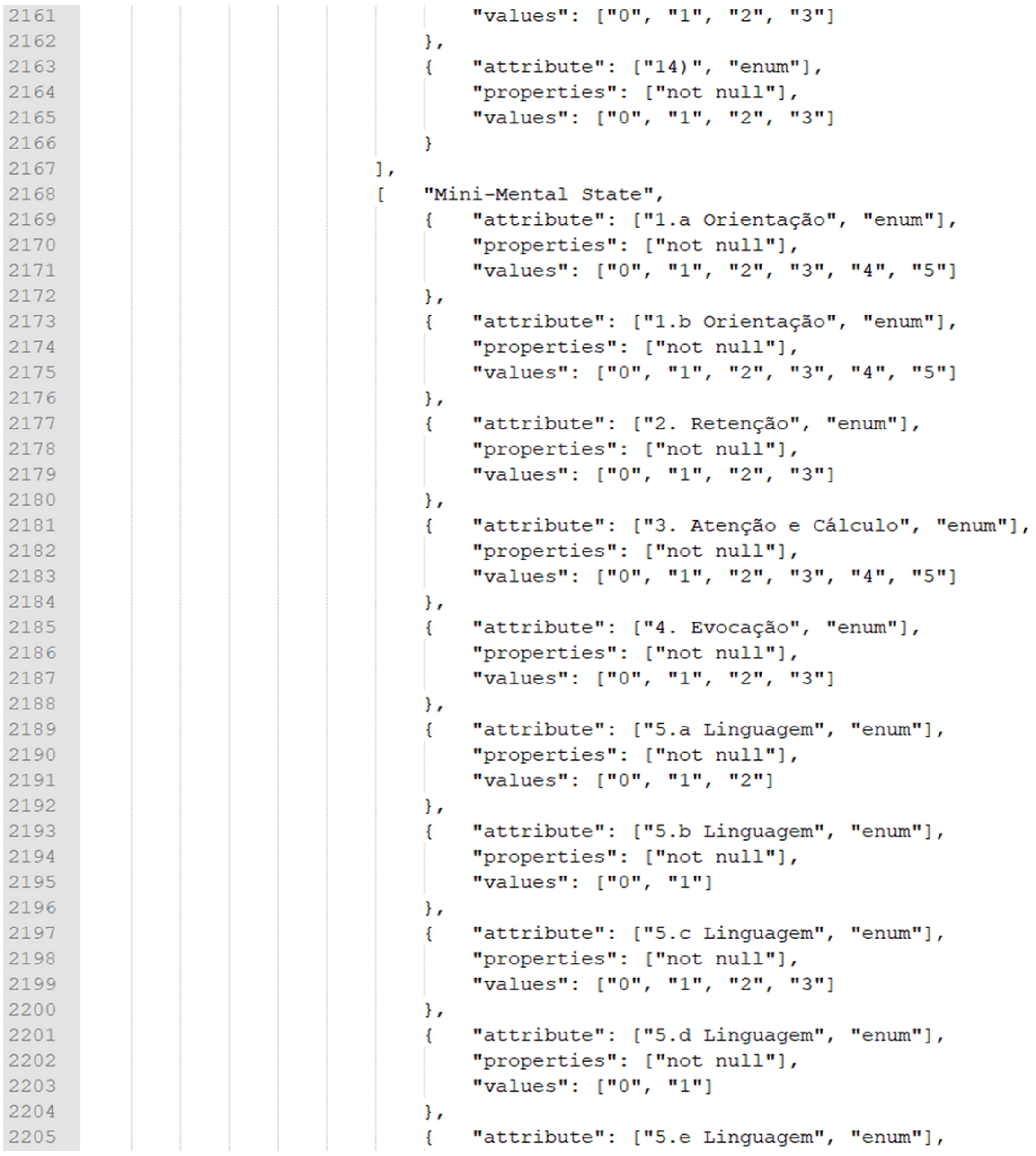
```
        {   "attribute": ["Incontinência urinária", "enum"],
            "properties": ["not null"],
            "values": ["Não", "Sim", "Não aplicável"] },
        { "attribute": ["Outros sintomas/sinais", "string"] },
        {   "attribute": ["Pressão arterial sistólica (mm Hg)", "float"],
            "min": "0" },
        {   "attribute": ["Pressão arterial diastólica (mm Hg)", "float"],
            "min": "0" },
        {   "attribute": ["Frequência cardíaca (cpm)", "float"],
            "min": "0" },
        {   "attribute": ["Ritmo cardíaco ECG", "enum+"],
            "properties": ["not null"],
            "values": ["Ritmo sinusal", "Fibrilação auricular/flutter",
                        "Desconhecido"] }
    ],
    [   "MOCA",
        {   "attribute": ["Capacidade visuo-espacial", "enum"],
            "properties": ["not null"],
            "values": ["0", "1", "2", "3", "4", "5"] },
        {   "attribute": ["Nomeação", "enum"],
            "properties": ["not null"],
            "values": ["0", "1", "2", "3"] },
        {   "attribute": ["Atenção", "enum"],
            "properties": ["not null"],
            "values": ["0", "1", "2", "3", "4", "5", "6"] },
        {   "attribute": ["Linguagem", "enum"],
            "properties": ["not null"],
            "values": ["0", "1", "2", "3"] },
        {   "attribute": ["Abstração", "enum"],
            "properties": ["not null"],
            "values": ["0", "1", "2"] },
        {   "attribute": ["Evocação diferida", "enum"],
            "properties": ["not null"],
            "values": ["0", "1", "2", "3", "4", "5"] },
        {   "attribute": ["Orientação", "enum"],
            "properties": ["not null"],
            "values": ["0", "1", "2", "3", "4", "5", "6"] },
        { "attribute": ["Memória", "string"] }
    ],
    [   "HADS",
        {   "attribute": ["1)", "enum"],
            "properties": ["not null"],
            "values": ["0", "1", "2", "3"]
        },
        {   "attribute": ["2)", "enum"],
```

Figure C.47: PRECISE Stroke - JSON file specification 47/57.



