qDocs: Citizen-Centered and Multi-Curator Document Automation Platform: The Curator Perspective

José António Trocado de Saldanha Sousa e Menezes

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisors: Prof. Doutor Alberto Manuel Rodrigues da Silva
Doutor João Paulo Pedro Mendes de Sousa Saraiva

Examination Committee

Chairperson: Prof. Mário Jorge Costa Gaspar da Silva
Supervisor: Prof. Doutor Alberto Manuel Rodrigues da Silva
Member of the Committee: Prof. João Carlos Ferreira

October 2019

Abstract

Document automation is a popular approach that supports the creation of electronic documents in a flexible and efficient way, which provides several benefits for organizations. Document automation platforms are popular in several areas and keep evolving over time. However, these platforms are most common in private organizations, therefore not being used with the purpose to help citizens in general. These systems allow the definition and management of (document) templates, which are extended versions of common documents with particular elements called fields, merge fields or form objects.

This dissertation introduces and discusses qDocs, a citizen-centered and multi-curator document automation platform, for managing dynamic electronic documents (e.g. id cards, forms, certificates) that are accessible to any citizen in a easy and secure way. qDocs provides a single point of access for citizens to create, use and manage their own documents. These documents are produced from templates curated by public or private organizations (named as curators) that also participate in this qDocs ecosystem. A curator creates its documents within the platform as templates that then become available for citizens belonging to it. qDocs allows templates such as identification documents, certificates, reports, forms and questionnaires. The design of these templates is defined in the scope of just one curator, but the orchestration of data and services can come from different curators.

qDocs is a platform that already had a first stage of development before this work, but it still lacked flexibility to fulfill specific requirements curators may have regarding documents, such as feature configuration or participant invitation. Also, before this work, every curator user could perform any action regarding its curator, with no access restrictions, like changing a document template. This dissertation provides a solution to help qDocs become available for every kind of curator, with management within the platform and with flexible configuration for curators that may have specific requirements for documents.

Keywords

Document Automation; Document; Template; Citizen-Centered; Curator; Citizen.

Resumo

A automação de documentos é uma abordagem que suporta a criação de documentos electrónicos de uma forma flexível e eficiente, o que oferece diversos benefícios para as organizações. As plataformas de automação de documentos são populares em diversas áreas e continuam a evoluir ao longo do tempo. No entanto, estas plataformas são mais comuns em organizações privadas, portanto, não são usadas com o objetivo de ajudar os cidadãos em geral. Estes sistemas permitem a definição e o gestão de modelos (de documentos), que são versões estendidas de documentos comuns com elementos específicos chamados campos, campos de fusão ou objetos de formulário.

Esta dissertação apresenta e discute o qDocs, uma plataforma de automação de documentos centrada no cidadão e multi-curador, para a gestão de documentos eletrónicos dinâmicos (por exemplo, cartões de identificação, formulários, certificados) que são acessíveis a qualquer cidadão de uma maneira fácil e segura. O qDocs fornece um único ponto de acesso para os cidadãos criarem, usarem e gerirem os seus próprios documentos. Esses documentos são produzidos a partir de modelos curados por organizações públicas ou privadas (nomeadas como curadores) que também participam no ecossistema do qDocs. Um curador cria os seus documentos dentro da plataforma como modelos que depois se tornam disponíveis para os cidadãos pertencentes a ele. O qDocs permite modelos como documentos de identificação, certificados, relatórios, formulários e questionários. O design desses modelos é definido no âmbito de um só curador, mas a orquestração de dados e serviços pode vir de diferentes curadores.

O qDocs é uma plataforma que já tinha um primeiro estado de desenvolvimento antes deste trabalho, mas ainda não tinha flexibilidade para atender aos requisitos específicos que os curadores podem ter em relação aos documentos, como a configuração de recursos ou o convite do participantes. Além disso, antes deste trabalho, cada utilizador do curador poderia realizar qualquer ação em relação ao seu curador, sem restrições de acesso, como alterar um modelo de documento. Esta dissertação fornece uma solução para ajudar o qDocs a a tornar-se disponível para todo tipo de curadores, com gerenciamento dentro da plataforma e com configuração flexível para curadores que podem ter requisitos específicos para documentos.

Palavras Chave

Automação de Documentos; Documento; Modelo; Centrado no Cidadão; Curador; Cidadão.

Contents

List of Figures

List of Tables

Acronyms

AJAX	Asynchronous Javascript and XML
API	Application Programming Interface
AWS	Amazon Web Services
BPMN	Business Process Modeling Notation
CMS	Content Management Systems
CRM	Customer Relationship Management
DBMS	Database Management System
DMS	Document Management System
DOM	Document Object Model
E-Commerce	Electronic Commerce
ERP	Enterprise Resource Planning
IoC	Inversion of Control
LHI	Law Help Interactive
MS-Excel	Microsoft-Excel
MS-Powerpoint	Microsoft-Powerpoint
MS-Word	Microsoft-Word
MVC	Model-View-Controller
ORM	Object-Relational Mapper
PC	Personal Computer
REST	Representational State Transfer
SPA	Single Page Application
UI	User Interface
UML	Unified Modeling Language
WS	Web Service

1 Introduction

Contents

Document automation is a popular approach that supports the creation of electronic documents in a flexible and efficient way. These software systems allow the definition and management of (document) templates, which are extended versions of common documents with particular elements called fields, merge fields, form objects, etc. Document automation systems allow to create and assembly new documents from these previously defined templates by replacing form objects with concrete data collected directly from end-users or from external data sources. Nowadays there are some document automation tools and platforms (e.g., ActiveDocs, SmartDocuments, HotDocs or xPressDocx), but these are mainly focused on document assembly for just one organization purpose, thus not accessible at all to the end-users and, in particular, to citizens when they intend to interact with such organizations, in the scope of administrative or bureaucratic processes [3].

This thesis introduces and discusses qDocs, a citizen-centered and multi-curator document automation platform for managing dynamic electronic documents that are accessible in a easy but secure way to any citizen through any device [4]. It has been developed and promoted by MDSS[1] with support and research developed by the Information and Decision Support System Group of INESC-ID[2].

The main purpose of qDocs is to provide citizens a single point of accessing and managing their own documents, like identification documents, forms, certificates, reports and questionnaires. These documents are provided by curators, which are private or public organizations that are connected to the qDocs platform. A curator creates its documents within the platform as templates that then become available for citizens belonging to it. qDocs is a collaborative platform that facilitates and promotes the relationship between citizens and curators through any process that involves the request, publication, access, delivery and sharing of electronic documents. The design of templates is defined in the scope of just one curator, but the orchestration of data services can come from different curators.

qDocs allows the search for documents based on several criteria such as kind of life events (e.g. birth, health or education), document type (e.g. certificates, forms or identification documents) or curator (e.g. Tax Services, University or Public Services).

Figure 1.1 presents the general architecture for the qDocs platform; which suggests the integration of four key applications: qDocs/Citizen, qDocs/Curator, qDocs/Admin, and qBox. The qDocs ecosystem involves the interaction of the qDocs platform itself, the citizens and the curators. Citizens interact through the qDocs/Citizen application. Curators create, define and configure templates through the qDocs/Curator app and then what delivers the data to the respective citizens is the qBox platform, which is the platform that integrates with all the curators' databases and applications. The qDocs/Admin application is used to manage and configure he general aspects of the qDocs platform.

[1]https://mdss.pt/index.php, last accessed 30th July 2019
[2]https://www.inesc-id.pt/, last accessed 30th July 2019

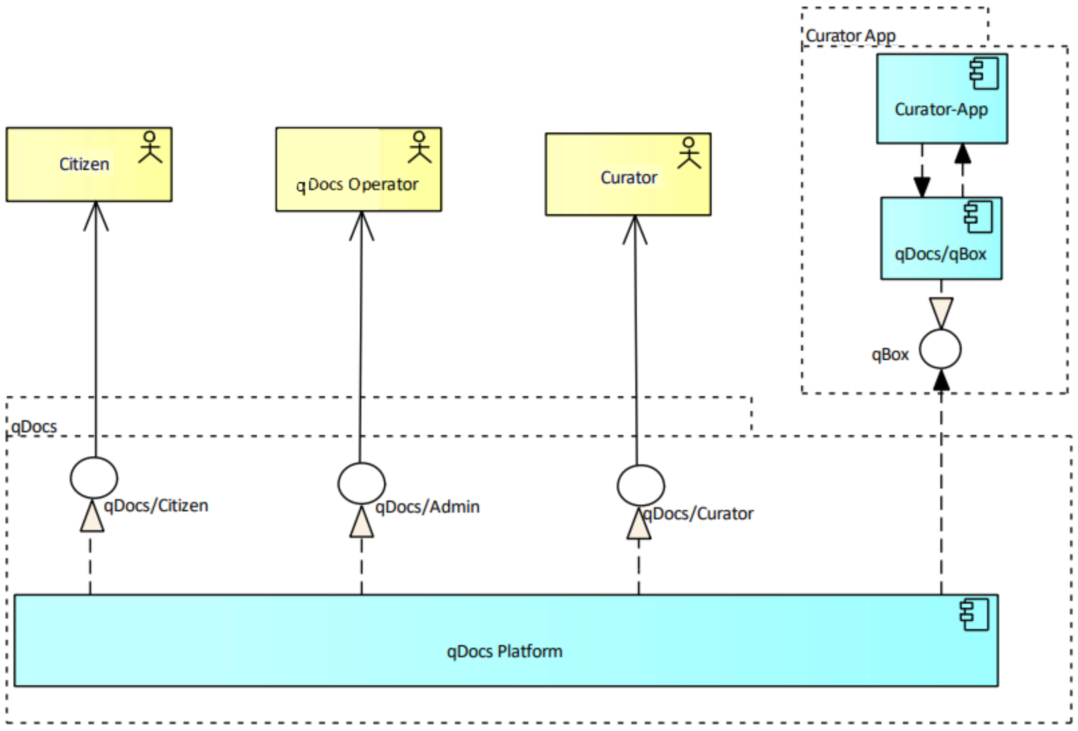


Figure 1.1: qDocs General Model (ArchiMate diagram)

Regarding data storage, data stored in one or more curators is dynamically merged in documents generated in real time. The fact that data storage does not leave its respective curator's data-centers, allows curators full control over their information. Document materialization only happens in the citizen's device, increasing privacy and information security. Only the citizen can manage and has full access over his own personal information.

For a citizen to have access to a document, the following happens in the qDocs ecosystem: first, the citizen requests a document to the qDocs platform; then the qDocs platform sends a "request acknowledged" message to qBox; after that, qBox sends a key to qDocs; then qDocs sends that key with the document template to the citizen's device; then the citizen's device sends the key directly to qBox; then qBox sends the respective data to the citizen and the document is generated in the citizen's device; Finally, qBox saves the data from the citizen so he/she can access the document anytime from that moment on.

1.1 Problem Statement

qDocs is an existing document automation platform that is still in an early stage of development so it only provides basic features for curators and citizens, like template creation for curators and document access for citizens.

Since qDocs is supposed to be used by any kind of curator, it must support the creation of templates with different features and requirements, i.e. qDocs must support participant invitation for documents that require interaction from different citizens, and also features such as payment or strong authentication in order to access documents like certificates or forms. Also curator users are unable to configure settings for their own curator and every curator as access to every functionality within its curator, without any access restrictions.

Citizens in the qDocs platform must be able to interact with their documents according to what was defined by the curator that owns the document template, i.e. a citizen has to pay for a certificate before accessing it if the curator configures the certificate's template in that manner. Also, citizens must be able to invite participants to interact with their document or vice-versa. For example, the jury of a dissertation defense meeting must sign the respective minute of the meeting although the document does not belong to any of the jury's members.

Therefore, this dissertation aims to provide curators management and configuration capabilities within their curator, as well as flexibility regarding template configuration. The same goes for citizens, that must be able to interact with their documents according to the configuration made by the respective curators, including.

1.2 Objectives

After defining the problem, the objectives of this work are the following:

- Add roles and respective functionality to provide different levels of responsibility in the scope of a curator — curator users do not have any access restrictions to functionalities within a curator, so by creating roles this can be achieved; also with the implementation of new functionalities they must attributed to specific roles;
- Add feature configuration to templates and test them in the citizen application — for example, a curator must be able to configure if a document can be exported or not, and this implementation must be tested in the qDocs/Citizen application;
- Add participant configuration to templates so citizens can invite participants to interact with their documents and vice-versa, for example a certificate that needs to be signed — this also must be implemented in the qDocs/Curator application and be tested in the qDocs/Citizen application;

- Add settings configuration to the qDocs/Curator and to the qDocs/Admin application — this provides more flexibility to the qDocs platform and to curators, since they can configure their own settings;
- Enhance User Interface (UI) to improve user experience — throughout this work, improving the usability experience for every kind of user must taken into account, for example improving navigation between interfaces;

1.3 Organization of the Document

The remainder of this dissertation is organized as follows: **Chapter 2** presents the related work, which is about document automation platforms; **Chapter 3** describes the technologies that are going to be used in the development of the this work; **Chapter 4** describes the functional requirements of the qDocs platform; **Chapter 5** presents a technical description of what was done with this work; **Chapter 6** presents two scenarios used for the evaluation of this work **Chapter 7** discusses the main conclusion achieved with this work along with suggestions for future work.

2 Related Work

Contents

Document automation provides several benefits for organizations such as improved efficiency in document production (i.e. documents take less time to be produced), reduced human errors (i.e., end-users only have to provide the necessary information rather than producing a document from scratch), costs (producing multiple documents takes less time, so costs are reduced) and turn-around times for clients (a document request can be fulfilled faster if a template already exists) [5–7].

This chapter performs a literature review to analyze the state-of-art of document automation platforms, in what areas they are used and which ones are citizen-centered, since it is one key characteristic of this research. In section 2.1 it introduces document automation platforms, then in section 2.2 presents the evaluation framework used to assess popular platforms that are described in section 2.3. Finally, section 2.4 compares and discusses the results obtained from the analysis.

2.1 Introduction

This research has been influenced by the relevance and impact of document automation platforms in the efficiency of businesses, but also on their lack of being citizen-centered.

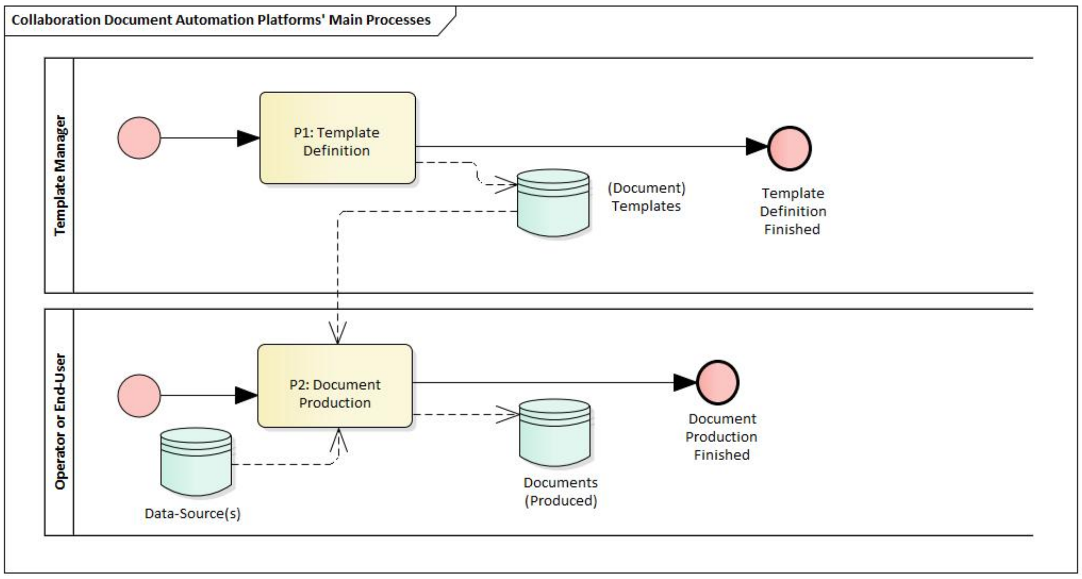


Figure 2.1: Document Automation Platforms' Main Processes (BPMN diagram)

Figure 2.1, represented in a Business Process Modeling Notation (BPMN) collaboration diagram, depicts the common processes and artifacts involved in the context of document automation platforms. First (P1), a template manager is responsible for the definition and design of document templates, which are stored in a persistent data storage for document templates. Second (P2), an operator or end-user uses the document templates together with information obtained from a data-source to assembly or produce specific documents, that can be stored in persistent data storage. Document automation (or document assembly) software tools intend to replace the manual filling of form-based documents with templates by allowing, for example, users answer software-driven interview questions. The information collected from end-users or external data sources can be used with such document templates to produce another set of concrete documents [8–10].

Document automation platforms are becoming popular in several areas, such as law [11], public administration [12, 13], e-business [14] or even the aerospace industry [15]. Document automation is particularly popular in the business and legal services areas, since it is where a great number of repetitive documents (e.g. contracts, surveys, minutes) are produced [16–18]. For example, "Do-It-Yourself (DIY) law" is becoming a reality and is improved due to these document automation tools, since it became easier for self-representing litigants to produce documents they need [19]; for example, A2J Author[1] is a platform designed for this purpose [20]. Regarding e-government, document automation is also changing the paradigm from "one-size-fits-all" to "one user, one document" [11], enabling the generation of personalized documents in domains with high variability, having a template that can be reusable and slightly changed to different organizations [21].

Document automation tools keep evolving, for example, they are using machine learning techniques to validate and automatically create contracts [22], can provide visual information, such as charts and tables, based on the end-user or external data source information [23] and document templates are easier to change or upgrade [24].

2.2 Evaluation Framework

To better analyze and compare document automation platforms an evaluation framework is defined. This framework involves the following dimensions enumerated as illustrated in Table 2.1: (a) Document Template Definition, (b) Document Generation, (c) Content Management, (d) Integration, (e) Platform Availability, (f) Digital Signature, (g) Citizen-Centered, (h) Other Features. These dimensions provide information about the architectural details of the platforms described.

The evaluation is based on the available information presented in each of the platforms' websites (demos included). A ranking system was not defined because the objective of this evaluation is to understand how document automation is addressed in the industry and what aspects should be considered in the development of the qDocs platform.

The citizen-centered dimension is the only one that has a subjective measurement, because there is no information about it but still is considered important since one objective of the qDocs platform is to be citizen-centered.

[1] https://www.a2jauthor.org/, last accessed 30th July 2019

Table 2.1: Evaluation Framework Overview

Dimension		Measurment Range	Observations
Document Template Definition	Used Formats	Set of file formats (e.g. MS-Word)	Measure can be a combination of one or more file formats
	Elements Supported	Set of elements (e.g. Repeating Items)	Measure can be a combination of one or more elements that are present in Table 2.2
	Extended Elements Supported	Set of elements (e.g. Link to External Data Sources)	Measure can be a combination of one or more elements
Document Generation	Output Formats	Set of file formats (e.g. PDF)	Measure can be a combination of one or more file formats
	Input Data-sets	Set of data-sets (e.g. Databases, Excel)	Measure can be a combination of one or more data-sets, can be summarized as all of the platforms' integration solutions
	Input Data-sets Formats	Set of file formats (e.g. XML)	Measure can be a combination of one or more file formats
Content Management	Templates	Internal, External	Measure can be a combination of one or more values
	Generated Documents	Internal, External	Measure can be a combination of one or more values
Integration	Deployable as a WS API	Yes, No	Values are mutually exclusive
	DBMS	Yes, No	Values are mutually exclusive
	WSs and/or APIs	Yes, No	Values are mutually exclusive
Platform Availability		Server, Cloud, Hybrid, Desktop	Measure can be a combination of one or more values
Digital/Electronic Signature		Yes, No	Values are mutually exclusive
Citizen Centric		Yes, No	Values are mutually exclusive
Other Features		Set of non-specific features	Measure can be a combination of features that are not measured in the previous dimensions

2.2.1 Document Template Definition

What makes document automation so useful is the ability to have a document template with a set of predefined elements so it is possible to produce the same type of document multiple times, making template definition a key feature for this type of platforms.

This dimension is divided into three sub-dimensions: (a) *Used Formats*, which determines the template file formats; (b) *Elements Supported*, which determines the elements that are supported by the templates, e.g. plain text, selection lists, text variables, conditional blocks and repeating items. These elements are described in Table 2.2; (c) *Extended Elements Supported*, which determines more complex elements that are supported by the templates, e.g. links to external data sources, which is an element that makes it possible to retrieve data from an external source; dynamic tables, charts and graphs, which are elements that make it possible to generate tables, charts and graphs based on user input; and variable sharing between document templates, which is an element that makes it possible to store a variable to be used in several document templates. In the qDocs context, these elements supported are called *FormObjects* when used individually and are called *FormBlocks* when grouped. The value for the *Used Formats* dimension is the set of possible formats that a template can have, being Microsoft-Word (MS-Word) file formats the most common. The value for the *Elements Supported* dimension is the set of elements that are supported by the platform and are present in Table 2.2. The value *All* is used if all the elements from Table 2.2 are supported by the platform. The value for *Extended Elements Supported* dimension is the set of more complex elements that are supported by the platform.

Table 2.2: Common Elements Supported

Name	Description
Plain Text	Input from the user with no restrictions
Selection Lists	List of options for the user to select
Repeating Items	Information that is repeated throughout the document
Text Variables	Information that is needed more than once in the document so it is stored in a variable. The variable can be populated through user input or automatically (e.g. date)
Conditional Blocks	Information that is added to the document only if some conditions are met

2.2.2 Document Generation

After having document templates available, a user can proceed to document generation, being this another key feature for these platforms.

This dimension is divided into three sub-dimensions: (a) *Output Formats*, which determines the file formats for the generated document; (b) *Input Data-Sets*, which determines the type of data-sets that the platform can read from, e.g. databases or Microsoft-Excel (MS-Excel) files; (c) *Input Data-Sets Formats*, which determines the file formats that the document template can read in order to retrieve the necessary data to produce the document, e.g. XML. The value for the *Input Data-Sets* dimension is the set of possible data-sets that a document that is going to be generated can read from and the values for the *Output Formats* and *Input Data-Sets Formats* dimensions are the set of file formats that the data-sets (input) and the generated document (output) can have.

2.2.3 Content Management

Document automation platforms need to provide a way of managing the created templates and generated documents.

This dimension is divided into two sub-dimensions: (a) *Templates*, which determines the way document templates are managed; (b) *Generated Documents*, which determines the way generated documents are managed. Both dimensions can have one or more values from their measurement range, that are the same: (a) *Internal*, if that management is supported using the platform itself or the local file system; (b) *External*, if the management is supported by other software (e.g. an external file system, Content Management Systems (CMS) or Document Management System (DMS)) rather than using the platform.

2.2.4 Integration

Integration makes interaction between different systems possible [25], promoting interoperability [26].

This dimension is divided into three sub-dimensions: (a) *Deployable as a Web Service (WS) Application Programming Interface (API)*, which determines if the platform can be available through a WS API or not, i.e. if the services it offers can be provided as WSs, meaning it is easier to integrate with other applications; (b) *Database Management Systems (DBMSs)*, which determines if the platform can integrate with DBMSs; (c) *WSs and/or APIs*, which determines if the platform can integrate with other WSs and/or APIs, i.e. if it can integrate with existing platforms. All sub-dimensions only allow a single value from their measurement range, which is the same: (a) yes, if the platform provides a WS API and/or can integrate with DBMSs, WSs and/or APIs; (b) no, if the platform does not provide a WS API and/or can not integrate with DBMSs, WSs and/or APIs.