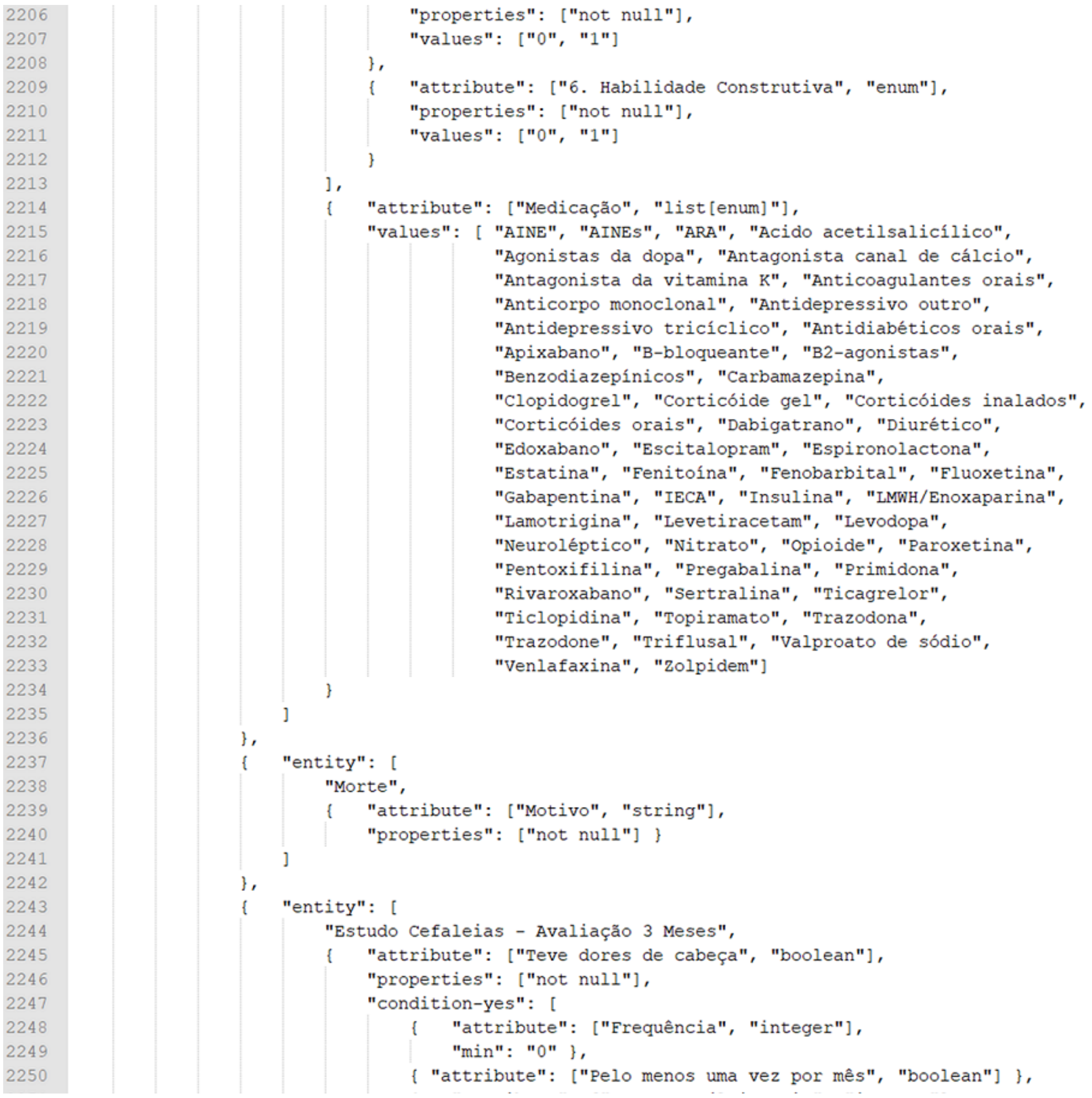
```
                                 "properties": ["not null"],
                                 "values": ["0", "1", "2", "3"]
                             },
                             {   "attribute": ["3)", "enum"],
                                 "properties": ["not null"],
                                 "values": ["0", "1", "2", "3"]
                             },
                             {   "attribute": ["4)", "enum"],
                                 "properties": ["not null"],
                                 "values": ["0", "1", "2", "3"]
                             },
                             {   "attribute": ["5)", "enum"],
                                 "properties": ["not null"],
                                 "values": ["0", "1", "2", "3"]
                             },
                             {   "attribute": ["6)", "enum"],
                                 "properties": ["not null"],
                                 "values": ["0", "1", "2", "3"]
                             },
                             {   "attribute": ["7)", "enum"],
                                 "properties": ["not null"],
                                 "values": ["0", "1", "2", "3"]
                             },
                             {   "attribute": ["8)", "enum"],
                                 "properties": ["not null"],
                                 "values": ["0", "1", "2", "3"]
                             },
                             {   "attribute": ["9)", "enum"],
                                 "properties": ["not null"],
                                 "values": ["0", "1", "2", "3"]
                             },
                             {   "attribute": ["10)", "enum"],
                                 "properties": ["not null"],
                                 "values": ["0", "1", "2", "3"]
                             },
                             {   "attribute": ["11)", "enum"],
                                 "properties": ["not null"],
                                 "values": ["0", "1", "2", "3"]
                             },
                             {   "attribute": ["12)", "enum"],
                                 "properties": ["not null"],
                                 "values": ["0", "1", "2", "3"]
                             },
                             {   "attribute": ["13)", "enum"],
                                 "properties": ["not null"],
```

Figure C.48: PRECISE Stroke - JSON file specification 48/57.