2.2.5 Platform Availability

Nowadays organizations and users can rely on the Cloud [27–29] to run an application, and not only on dedicated servers, so knowing how a platform makes its services available is an important concern.

This dimension measures how the platform is available and allows the usage of one or more values from its measurement range which can be: (a) *Server*, if the platform is available through installation in a dedicated server; (b) *Cloud*, if the platform is available as an API in the cloud; (c) *Hybrid*, if the platform is available through a mix between the *Server* and *Cloud* values; (d) *Desktop*, if it is available as a Microsoft Office add-in and/or if it has its own API for installation in a Personal Computer (PC).

2.2.6 Digital Signature

This dimension measures whether a platform supports digital signatures or not, i.e. if documents can be signed through the platform before or after being generated (e.g. a contract). It only allows a single value from its measurement range: (a) yes, if the platform supports digital signatures; (b) no, if the platform does not support digital signatures.

2.2.7 Citizen-Centered

This dimension measures if the platform is focused on the citizen or not, i.e. if the platform's purpose is to be used by the citizens therefore being for public use or if the platform was developed to be sold as a solution for an organization. It only allows a single value from its measurement range: (a) yes, if the platform is citizen-centered; (b) no, if the platform is not citizen-centered.

2.2.8 Other Features

This dimension measures features of a platform that do not fit in the other dimensions. The value for this dimension is a set of non-specific features, such as *HotDocs Market*, which is an Electronic Commerce (E-Commerce) platform for document templates; or such as signature notification, which allows the owner of a generated document to be notified when a user digitally signs that document.

2.3 Document Automation Platforms

The need for document automation is really present in many areas such as e-business [14], legal services [8, 18], public administration [12, 13] or even in the aerospace industry [15], so the emergence of these platforms came along with concrete examples like: (a) ActiveDocs Opus, (b) Smart Documents, (c) HotDocs, (d) A2J Author, (e) xPressDocx, (f) Templafy, (g) Clio and (h) SmartDocs - Acculynx.

These platforms share a common set of features and processes such as: (a) document template definition; (b) document generation; and (c) content management. Their differences are in how they provide such features. These platforms provide their services via Cloud, Server, APIs, or even Microsoft Office add-ins. Regarding template creation, some use MS-Word as document template format while others use PDF as well. They support different elements such as plain text, selection lists, text variables, repeating items and conditional blocks. Regarding document generation, some platforms may receive data from different sources as input. The generated documents can be produced in many formats, e.g. PDF, HTML, XML, ODT or TIFF. Most of these platforms provide demos in their websites and offer several integration possibilities. Some platforms offer extra features such as digital signatures and customized workflow for documents.

ActiveDocs Opus ActiveDocs Opus by ActiveDocs[2] is available via Server or Cloud, regardless of platform (Amazon Web Services (AWS), Microsoft Azure, etc.). It supports more complex elements such as: (a) links to external data sources and (b) dynamic tables, charts and graphs. It receives input data-sets as XML from DBMSs and APIs, and the generated documents can have several formats such as PDF, DOC and HTML. It integrates with several DBMSs and APIs: (a) Microsoft products, (b) SAP, (c) Oracle, (d) DMSs and (e) legacy applications. The platform is not designed to be citizen-centered.

Smart Documents Smart Documents[3] offers services that can be integrated with existing Customer Relationship Management (CRM) or Enterprise Resource Planning (ERP) systems. Documents that are going to be generated receive input data-sets as a XML file from DBMSs. It is deployable as a WS API. The platform has a Data Retrieval Module to integrate with DBMSs and integrates with other WSs and APIs. It is not designed to be citizen-centered.

HotDocs HotDocs[4] is a document assembly software that is available via Server, Cloud or as mix in-between (Hybrid). HotDocs *Developer* is HotDocs' module for template creation. The used formats for templates are MS-Word file formats or PDF, that works inside de platforms' template editor. It supports a more complex element that is variable sharing between documents, which lets a text variable that was

[2]https://www.activedocs.com/, last accessed 30th July 2019
[3]https://smartdocuments.eu/en/, last accessed 30th July 2019
[4]https://www.hotdocs.com/, last accessed 30th July 2019

created in a document template to be reused in another document template. HotDocs *User* is its module for document generation, storage and management. Generated documents can be MS-Word or PDF files. Besides the generated document, a file with the information provided to fill the document is created as a XML file so it can be used as input data-set for future use. The platform is deployable as a WS API, being able to integrate with DBMSs, WSs and APIs. It is not designed to be citizen-centered. Has an extra feature called HotDocs *Market*: an E-Commerce platform that enables the distribution of content subscription to legal professionals.

A2J Author A2J (Access to Justice) Author[5] is a document assembly platform that is available via Cloud. After the template is created, it is uploaded to the Law Help Interactive (LHI) server, which provides internet-based document assembly services powered by the HotDocs platform. A2J Guided Interview is the software's component for document generation. All the data used to produce the document is given by the user and is sent to the LHI server as a XML file. The generated document is a PDF file. A2J Author is designed to be citizen-centered because the main purpose of this platform is to deliver greater access to justice for self-represented litigants by enabling non-technical authors from the courts, clerk's offices, legal services organizations, and law schools to create interactive templates [20]. The software is free to interested court, legal services organizations, and other non-profits for non-commercial use and has a guide for creating templates and a document generation demo available.

xPressDox xPressDox[6] is a document automation platform that provides its services through three different channels: (a) Desktop, (b) Servers and (c) Cloud (APIs). xPressDox *Desktop* works as a MS-Word add-in to create and run templates. It includes different versions (*Supervisor*, *Author* and *Runner*) that have different permissions regarding a template. Documents that are going to be produced receive input from DBMSs. The generated document is a MS-Word or PDF file. xPressDox is deployable as a Representational State Transfer (REST), .NET or COM API. xPressDox *Servers* integrate with DBMSs, WSs and APIs. They are available in three different configurations (*Windows Authentication*, *API* and *Cloud Integration*) and can be own hosted or not. The platform is not designed to be citizen-centered.

[5]https://www.a2jauthor.org/, last accessed 30th July 2019
[6]http://xpressdox.com/, last accessed 30th July 2019

Templafy Templafy[7] enables its main features via Microsoft *Office* add-ins. Regarding document template definition, the templates defined have Microsoft *Office* file formats: (a) MS-Word; and (b) Microsoft-Powerpoint (MS-Powerpoint). Regarding content management, templates are managed through an external web interface and generated documents are managed through an external web interface or through a CMS. It is not designed to be citizen-centered. It also has a feature called *BrandChecker* that is used to check for non-compliant documents within a company, i.e. documents within an organization have a tag that if it is not used, it is detected by the platform.

Clio Clio[8] is a document automation platform similar to Templafy, so it is not going to be considered for discussion in section 2.4. It also enables its features main features via a MS-Word add-in. Regarding all of the evaluation framework's dimensions, it presents the same values as Templafy, except for the other features dimension, because it is not applicable.

SmartDocs - Acculynx SmartDocs[9] by Acculynx[10] offers its services through a software platform that is embedded in a CRM platform (Acculynx). Documents can be uploaded in order to be transformed into templates and they can be managed through the cloud. Digital signatures, legally binding contracts and signature notification are key features of the software. Digital signatures make it possible for a user to sign a document online on any device. Legally binding contracts are logs for each user's digital signature that include its name, email address, date, time and IP address. Signature notification allows a document's owner to be notified when a document is signed with a digital signature. The platform is not designed to be citizen-centered.

2.4 Comparison and Discussion

The results of applying the evaluation framework to the platforms described are shown in Tables 2.3, 2.4 and 2.5. Despite not having information from all the evaluated platforms regarding all dimensions, it is possible to understand common features regarding document automation platforms.

Regarding **document template definition**, MS-Word file formats are the most used. Using these file formats gives flexibility to new users, because they can import templates that were already created with MS-Word into the platform and then edit them. HotDocs additionally uses PDF files and A2J Author does not have a specific used format. There is few information about elements (and extended elements). However, plain text, selection lists, repeating items, text variables and conditional blocks are common elements in all platforms. These elements are all described in Table 2.2 as presented in subsection

[7] https://www.templafy.com/, last accessed 30th July 2019
[8] https://www.clio.com/eu/features/legal-documents/, last accessed 30th July 2019
[9] http://www.acculynx.com/smartdocs, last accessed 30th July 2019
[10] http://www.acculynx.com/, last accessed 30th July 2019

2.2. HotDocs additionally has variable sharing between documents, which lets a text variable that was created in a document template to be reused in another document template. This is an interesting feature to speed up variable creation in templates that require the same type of information, e.g. a variable that stores a name or a license number. ActiveDocs Opus supports dynamic tables, charts and graphs, which is an interesting feature for all document automation platforms. For a citizen-centered platform like qDocs, this feature would give more flexibility for curators to produce templates and for non citizen-centered platforms it is useful to create detailed reports, for example.

Regarding **document generation**, PDF is the most used output file format and MS-Word file formats are common too. HotDocs and ActiveDocs Opus also support XML. There is few information about input data-sets and their formats. All platforms can gather data directly from DBMSs. ActiveDocs Opus can also gather data from APIs. This data is read as a XML file for all platforms. Besides XML, JSON could be used too, since it is a standard format for most applications.

Table 2.3: Document Template Definition and Document Generation Table

	Document Template Definition			**Document Generation**		
Platforms	**Used Formats**	**Elements Supported**	**Extended Elements Supported**	**Output Formats**	**Input Data -Sets**	**Input Data -Sets Formats**
ActiveDocs	MS-Word	All	Link to External Data Sources; Dynamic Tables, Charts and Graphs	MS-Word; PDF; XPS; ODT; RTF; TIFF; XML; HTML	DBMSs; APIs	XML
Smart Documents	MS-Word	All	NA	PDF	DBMSs	XML
HotDocs	MS-Word; PDF	All	Variable Sharing between Documents	MS-Word; PDF; XML	DBMSs	XML
A2J Author	NA	All	NA	PDF	NA	XML
xPressDox	MS-Word	All	NA	MS-Word; PDF	DBMSs	NA
Templafy	MS-Word; MS-Powerpoint	All	NA	NA	NA	NA
SmartDocs – Acculynx	NA	NA	NA	NA	NA	NA

Regarding **content management**, most of the evaluated platforms support this dimension by doing content management within the platform. Only Templafy needs the support of an external web interface and/or CMS because it offers its service via a Microsoft add-in. ActiveDocs Opus can manage its generated documents through a DMS instead of doing it through the platform.

Regarding **integration**, not all platforms are deployable as WS APIs, but most of them can integrate with DBMSs, WSs and APIs. Smart Documents, HotDocs and xPressDocx offer all types of integration. On the contrary, SmartDocs - Acculynx is the only platform that does not offer any type of integration.

Regarding **platform availability**, most of the platforms offer their services via Server, i.e. they are installed in a dedicated server. Some also offer their service via Cloud API. HotDocs also supports a mix between Server and Cloud. Templafy only provides its services via a Microsoft Office add-in, which is also possible in the xPressDocx platform. A2J Author only provides its services via Cloud.

Table 2.4: Content Management and Integration Table

	Content Management		**Integration**		
Platforms	**Templates**	**Generated Documents**	**Deployable as a WS API**	**DBMSs**	**WSs and/or APIs**
ActiveDocs	Internal	Platform; External (Integrated with a DMS)	No	Yes	Yes
Smart Documents	Internal	Internal	Yes	Yes	Yes
HotDocs	Internal (Developer Module)	Internal (User Module)	Yes	Yes (since available through WS APIs)	Yes (since available through WS APIs)
A2J Author	NA	NA	NA	NA	NA
xPressDox	Internal	Internal	Yes	Yes (Server or Cloud)	Yes (Server or Cloud)
Templafy	External (Web Interface)	External (Web Interface or integrated with a CMS	No	Yes	Yes
SmartDocs – Acculynx	Internal	NA	No	No	No

Regarding the **digital signature dimension**, only ActiveDocs Opus and SmartDocs - Acculynx support digital signatures.

The **other features dimension** presents features that may be relevant to implement in the qDocs platform. Only A2J Author was designed to be citizen-centered, because its objective is to help self-represented litigants. Most platforms are not designed to be citizen-centered, i.e. they were developed as a solution for organizations, not as way to help citizens in processes that involve document production.

Table 2.5: Solo Dimensions Table

Platforms	Platform Availability	Digital Signature	Citizen Centered	Other Features
ActiveDocs	Server; Cloud	Yes	No	Integrates with Microsoft products
Smart Documents	Server	No	No	NA
HotDocs	Server; Cloud; Hybrid; Desktop	No	No	HotDocs Market (E-commerce platform)
A2J Author	Cloud	No	Yes	LHI server that provides services powered by HotDocs software
xPressDox	Server; Cloud; Desktop	No	No	Available through .NET and COM APIs
Templafy	Desktop	No	No	Can check for non-compliant documents within a company
SmartDocs – Acculynx	Server; Cloud	Yes	No	Legally binding contracts, Signature Notification

Although only one of the evaluated platforms was designed to be citizen-centered, there are some aspects to take into account for platforms that are like qDocs. The process of document generation adopted should be the same for both citizen-centered and not citizen-centered platforms, since it is a simple, user-friendly method. Regarding availability, cloud is the best option, because all services can be available to users in a single point of access (this method is already adopted by A2J Author). Digital signatures should be supported to make these platforms more flexible. One ”other feature” that may be useful for a citizen-centered platform is a workflow trigger for a document, i.e. a user generates a document and after it is generated, it is automatically sent to another party connected to the platform (in qDocs case, the document would be sent to certain curators and citizens, for example). These suggestions are relative to the user (citizen) side of a platform to-be developed.

On the other side, a citizen-centered platform like qDocs needs to interact with several entities to gather information. These type of platforms need to be able to integrate with legacy applications, DBMSs, APIs and WSs. Regarding document template definition, these platforms should be able to import a vast number of file formats, focusing on MS-Word file formats since they are the most common. For elements supported, the ones that were common between the evaluated platforms should be adopted. Adding the variable sharing feature would be useful too, as stated before. Content management should be done directly in the platform, since its availability would be via an API in the cloud.

3 Technology

Contents

qDocs is a Single Page Application (SPA) and uses two frameworks that integrate with one another.

This chapter describes the technologies that were used to develop the qDocs platform. Section 3.1 presents the ASP.NET Core framework, section 3.2 presents the Angular framework and Section 3.3 describes how SPAs work.

These frameworks use the following software programming languages: C#, Typescript (which is compiled to Javascript), HTML and CSS. These frameworks were learned through an online course available at Udemy[1].

3.1 ASP.NET Core

ASP.NET Core[2] is an open-source[3], cross-platform framework to build cloud-based web applications created by Microsoft, built on top of the .NET Core[4] platform, an open-source[5] general purpose development platform created by Microsoft. Being a cross-platform framework, its applications can be developed and run on Windows, macOS and Linux operating systems.

ASP.NET Core can also integrate with client-side[6] frameworks such as Angular, React or Bootstrap.

ASP.NET Core has built-in support for the dependency injection[7] software design pattern, a technique for achieving Inversion of Control (IoC) between classes and their dependencies. Without IoC the flow of program logic is typically determined by objects that are bound to one another. On the other hand, with inversion of control, the flow depends on the defined abstractions to be implemented that are built up during program execution. Therefore, dependency injection allows applications to be composed of loosely coupled modules and makes them easier to test and maintain. In addition, ASP.NET Core uses the Model-View-Controller (MVC)[8] architectural pattern, summarized in Figure 3.1. This pattern separates an application into three groups of components with different responsibilities:

- **Models** - represent the state of the application and any business logic or operations that should be performed by it, i.e. perform user actions and/or retrieve results of queries;
- **Views** - present the corresponding Models data through the user interface;
- **Controllers** - respond to user input and interaction, selecting which Models to work with and which Views to render.

[1]https://www.udemy.com/build-an-app-with-aspnet-core-and-angular-from-scratch/, last accessed 30th July 2019
[2]https://docs.microsoft.com/en-us/aspnet/core/?view=aspnetcore-2.1, last accessed 30th July 2019
[3]https://github.com/aspnet/AspNetCore, last accessed 30th July 2019
[4]https://docs.microsoft.com/en-us/dotnet/core/, last accessed 30th July 2019
[5]https://github.com/dotnet/core, last accessed 30th July 2019
[6]https://docs.microsoft.com/en-us/aspnet/core/client-side/index?view=aspnetcore-2.1, last accessed 30th July 2019
[7]https://docs.microsoft.com/en-us/aspnet/core/fundamentals/dependency-injection?view=aspnetcore-2.1, last accessed 30th July 2019
[8]https://docs.microsoft.com/en-us/aspnet/core/mvc/overview?view=aspnetcore-2.1, last accessed 30th July 2019

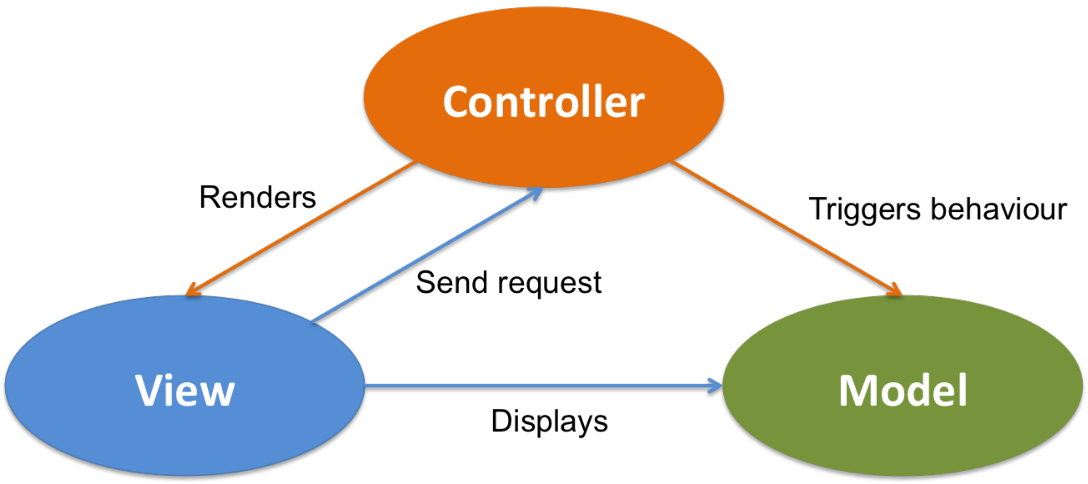


Figure 3.1: MVC Domain Model

Finally, ASP.NET Core uses Entity Framework Core[9], which is a data access technology that works as an Object-Relational Mapper (ORM), enabling interaction with databases using .NET objects. The Entity Framework enables work with data in the form of domain-specific objects and properties without the concern of the underlying database tables and columns where data is stored. This means that work can be done at a higher level of abstraction when dealing with data, and creation and maintenance of data-oriented applications is easier[10].

3.2 Angular

Angular[11] is a platform and framework for building client-side applications (i.e. applications that are in the web browser). Its applications are built by composing the following elements: (a) HTML templates with the Angular markup; (b) component classes to manage those templates; and (c) services to add application logic. These elements are all boxed in modules.

An Angular application has at least one root module (that enables bootstrapping) and may have many feature modules. Angular modules are called NgModules. They provide a compilation context for a set of components that are dedicated to an application domain, workflow or a closely related set of capabilities.

Angular applications have at least one component, which is the root component that connects a component hierarchy with the page Document Object Model (DOM). A component defines a class that contains application data and logic. It also has metadata that associates it with an HTML template that defines a view to be displayed in a target environment. HTML templates have directives that provide

[9]https://docs.microsoft.com/en-us/ef/core/, last accessed 30th July 2019
[10]https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/ef/overview, last accessed 30th July 2019
[11]https://angular.io/guide/architecture, last accessed 30th July 2019

program logic and binding markup that connects an Angular application and the DOM. There are two different types of data binding: (1) Event Binding - an application responds to user input in the target environment by updating the application data; and (2) Property Binding - values that are computed from application data are interpolated into the target environment. Before a view is displayed, the Angular framework evaluates the directives and resolves the binding syntax in the template to modify the HTML elements and the DOM, according to the application's data and logic.

Angular service classes provide data and/or logic that is not associated with a specific view, but that is going to be shared across components. Services have specific metadata that allows them to be injected into client components as a dependency. With dependency injection, component classes delegate tasks (such as fetching data from a server to the service classes).

Finally, the Angular framework has a Router NgModule that enables the definition of a navigation path among the different states and view hierarchies in an application.

Figure 3.2 represents the architecture overview of the Angular framework.

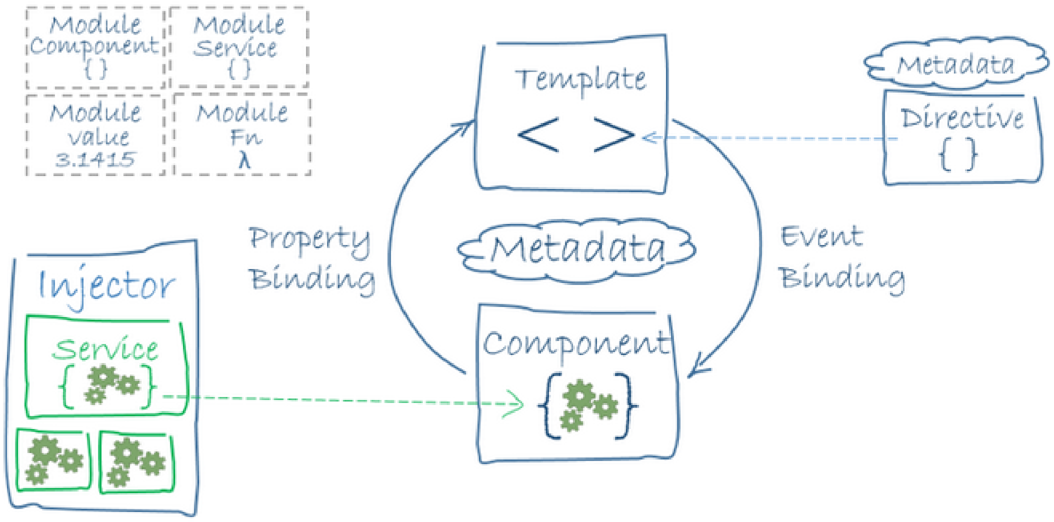


Figure 3.2: Angular Architecture Overview, (retrieved from [1])

3.3 Single Page Application (SPA)

SPAs are web applications or websites that are contained in a single web page. They are composed of individual components which can be updated independently, with the goal to be able to update parts of an interface without sending or receiving a full-page request, providing a more fluid user experience [2, 30, 31]. There is no refresh on page request, unlike in multi-page applications in which there may be a page reload on each request. All the necessary content is injected into the page through the use of

Asynchronous Javascript and XML (AJAX) and HTML templates to render the content.

On multi-page applications, HTML code is generated on the server. The template used is the server-side template, so the web page is generated and sent to the browser. However, in the case of SPAs, the HTML code is generated in real time on the client browser, according to the view.