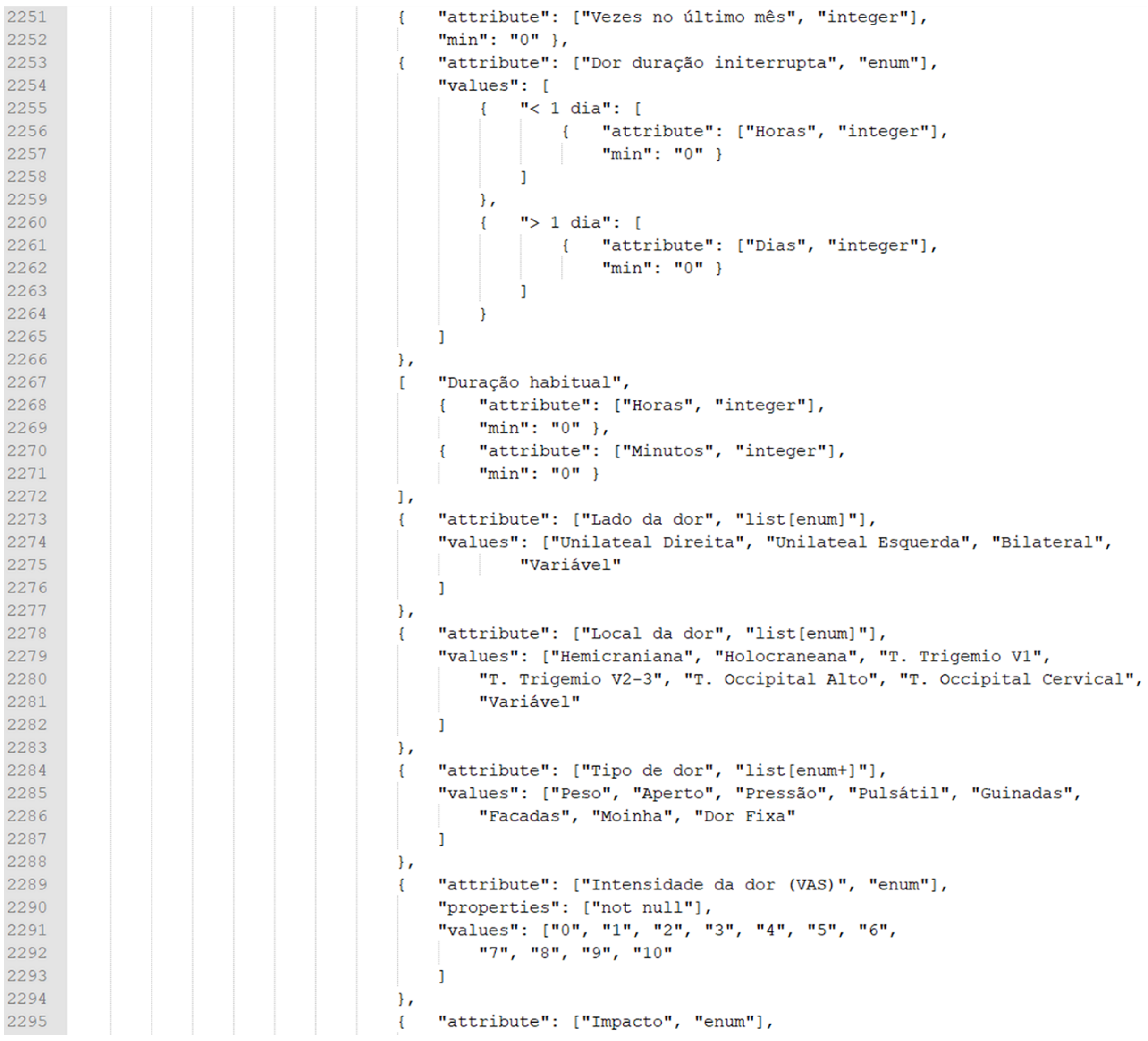
```
        "values": ["0", "1", "2", "3"]
    },
    {   "attribute": ["14)", "enum"],
        "properties": ["not null"],
        "values": ["0", "1", "2", "3"]
    }
],
[   "Mini-Mental State",
    {   "attribute": ["1.a Orientação", "enum"],
        "properties": ["not null"],
        "values": ["0", "1", "2", "3", "4", "5"]
    },
    {   "attribute": ["1.b Orientação", "enum"],
        "properties": ["not null"],
        "values": ["0", "1", "2", "3", "4", "5"]
    },
    {   "attribute": ["2. Retenção", "enum"],
        "properties": ["not null"],
        "values": ["0", "1", "2", "3"]
    },
    {   "attribute": ["3. Atenção e Cálculo", "enum"],
        "properties": ["not null"],
        "values": ["0", "1", "2", "3", "4", "5"]
    },
    {   "attribute": ["4. Evocação", "enum"],
        "properties": ["not null"],
        "values": ["0", "1", "2", "3"]
    },
    {   "attribute": ["5.a Linguagem", "enum"],
        "properties": ["not null"],
        "values": ["0", "1", "2"]
    },
    {   "attribute": ["5.b Linguagem", "enum"],
        "properties": ["not null"],
        "values": ["0", "1"]
    },
    {   "attribute": ["5.c Linguagem", "enum"],
        "properties": ["not null"],
        "values": ["0", "1", "2", "3"]
    },
    {   "attribute": ["5.d Linguagem", "enum"],
        "properties": ["not null"],
        "values": ["0", "1"]
    },
    {   "attribute": ["5.e Linguagem", "enum"],
```

Figure C.49: PRECISE Stroke - JSON file specification 49/57.