Figure 3.3 represents the communication between the user, a SPA and a server. The communication proceeds according to the following steps:

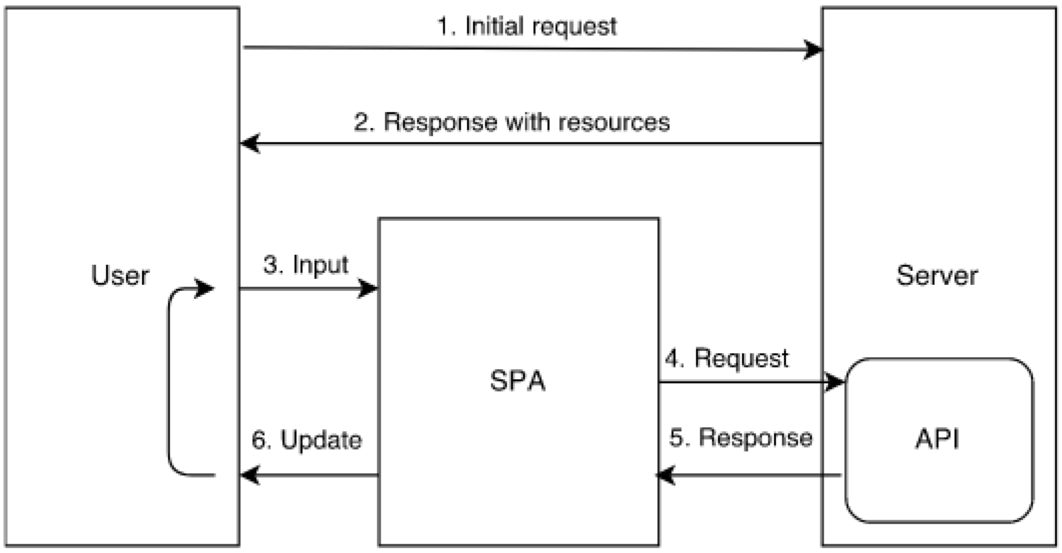


Figure 3.3: Communication between a User and a Server, (retrieved from [2])

1. **Initial Request**: HTTP request from the user to a specific URL.
2. **Response with resources**: the web server responds by sending all the resources (Javascript, CSS and HTML files). When the user receives the response from the server, the SPA is executed and loaded into the web browser.
3. **Input**: The user can send input to the SPA causing changes to its state. These changes are handled either by the application or by requesting new data via AJAX from the server API.
4. **Request**: The request from the SPA to the server-side API is asynchronous (i.e., the request returns control to the SPA before it receives the response from the API).
5. **Response**: The response is sent back to the SPA as soon as the API can handle the request.
6. **Update**: The SPA updates the components with the new data received from the server. This update is executed by a re-rendering of the DOM and when it is complete, the SPA can receive new input from the user. This is typically done by providing a callback function in the request phase, which will be invoked when the response is received from the API.

In the qDocs platform, the Single Page Application (SPA) is implemented with the Angular framework, and the API is implemented using ASP.NET Core.

4

qDocs Functional Requirements

Contents

qDocs is defined by the integration of three applications: (a) qDocs/Citizen aimed at any citizen looking for the benefits of the system; (b) qDocs/Curator particularly oriented to the curators that make up the qDocs ecosystem; and (c) qDocs/Admin to manage and configure the available features for the qDocs/Citizen and qDocs/Curator applications.

Since qDocs is a platform that already had an initial development, the proposed solution was to further develop and extend the platform in the context of the curator. The development of this work is not restricted to the qDocs/Curator application, because to enhance the curator' user's experience, changes must be made to the whole qDocs ecosystem, therefore to the qDocs/Citizen and qDocs/Admin applications. The changes in the qDocs platform will be functional, keeping the architecture of the platform.

The following sections present the functional requirements for each qDocs application, as well as the cross-cutting concerns of this work. Sections 4.1, 4.2 and 4.3 begin with a presentation of the domain model of the respective application, followed by use case diagrams to better explain what are the requirements for each application. Both domain models and use case diagrams have the same color scheme:

- **White** - if the class/use case already existed and was untouched throughout this work;
- **Yellow** - if the class/use case already existed but had technical modifications;
- **Green** - if the class/use case was created due to this work;
- **Grey** - if the class/use case does not exist in the platform but is scheduled for future work.

In addition to the color scheme, it is understood that use cases with the word "Manage" are comprised as creation, editing and deletion.

Finally, section 4.4 presents the considerations to take into account throughout this work.

4.1 qDocs/Admin

The qDocs/Admin application is used to manage and configure the available features for the qDocs/Citizen and qDocs/Curator applications — provides administration features for the qDocs ecosystem.

In this section, Figure 4.2 presents the only use case diagram because the only actor of the qDocs/Admin application is the Admin.

4.1.1 Domain Model

Figure 4.1 presents a Unified Modeling Language (UML) class diagram of the domain model related to the qDocs/Admin application.

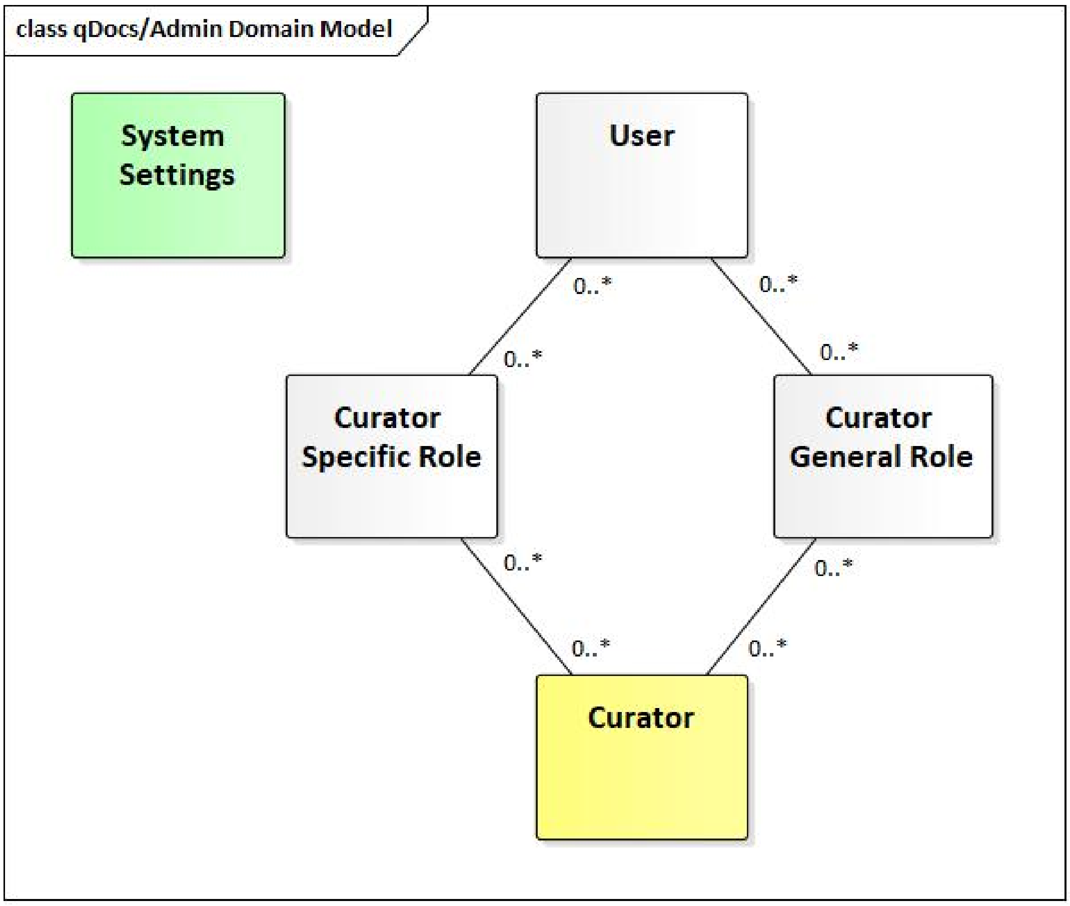


Figure 4.1: qDocs/Admin Domain Model (UML Class Diagram)

The **Users** class represents all users in the qDocs ecosystem. A **curator general role** represents a role with a certain degree of responsibility regarding a curator. A **curator specific role** represents a role that citizens may have regarding a certain curator. These roles are further explained in subsection 4.2.2. Users may have multiple curator general roles and curator specific roles, and vice-versa.

The **Curators** class represents curators in the qDocs ecosystem. A curator must be associated with at least one curator specific role and one curator general role. These roles may be associated with multiple curators.

The **System Settings** class represents all the settings that are set for the qDocs platform, e.g. language and services available for the curators. This is a singleton class.

4.1.2 Use Case

The Admin is responsible for the following use cases:

- **Manage Users** — includes creating new users and give them Admin permissions;
- **Manage Curators** — includes changing their information and associating new Administrators to them.
- **System Settings Configuration** — configure available template features (e.g. authentication using Google Id) for curators and changing general settings (e.g. language).

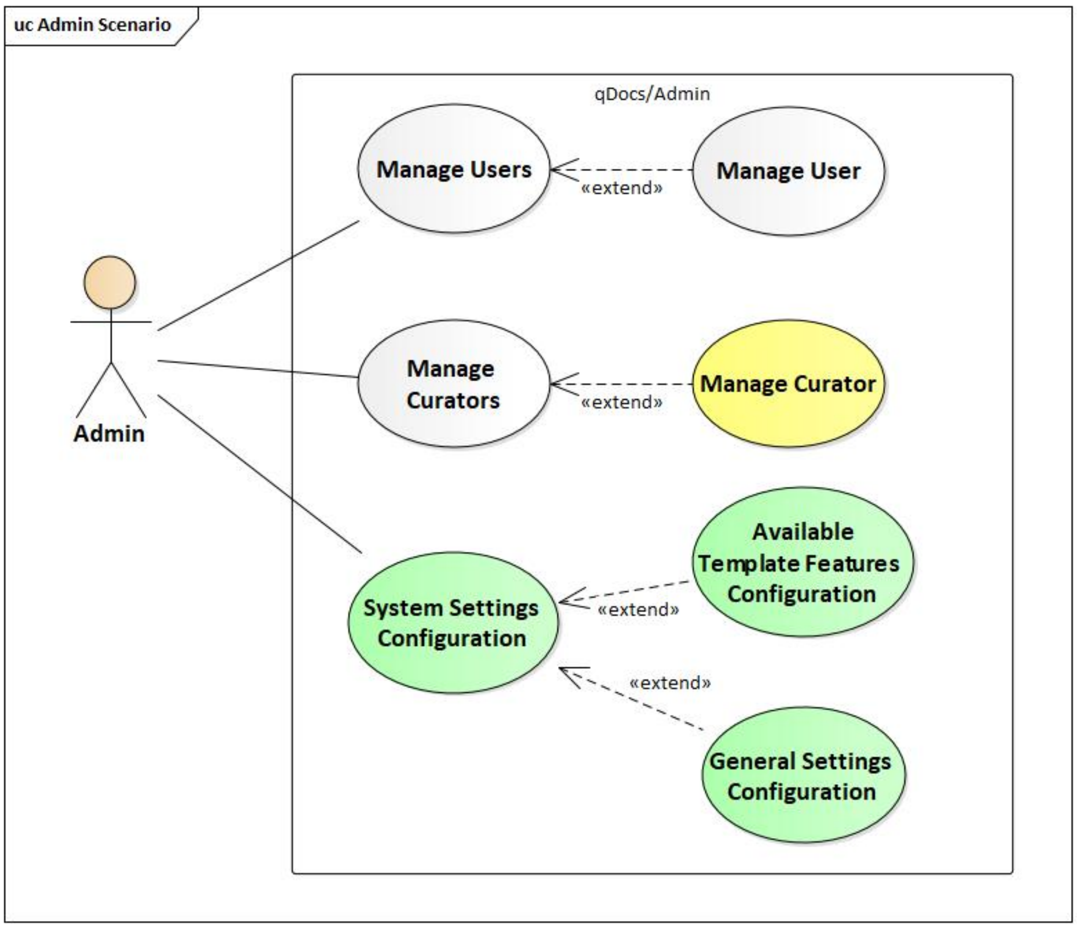


Figure 4.2: Citizen Use Case Diagram

4.2 qDocs/Curator

The qDocs/Curator application is particularly oriented to the curators that make up the qDocs ecosystem — it allows the design of electronic documents (e.g. identification documents, certificates, reports, forms, questionnaires) based on the orchestration of services and data from different curators, managed according a distributed, dynamic and secure manner.

In this section, there is a description of the different curator roles before the presentation of the use case diagrams used to explain the functional requirements of this application, since the actors of the qDocs/Curator application are defined by each curator general role. These curator general roles are the following: (a) Administrator; (b) Data Manager; (c) Templates Editor; (d) Templates Manager; (e) Auditor; (f) Documents Manager.

4.2.1 qDocs/Curator Domain Model

Figure 4.3 presents a UML class diagram of the domain model related to the qDocs/Curator application.

Document templates are represented by two classes: Document Template and Document Template Version. Document Template Version holds the information regarding a particular version such as date of creation and last update, and if the version is the one being used by the curator (if it is active or not). Document Template stores the basic information of a document template such as full and short names, and state and aggregates the first class. Document templates can have multiple versions and a document template version is associated with only one document template. Document Templates have an attribute called Template State, which is an enumeration and represents the current state of the document template. Document templates can have multiple versions and a document template version is associated with only one document template.

Document Types are used to group document templates that have similarities, e.g. forms or certificates. A document type can have multiple document templates associated with it and vice-versa.

Life Events are used to group document templates that have similarities regarding certain life events, e.g. Birth or Marriage. A life event can have multiple document templates associated with it and vice-versa.

Curator represents the curator to which the document template is associated with. A curator can have multiple document templates associated with it and a document template can only be associated with one curator.

Specific Roles (also known as User groups) are used to group document templates to specific curator roles, e.g. an university diploma can only be accessed by students, which can be a specific curator role. A specific role can have multiple document templates associated with it and vice-versa.

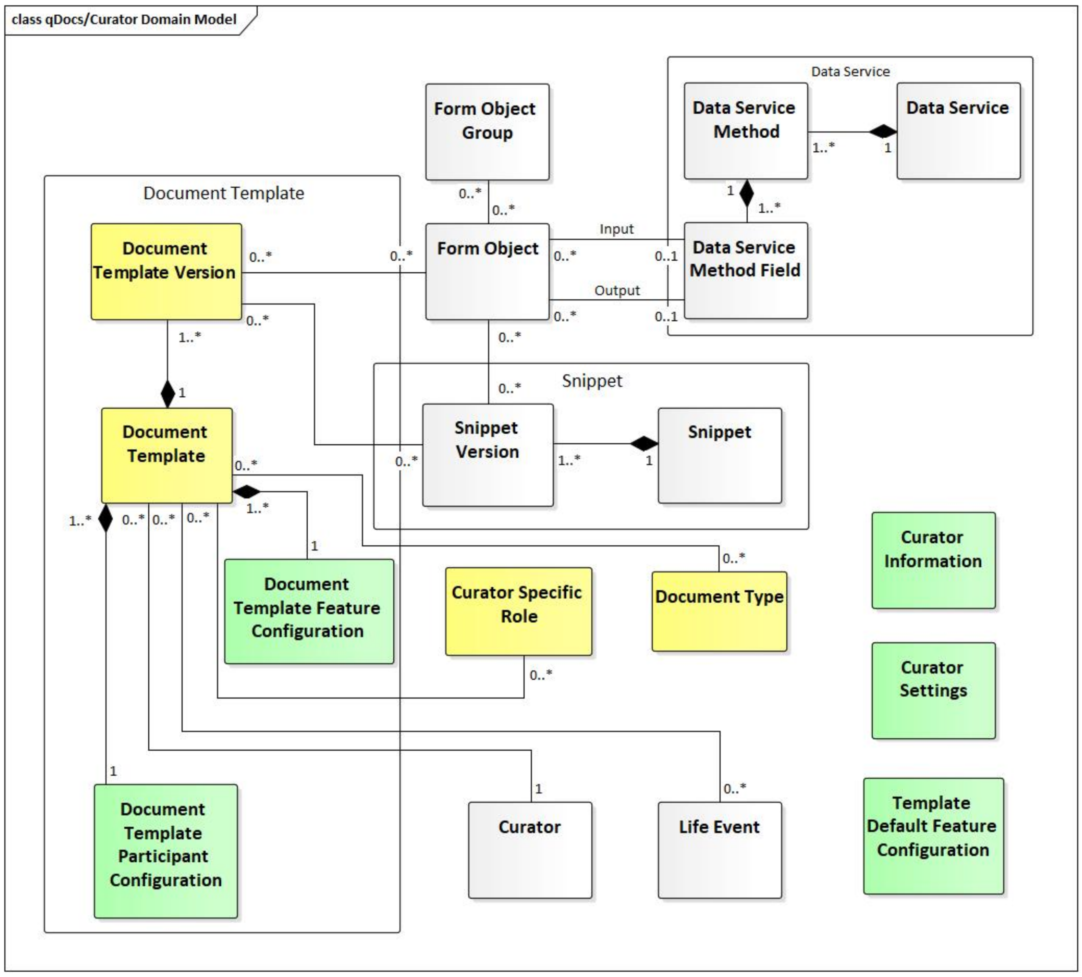


Figure 4.3: qDocs/Curator Domain Model (UML Class Diagram)

A **data service** is provided by a curator through the qBox platform. It is composed by one or more data service methods, which organize the retrieved data in tables. Data service methods are composed by one or more data service method fields, which are the content of the table cells created by the data service method.

Form objects serve as a link between document template versions and data service method fields. They are associated with document template versions and receive or retrieve data from a data service method field, depending if the data service method field is an input or an output, respectively. The data received or retrieved by the data service method field is then used by the document template. Like document groups, form objects may be associated with **form object groups**, which group form objects that have similarities, e.g. input or output.

Snippets are blocks of text edited in the qDocs platform that may be used in document template edition. They can contain all the elements available in the qDocs template editor, such as form objects or even other snippets. Snippets are represented by two classes: Snippet and Snippet Version. Snippet

Version holds the information regarding a particular version such as date of creation. Snippet stores the basic information of a snippet such as full and short names, and aggregates the first class. Snippets can have multiple versions and a snippet version is associated with only one snippet. Snippet versions may be associated with multiple templates and vice-versa.

Template feature configuration is comprised as the set of features that a document template version has enabled/disabled, e.g. Data Export or Strong Authentication. It has default values that are set by the administrator of the curator. A template feature configuration can be associated with many document template versions, but each document template version has only one template feature configuration associated to it.

Template participant configuration is comprised as the set of the participants a template version may have. Participants are entities that have a role towards a document, e.g. the president of a university must sign a student's diploma, meaning the president is a participant in the student's document in the qDocs context. Participants may be invited to access a document that was created by a citizen and perform the activities that are configured in the template participant configuration, Data Export or Sign.

Curator Information represents all the information regarding a curator, e.g. short name, full name and fiscal id. This is a singleton class.

Curator Settings represent all the settings that are set by the administrator of the curator, e.g. language and services available for that curator. This is a singleton class.

Template default feature configuration is comprised as the set of default features that a document template version has enabled/disabled, e.g. Data Export or Strong Authentication. This is a singleton class.

This work further develops the classes presented in this domain model, taking into account the different roles that each user may have.

4.2.2 Curator Roles

In the qDocs/Curator application, users can have general and specific roles. **Curator specific roles** are the ones that citizens have regarding a certain curator, e.g. in a university a specific role can be a student or a professor. Each specific role can request a determined set of documents to its curator. Each curator defines its own set of specific roles. On the other hand, **curator general roles** are associated with a certain degree of responsibility regarding a curator. Therefore, each curator general role has access to a specific set of features. To better understand the functional requirements of the qDocs/Curator application, there is a use case diagram for each curator general role (Figures 4.4, 4.5, 4.6, 4.9, 4.10 and 4.11).

4.2.3 Administrator

The Administrator is responsible for the following use cases:

- **Associate Users** — manually or in bulk.
- **Manage Users** — includes assigning general and specific roles to a single user.
- **Manage Curator-Level Entities** — manage document types, form object groups and curator specific roles (also named user groups);
- **Settings Configuration** — configure general settings (e.g. language), configure available template features (e.g. authentication using Google Id), configure the default values for templates' features (e.g. making the Data Export feature disabled);
- **Edit Curator** — change curator information (e.g. Name, Fiscal Id, Image).

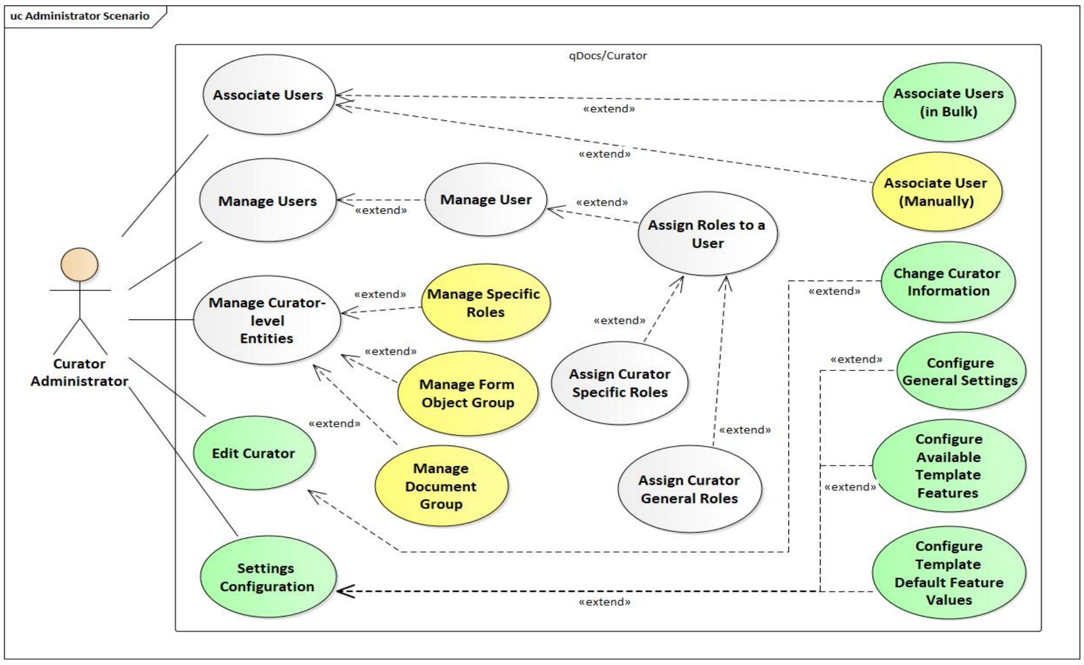

uc Administrator Scenario
qDocs/Curator
Curator Administrator
Associate Users
Manage Users
Manage User
Manage Curator-level Entities
Edit Curator
Settings Configuration
Associate Users (in Bulk)
Associate User (Manually)
Assign Roles to a User
Manage Specific Roles
Manage Form Object Group
Manage Document Group
Assign Curator Specific Roles
Assign Curator General Roles
Change Curator Information
Configure General Settings
Configure Available Template Features
Configure Template Default Feature Values
«extend»

Figure 4.4: Administrator Use Case Diagram

4.2.4 Data Manager

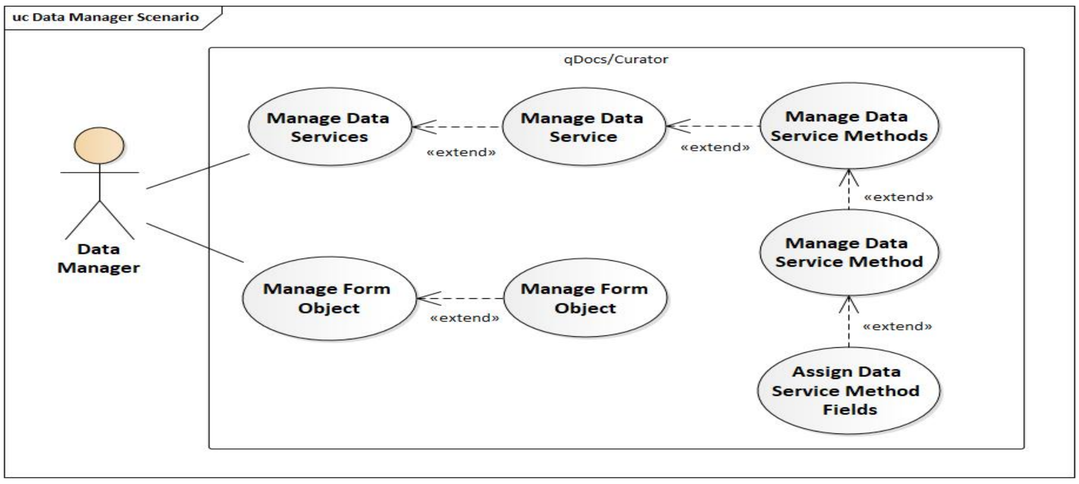


Figure 4.5: Data Manager Use Case Diagram

The Data Manager is responsible for the following use cases:

- **Manage Data Services** — includes managing the respective data service methods and assigning data service method fields to them;
- **Manage Form Objects**.