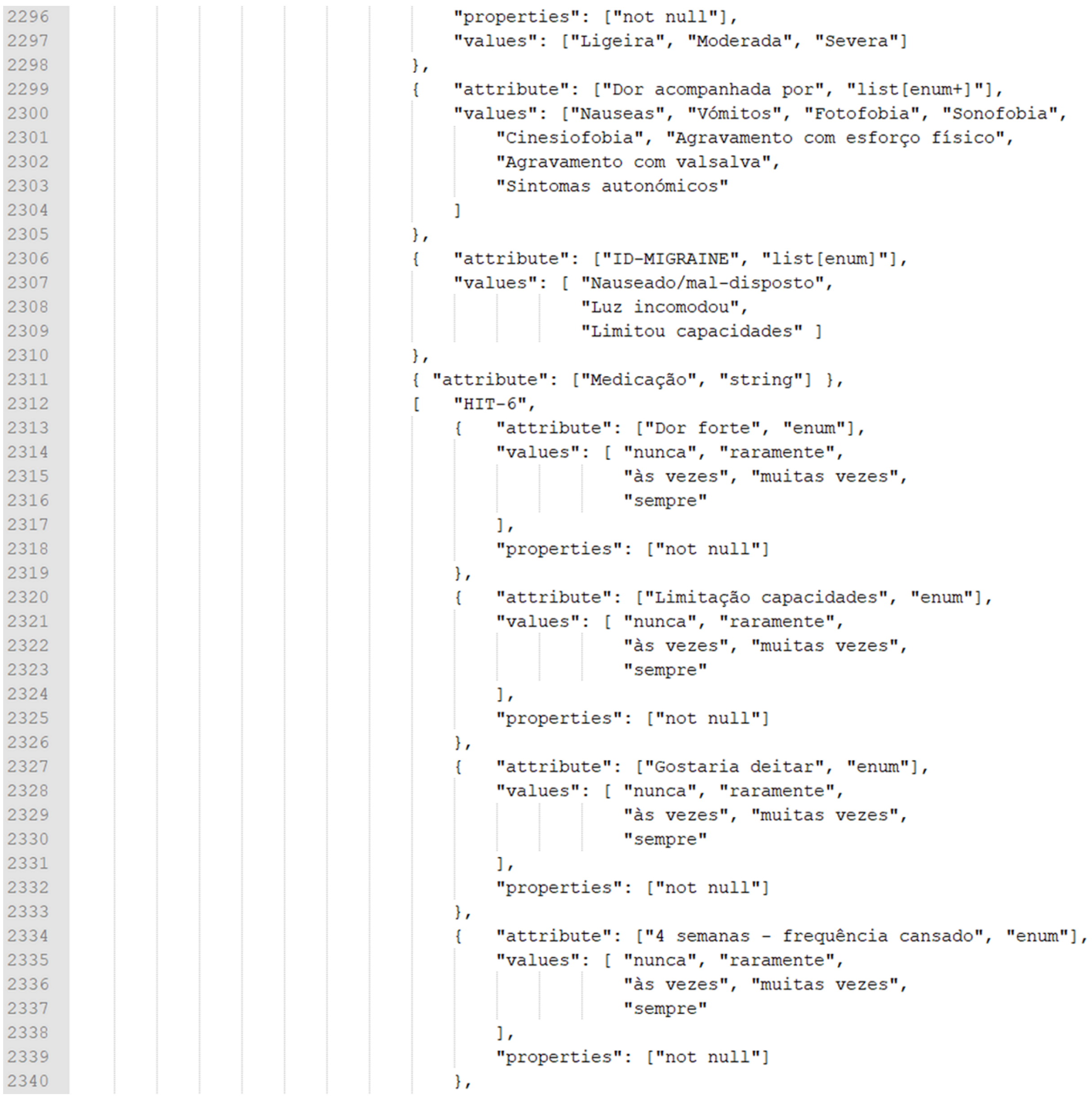
```
                                "properties": ["not null"],
                                "values": ["0", "1"]
                            },
                            {   "attribute": ["6. Habilidade Construtiva", "enum"],
                                "properties": ["not null"],
                                "values": ["0", "1"]
                            }
                        ],
                        {   "attribute": ["Medicação", "list[enum]"],
                            "values": [ "AINE", "AINEs", "ARA", "Acido acetilsalicílico",
                                        "Agonistas da dopa", "Antagonista canal de cálcio",
                                        "Antagonista da vitamina K", "Anticoagulantes orais",
                                        "Anticorpo monoclonal", "Antidepressivo outro",
                                        "Antidepressivo tricíclico", "Antidiabéticos orais",
                                        "Apixabano", "B-bloqueante", "B2-agonistas",
                                        "Benzodiazepínicos", "Carbamazepina",
                                        "Clopidogrel", "Corticóide gel", "Corticóides inalados",
                                        "Corticóides orais", "Dabigatrano", "Diurético",
                                        "Edoxabano", "Escitalopram", "Espironolactona",
                                        "Estatina", "Fenitoína", "Fenobarbital", "Fluoxetina",
                                        "Gabapentina", "IECA", "Insulina", "LMWH/Enoxaparina",
                                        "Lamotrigina", "Levetiracetam", "Levodopa",
                                        "Neuroléptico", "Nitrato", "Opioide", "Paroxetina",
                                        "Pentoxifilina", "Pregabalina", "Primidona",
                                        "Rivaroxabano", "Sertralina", "Ticagrelor",
                                        "Ticlopidina", "Topiramato", "Trazodona",
                                        "Trazodone", "Triflusal", "Valproato de sódio",
                                        "Venlafaxina", "Zolpidem"]
                        }
                    ]
                },
                {   "entity": [
                        "Morte",
                        {   "attribute": ["Motivo", "string"],
                            "properties": ["not null"] }
                    ]
                },
                {   "entity": [
                        "Estudo Cefaleias - Avaliação 3 Meses",
                        {   "attribute": ["Teve dores de cabeça", "boolean"],
                            "properties": ["not null"],
                            "condition-yes": [
                                {   "attribute": ["Frequência", "integer"],
                                    "min": "0" },
                                { "attribute": ["Pelo menos uma vez por mês", "boolean"] },
```

Figure C.50: PRECISE Stroke - JSON file specification 50/57.