4.2.5 Templates Editor

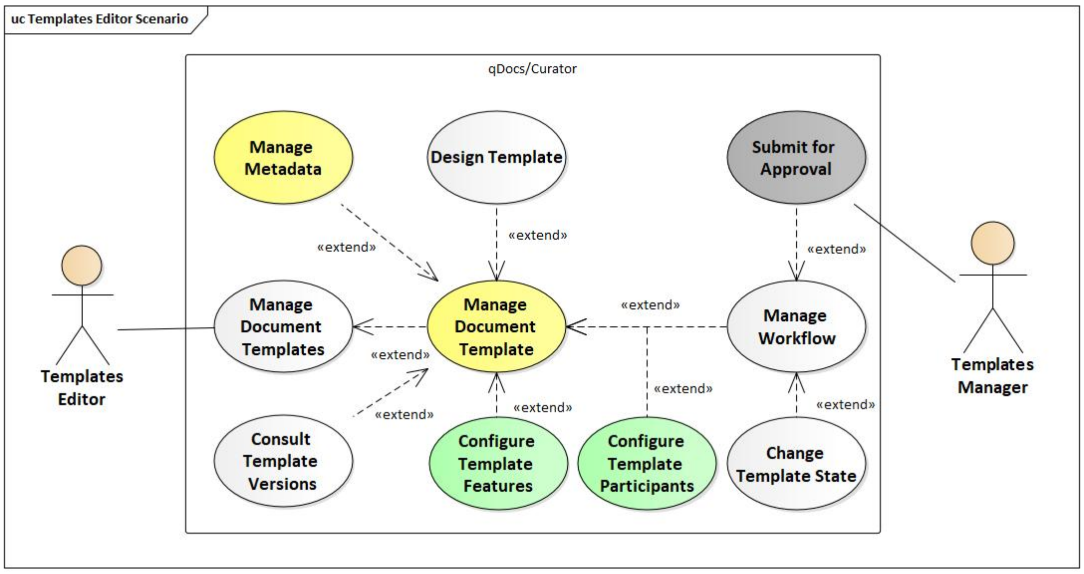


Figure 4.6: Templates Editor Use Case Diagram

The Templates Editor is responsible for the following use cases:

- **Manage Document Templates** — includes designing them and managing their metadata, i.e. configure their information, groups, features and their participants;
- **Manage Workflow** — change the state of a document template, including submitting the document template for approval;
- **Consult Versions** — consult all versions created of a document template;
- **Configure Features** — configure a document template's features;
- **Configure Template Participants** — configure which users can access and manipulate the document, i.e. the document's participants.

A workflow is comprised as a set of states that are associated with a document template. There are several states for document templates. They are represented in Figure 4.7 and their sequence happens as follows:

- **Created** — after creating a document template;
- **Pending for Approval** — after the document template is designed it can be submitted for approval or if it was rejected it can be submitted for approval again;
- **Rejected** — if the responsible for approving the document template rejects it;
- **Active** — if the responsible for approving the document template approves it or if the responsible for the document template activates it after it is inactive or deprecated;
- **Deprecated** — when a new document template version is activated;
- **Inactive** — if the responsible for the document template deactivates it.

Figure 4.8 presents a state machine diagram for a document template's workflow.

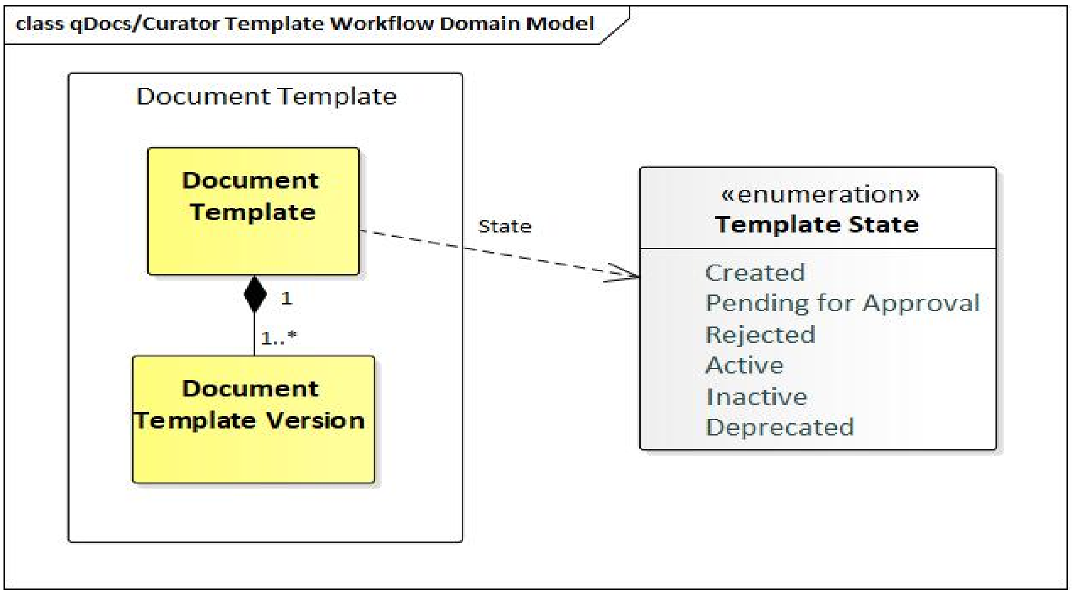


Figure 4.7: Document Template Workflow Domain Model(UML Class Diagram)

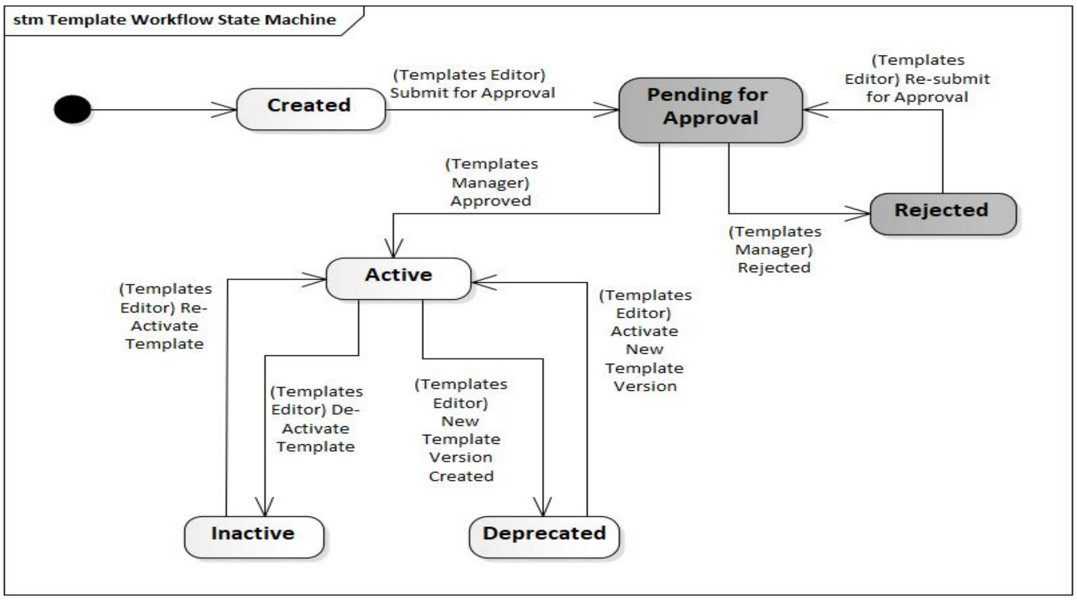

stm Template Workflow State Machine
Created
(Templates Editor) Submit for Approval
Pending for Approval
(Templates Editor) Re-submit for Approval
(Templates Manager) Approved
Rejected
(Templates Manager) Rejected
Active
(Templates Editor) Re-Activate Template
(Templates Editor) De-Activate Template
(Templates Editor) New Template Version Created
(Templates Editor) Activate New Template Version
Inactive
Deprecated

Figure 4.8: Document Template Workflow State Machine Diagram

4.2.6 Templates Manager

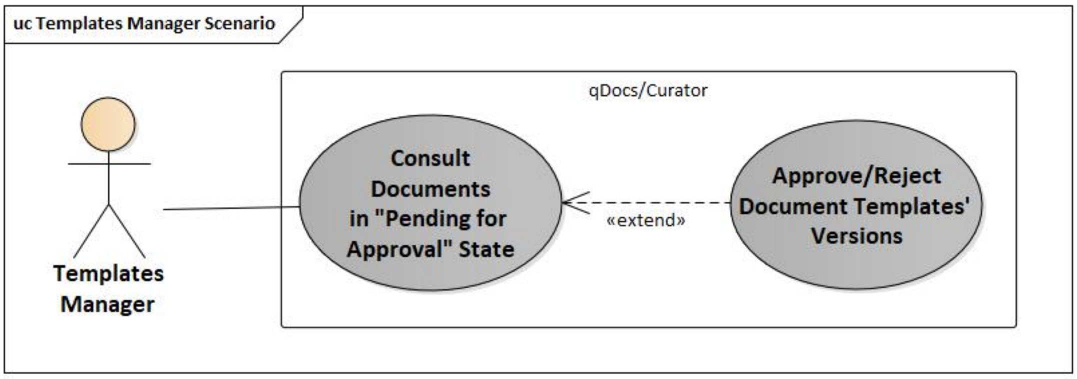

uc Templates Manager Scenario
qDocs/Curator
Templates Manager
Consult Documents in "Pending for Approval" State
«extend»
Approve/Reject Document Templates' Versions

Figure 4.9: Templates Manager Use Case Diagram

The Templates Manager is responsible for the following use cases:

- **Consult Document Templates in ”Pending for Approval” State** — includes approving/rejecting document template versions that are submitted for approval, therefore in the ”Pending for Approval” state.