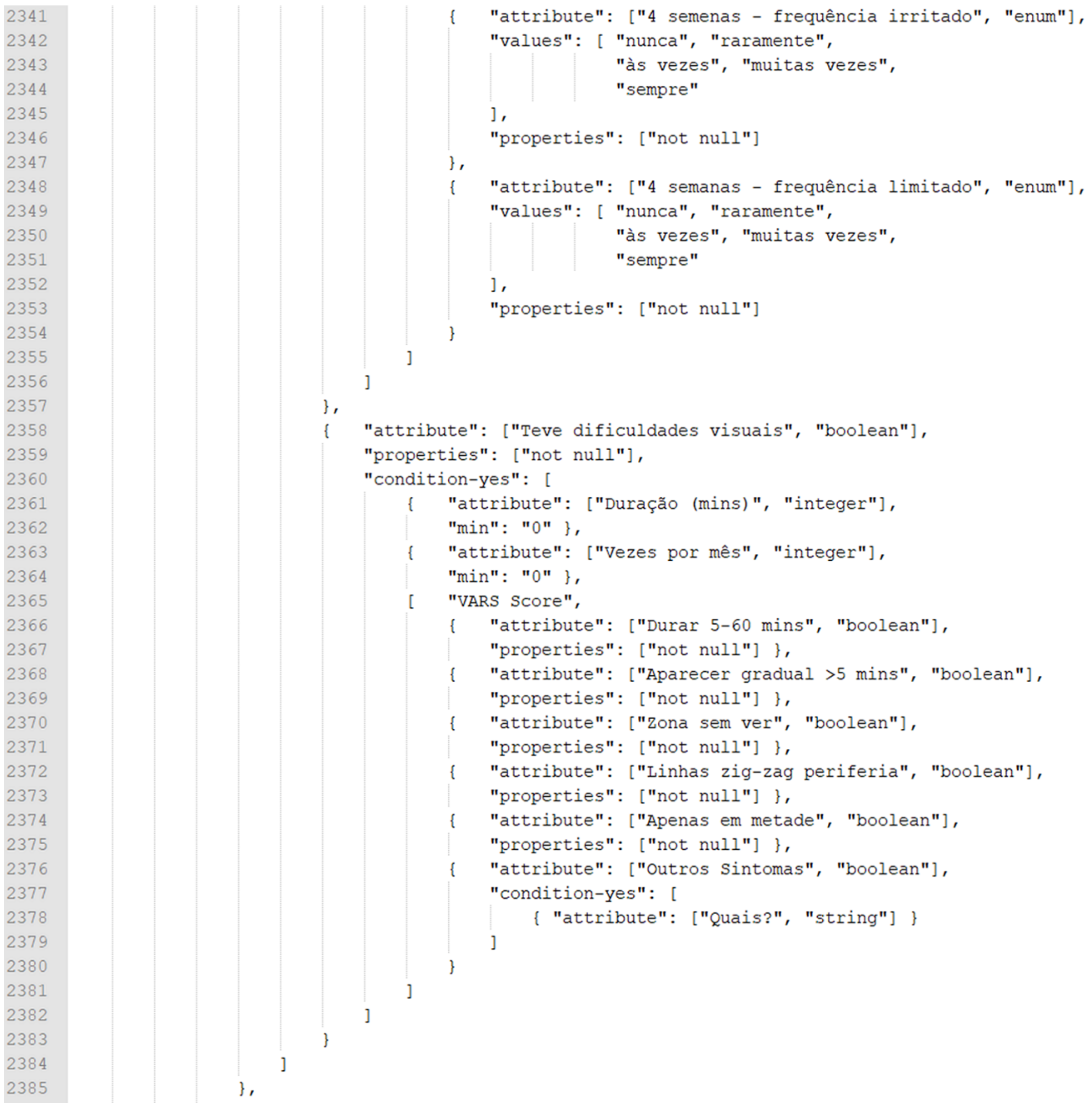
```
{   "attribute": ["Vezes no último mês", "integer"],
    "min": "0" },
{   "attribute": ["Dor duração ininterrupta", "enum"],
    "values": [
        {   "< 1 dia": [
                {   "attribute": ["Horas", "integer"],
                    "min": "0" }
            ]
        },
        {   "> 1 dia": [
                {   "attribute": ["Dias", "integer"],
                    "min": "0" }
            ]
        }
    ]
},
[   "Duração habitual",
    {   "attribute": ["Horas", "integer"],
        "min": "0" },
    {   "attribute": ["Minutos", "integer"],
        "min": "0" }
],
{   "attribute": ["Lado da dor", "list[enum]"],
    "values": ["Unilateal Direita", "Unilateal Esquerda", "Bilateral",
            "Variável"
    ]
},
{   "attribute": ["Local da dor", "list[enum]"],
    "values": ["Hemicraniana", "Holocraneana", "T. Trigemio V1",
        "T. Trigemio V2-3", "T. Occipital Alto", "T. Occipital Cervical",
        "Variável"
    ]
},
{   "attribute": ["Tipo de dor", "list[enum+]"],
    "values": ["Peso", "Aperto", "Pressão", "Pulsátil", "Guinadas",
        "Facadas", "Moinha", "Dor Fixa"
    ]
},
{   "attribute": ["Intensidade da dor (VAS)", "enum"],
    "properties": ["not null"],
    "values": ["0", "1", "2", "3", "4", "5", "6",
        "7", "8", "9", "10"
    ]
},
{   "attribute": ["Impacto", "enum"],
```

Figure C.51: PRECISE Stroke - JSON file specification 51/57.



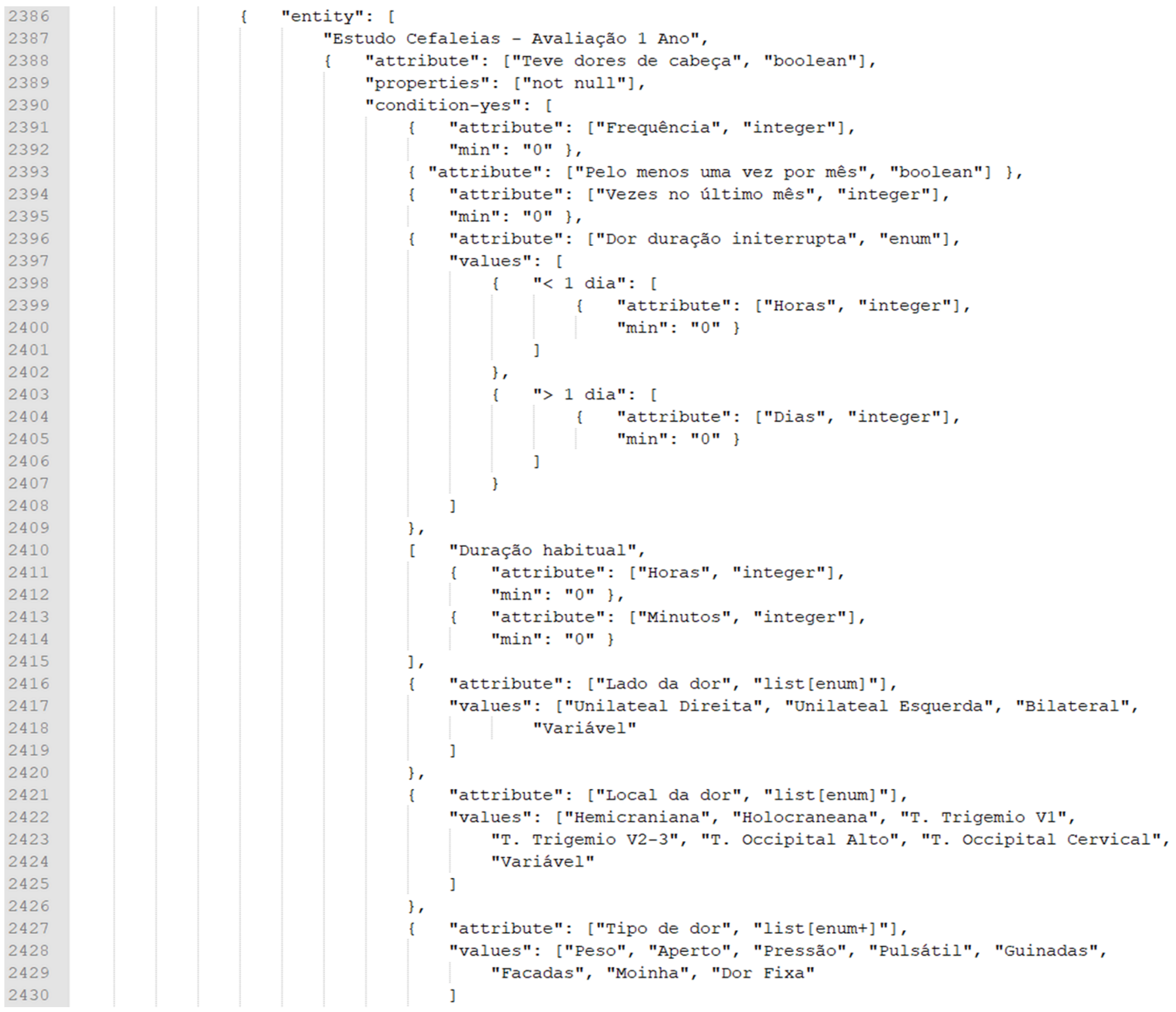
```
                                "properties": ["not null"],
                                "values": ["Ligeira", "Moderada", "Severa"]
                            },
                            {   "attribute": ["Dor acompanhada por", "list[enum+]"],
                                "values": ["Nauseas", "Vómitos", "Fotofobia", "Sonofobia",
                                    "Cinesiofobia", "Agravamento com esforço físico",
                                    "Agravamento com valsalva",
                                    "Sintomas autonómicos"
                                ]
                            },
                            {   "attribute": ["ID-MIGRAINE", "list[enum]"],
                                "values": [ "Nauseado/mal-disposto",
                                            "Luz incomodou",
                                            "Limitou capacidades" ]
                            },
                            { "attribute": ["Medicação", "string"] },
                            [   "HIT-6",
                                {   "attribute": ["Dor forte", "enum"],
                                    "values": [ "nunca", "raramente",
                                                "às vezes", "muitas vezes",
                                                "sempre"
                                    ],
                                    "properties": ["not null"]
                                },
                                {   "attribute": ["Limitação capacidades", "enum"],
                                    "values": [ "nunca", "raramente",
                                                "às vezes", "muitas vezes",
                                                "sempre"
                                    ],
                                    "properties": ["not null"]
                                },
                                {   "attribute": ["Gostaria deitar", "enum"],
                                    "values": [ "nunca", "raramente",
                                                "às vezes", "muitas vezes",
                                                "sempre"
                                    ],
                                    "properties": ["not null"]
                                },
                                {   "attribute": ["4 semanas - frequência cansado", "enum"],
                                    "values": [ "nunca", "raramente",
                                                "às vezes", "muitas vezes",
                                                "sempre"
                                    ],
                                    "properties": ["not null"]
                                },
```

Figure C.52: PRECISE Stroke - JSON file specification 52/57.



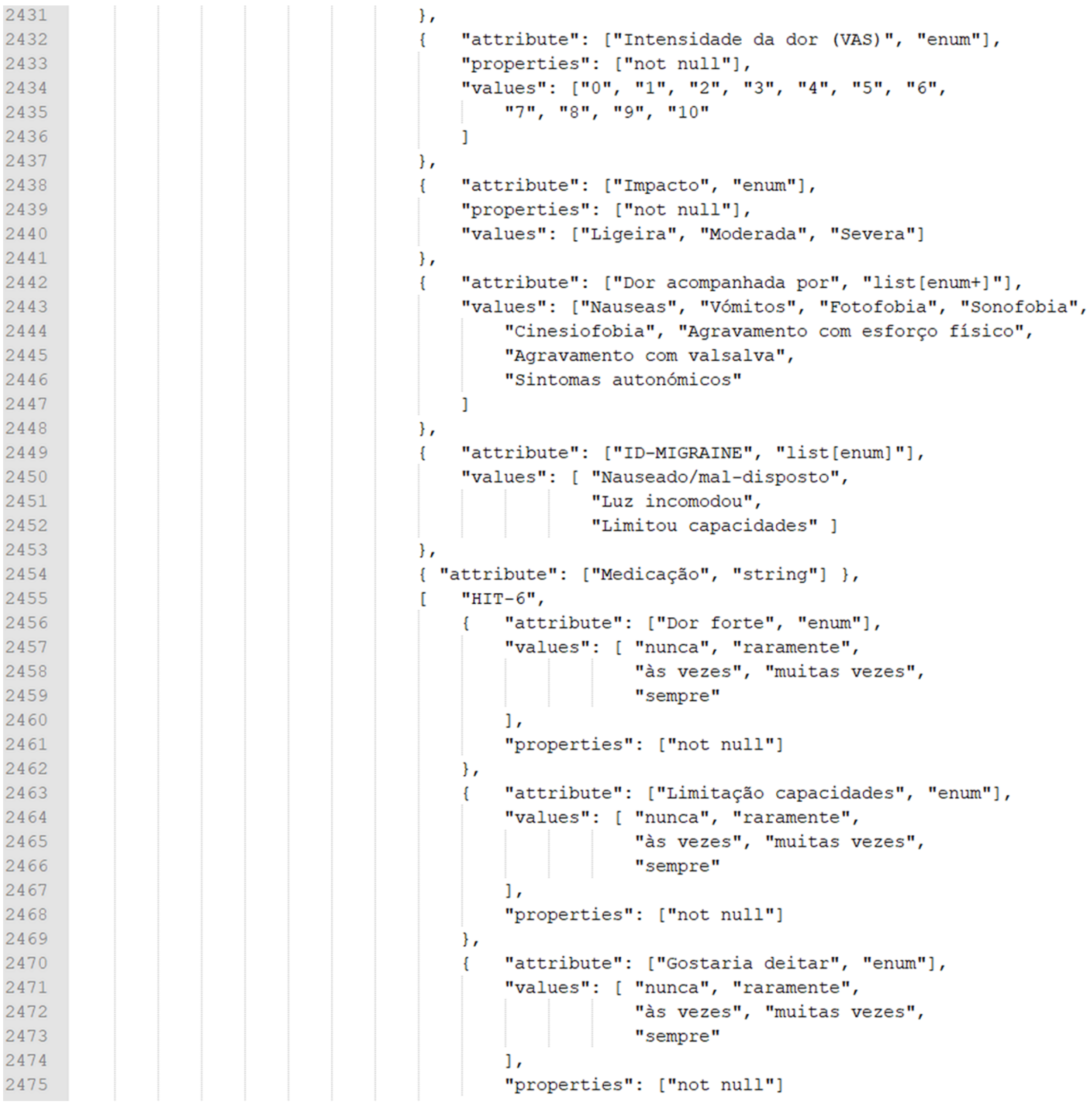
```
                                    {   "attribute": ["4 semenas - frequência irritado", "enum"],
                                        "values": [ "nunca", "raramente",
                                                    "às vezes", "muitas vezes",
                                                    "sempre"
                                        ],
                                        "properties": ["not null"]
                                    },
                                    {   "attribute": ["4 semanas - frequência limitado", "enum"],
                                        "values": [ "nunca", "raramente",
                                                    "às vezes", "muitas vezes",
                                                    "sempre"
                                        ],
                                        "properties": ["not null"]
                                    }
                                ]
                            ]
                        },
                        {   "attribute": ["Teve dificuldades visuais", "boolean"],
                            "properties": ["not null"],
                            "condition-yes": [
                                {   "attribute": ["Duração (mins)", "integer"],
                                    "min": "0" },
                                {   "attribute": ["Vezes por mês", "integer"],
                                    "min": "0" },
                                [   "VARS Score",
                                    {   "attribute": ["Durar 5-60 mins", "boolean"],
                                        "properties": ["not null"] },
                                    {   "attribute": ["Aparecer gradual >5 mins", "boolean"],
                                        "properties": ["not null"] },
                                    {   "attribute": ["Zona sem ver", "boolean"],
                                        "properties": ["not null"] },
                                    {   "attribute": ["Linhas zig-zag periferia", "boolean"],
                                        "properties": ["not null"] },
                                    {   "attribute": ["Apenas em metade", "boolean"],
                                        "properties": ["not null"] },
                                    {   "attribute": ["Outros Sintomas", "boolean"],
                                        "condition-yes": [
                                            { "attribute": ["Quais?", "string"] }
                                        ]
                                    }
                                ]
                            ]
                        }
                    ]
                },
```

Figure C.53: PRECISE Stroke - JSON file specification 53/57.



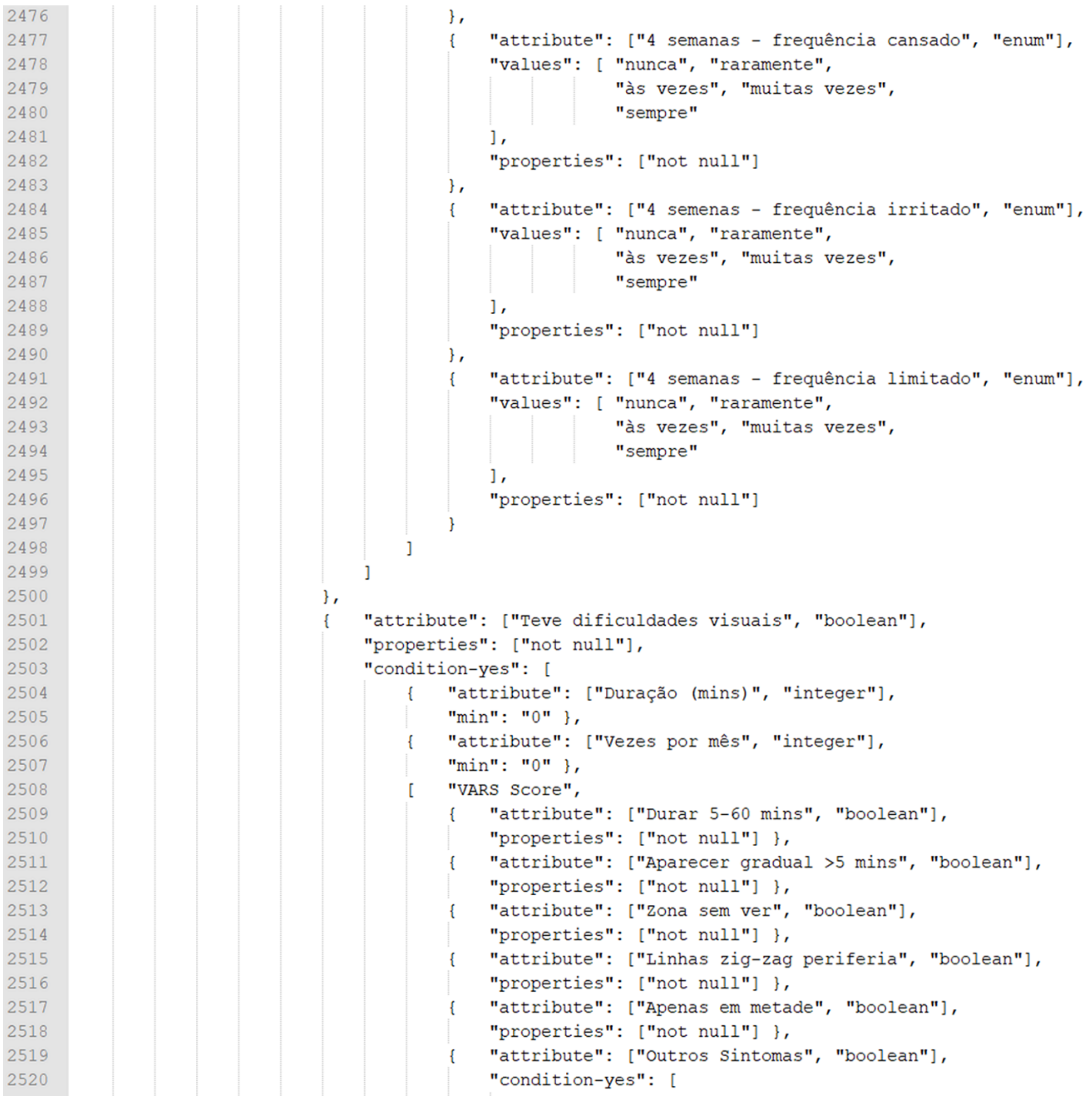
```
{   "entity": [
        "Estudo Cefaleias - Avaliação 1 Ano",
        {   "attribute": ["Teve dores de cabeça", "boolean"],
            "properties": ["not null"],
            "condition-yes": [
                {   "attribute": ["Frequência", "integer"],
                    "min": "0" },
                { "attribute": ["Pelo menos uma vez por mês", "boolean"] },
                {   "attribute": ["Vezes no último mês", "integer"],
                    "min": "0" },
                {   "attribute": ["Dor duração initerrupta", "enum"],
                    "values": [
                        {   "< 1 dia": [
                                {   "attribute": ["Horas", "integer"],
                                    "min": "0" }
                            ]
                        },
                        {   "> 1 dia": [
                                {   "attribute": ["Dias", "integer"],
                                    "min": "0" }
                            ]
                        }
                    ]
                },
                [   "Duração habitual",
                    {   "attribute": ["Horas", "integer"],
                        "min": "0" },
                    {   "attribute": ["Minutos", "integer"],
                        "min": "0" }
                ],
                {   "attribute": ["Lado da dor", "list[enum]"],
                    "values": ["Unilateal Direita", "Unilateal Esquerda", "Bilateral",
                            "Variável"
                    ]
                },
                {   "attribute": ["Local da dor", "list[enum]"],
                    "values": ["Hemicraniana", "Holocraneana", "T. Trigemio V1",
                        "T. Trigemio V2-3", "T. Occipital Alto", "T. Occipital Cervical",
                        "Variável"
                    ]
                },
                {   "attribute": ["Tipo de dor", "list[enum+]"],
                    "values": ["Peso", "Aperto", "Pressão", "Pulsátil", "Guinadas",
                        "Facadas", "Moinha", "Dor Fixa"
                    ]
```

Figure C.54: PRECISE Stroke - JSON file specification 54/57.



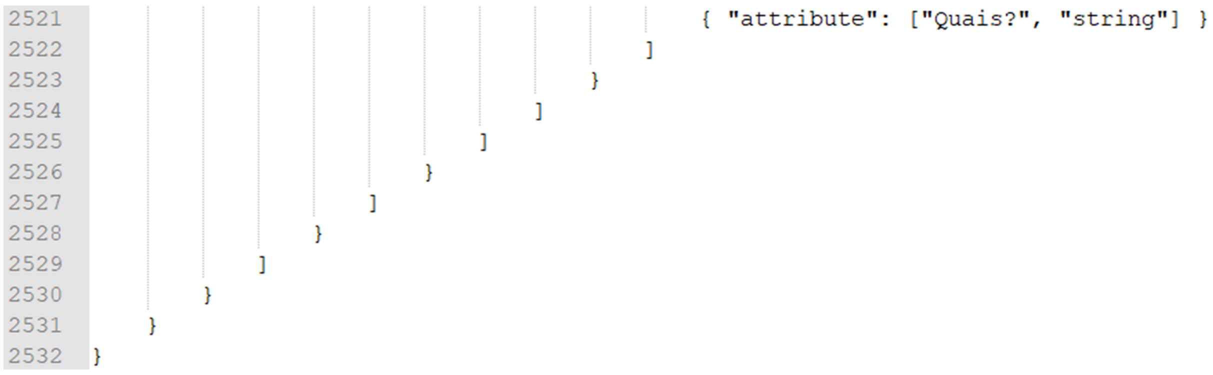
```
                                },
                                {   "attribute": ["Intensidade da dor (VAS)", "enum"],
                                    "properties": ["not null"],
                                    "values": ["0", "1", "2", "3", "4", "5", "6",
                                        "7", "8", "9", "10"
                                    ]
                                },
                                {   "attribute": ["Impacto", "enum"],
                                    "properties": ["not null"],
                                    "values": ["Ligeira", "Moderada", "Severa"]
                                },
                                {   "attribute": ["Dor acompanhada por", "list[enum+]"],
                                    "values": ["Nauseas", "Vómitos", "Fotofobia", "Sonofobia",
                                        "Cinesiofobia", "Agravamento com esforço físico",
                                        "Agravamento com valsalva",
                                        "Sintomas autonómicos"
                                    ]
                                },
                                {   "attribute": ["ID-MIGRAINE", "list[enum]"],
                                    "values": [ "Nauseado/mal-disposto",
                                                "Luz incomodou",
                                                "Limitou capacidades" ]
                                },
                                { "attribute": ["Medicação", "string"] },
                                [   "HIT-6",
                                    {   "attribute": ["Dor forte", "enum"],
                                        "values": [ "nunca", "raramente",
                                                    "às vezes", "muitas vezes",
                                                    "sempre"
                                        ],
                                        "properties": ["not null"]
                                    },
                                    {   "attribute": ["Limitação capacidades", "enum"],
                                        "values": [ "nunca", "raramente",
                                                    "às vezes", "muitas vezes",
                                                    "sempre"
                                        ],
                                        "properties": ["not null"]
                                    },
                                    {   "attribute": ["Gostaria deitar", "enum"],
                                        "values": [ "nunca", "raramente",
                                                    "às vezes", "muitas vezes",
                                                    "sempre"
                                        ],
                                        "properties": ["not null"]
```

Figure C.55: PRECISE Stroke - JSON file specification 55/57.



```
            },
            {   "attribute": ["4 semanas - frequência cansado", "enum"],
                "values": [ "nunca", "raramente",
                            "às vezes", "muitas vezes",
                            "sempre"
                ],
                "properties": ["not null"]
            },
            {   "attribute": ["4 semenas - frequência irritado", "enum"],
                "values": [ "nunca", "raramente",
                            "às vezes", "muitas vezes",
                            "sempre"
                ],
                "properties": ["not null"]
            },
            {   "attribute": ["4 semanas - frequência limitado", "enum"],
                "values": [ "nunca", "raramente",
                            "às vezes", "muitas vezes",
                            "sempre"
                ],
                "properties": ["not null"]
            }
        ]
    ]
},
{   "attribute": ["Teve dificuldades visuais", "boolean"],
    "properties": ["not null"],
    "condition-yes": [
        {   "attribute": ["Duração (mins)", "integer"],
            "min": "0" },
        {   "attribute": ["Vezes por mês", "integer"],
            "min": "0" },
        [   "VARS Score",
            {   "attribute": ["Durar 5-60 mins", "boolean"],
                "properties": ["not null"] },
            {   "attribute": ["Aparecer gradual >5 mins", "boolean"],
                "properties": ["not null"] },
            {   "attribute": ["Zona sem ver", "boolean"],
                "properties": ["not null"] },
            {   "attribute": ["Linhas zig-zag periferia", "boolean"],
                "properties": ["not null"] },
            {   "attribute": ["Apenas em metade", "boolean"],
                "properties": ["not null"] },
            {   "attribute": ["Outros Sintomas", "boolean"],
                "condition-yes": [
```

Figure C.56: PRECISE Stroke - JSON file specification 56/57.



```
                                            { "attribute": ["Quais?", "string"] }
                                        ]
                                    }
                                ]
                            ]
                        }
                    ]
                }
            ]
        }
    }
}
```

Figure C.57: PRECISE Stroke - JSON file specification 57/57.