4.2.7 Auditor

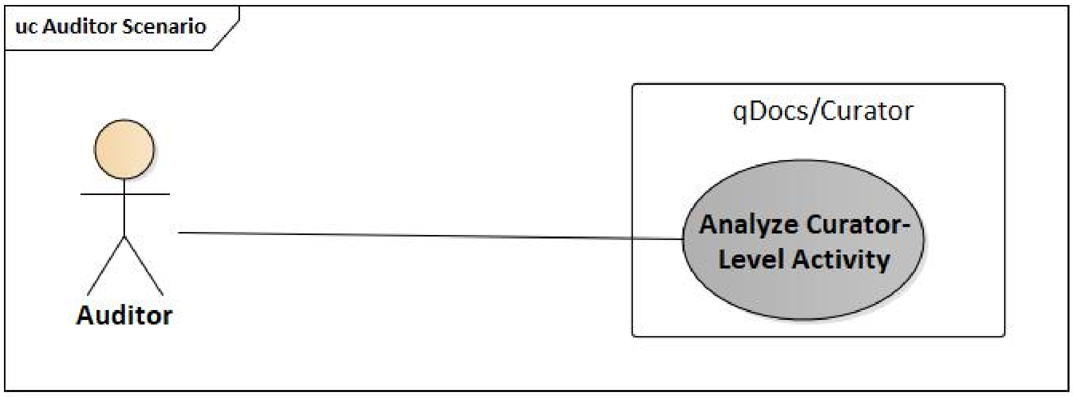


Figure 4.10: Auditor Use Case Diagram

The Auditor is responsible for the following use cases:

- **Analyze Curator-Level Activity** — includes visualizing dashboards and timeline graphics with the most relevant KPIs at Curator-Level.

4.2.8 Documents Manager

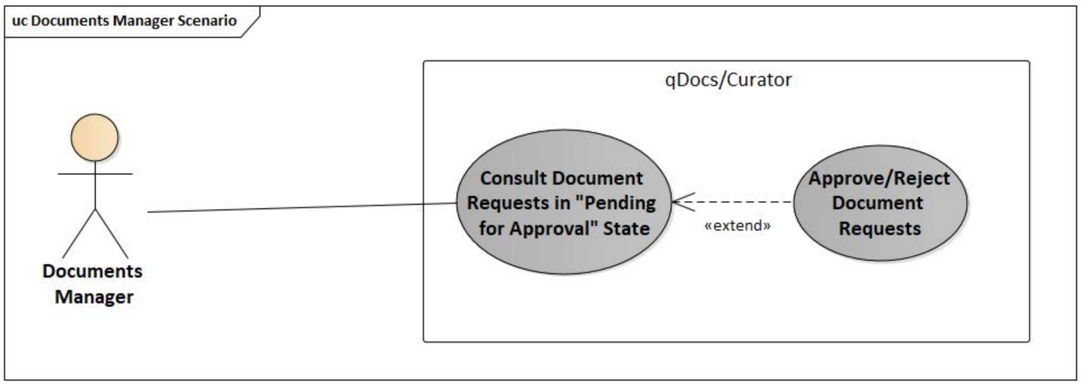

uc Documents Manager Scenario
qDocs/Curator
Documents Manager
Consult Document Requests in "Pending for Approval" State
«extend»
Approve/Reject Document Requests

Figure 4.11: Documents Manager Use Case Diagram

The Documents Manager is responsible for the following use cases:

- **Consult Document in ”Pending for Approval” State** — includes approving/rejecting submission requests made by citizens, i.e. when a citizen requests a document, the Documents Manager consults the request, and then it can approve (then the citizen who requested the document can have access to it through its user interface) or reject (the citizen shall be notified of the situation and request the document again) that request.

4.3 qDocs/Citizen

The qDocs/Citizen application is aimed at any citizen looking for the benefits of the system — allows citizens to access and manage their documents in a secure manner.

In this section, Figure 4.13 presents the only use case diagram because the only actor of the qDocs/Citizen application is the Citizen.

4.3.1 Domain Model

Figure 4.12 presents a UML class diagram of the domain model related to the qDocs/Citizen application.

Document represents a document that is accessible by a citizen.

User represents all citizens using the qDocs platform. A user may be associated with documents as two different entities: owner or participant. If the user is a owner, then it may have multiple documents and a document can only have one owner. If the user is a participant, then it can be associated with multiple documents and vice-versa.

Document Share represents all the information regarding a document that was shared, i.e. if a document is shared, it has information regarding who shared the document and who accessed it besides the owner of the document.

Curator represents the curator to which a document is associated. A document may be associated with one curator but a curator may be associated with multiple documents.

Document Types is comprised as the document groups to which a document is associated (e.g. forms or certificates). A document may be associated with multiple document types and vice-versa.

Life Events is comprised as the life events to which a document is associated (e.g. birth or marriage). A document may be associated with multiple document types and vice-versa.

Curator Specific Roles (also known as user group) is comprised as the roles to which a document is associated (e.g. student or president). A document may be associated with multiple user groups and vice-versa.

Citizen Information represents all the information regarding a citizen, e.g. id, name and curators to which a citizen is associated with. This is a singleton class.

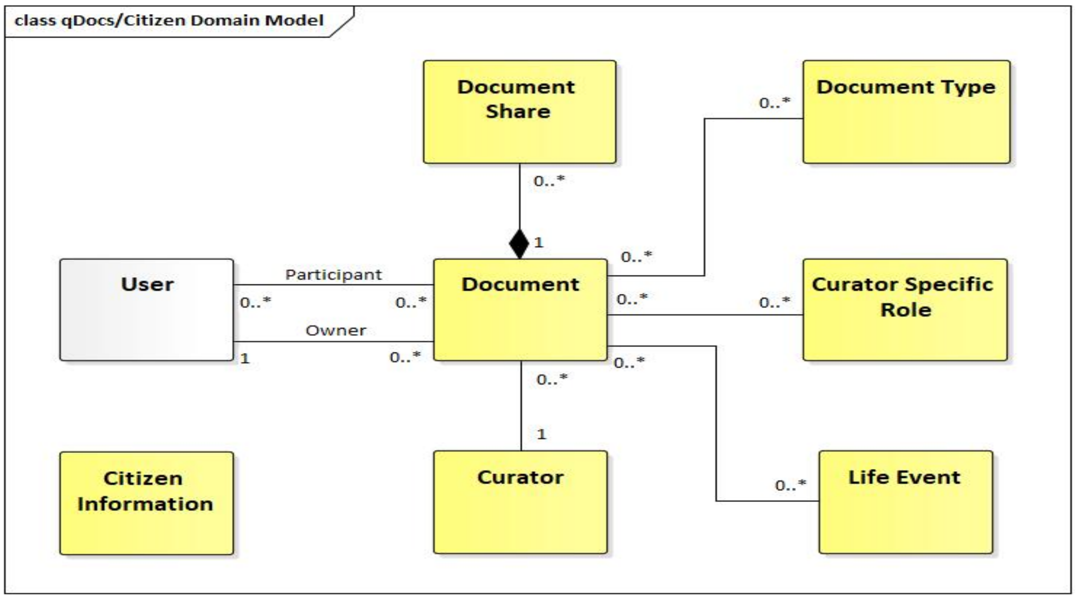


Figure 4.12: qDocs/Citizen Domain Model (UML Class Diagram)

4.3.2 Use Case

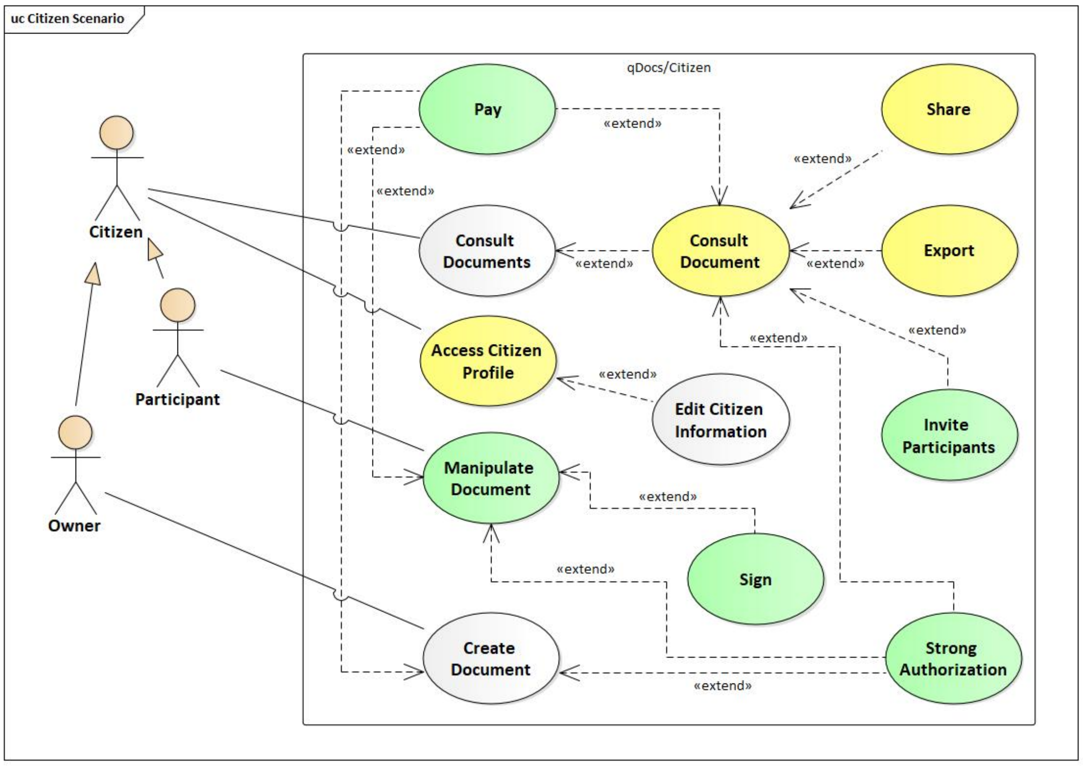


Figure 4.13: Citizen Use Case Diagram

The Citizen is responsible for the following use cases:

- **Create Document** — can only create documents to which it has access and if it can be the owner of the document;
- **Consult Documents** — this includes inviting participants to the document and use enabled document features, such as Share or Export;
- **Manipulate Document** — only participants can do this, it includes using features that will affect a document, such as Sign;
- **Strong Authentication** — a citizen may be requested to go through a strong authentication process in order to create, consult and manipulate a document.
- **Pay** — a citizen may be requested to pay a determined value in order to create, consult or manipulate a document.
- **Access Citizen Profile** — a citizen can access its profile to check its information, including the curators to which it is associated to; a citizen can also edit its information, e.g. its alias and email.

4.4 Cross-Cutting Concerns

During the development of the qDocs platform there are some requirements that are cross-cutting to the whole qDocs platform. The platform must provide security features such as confidentiality, integrity, authenticity, and non-repudiation. It should provide security at communication-level using a Kerberos-based protocol; support Personal Data Privacy; support User Authentication based on citizen id and password or other means such as AMA's autenticacao.gov.pt service; and support User Authorization based on general and specific roles. Integration is another important concern, since qDocs is a cloud-based platform must support interoperability with other external system that will request information from different sources, some of them Curators with legacy applications [25, 26]. The platform must integrate with AMA's managed external systems. This will be done with qBox, which is a Curator' specific external system to provide such integration. Finally, the platform must support multi-language and take UI aspects, i.e the platform must be provide an attractive, simple and easy to use user experience.

In this work the focus is on the User Authorization based on general and specific roles and everything related with the UI.

5 qDocs Technical Description

Contents

This chapter presents what is accomplished with this work. Section 5.1 presents the architecture of the platform and section 5.2 describes what is developed in each qDocs application.

5.1 Architecture Overview

Since it is a web based solution, the architecture is the one of a full stack application. Figure 5.1 shows the technological architecture of the qDocs platform with the technologies used in each side of the application, including the database. The application has three major components: client, server and database. Users access the application through the client and make requests to the server through it. The server then retrieves the requested information from the database and sends it to the client. The database uses PostgreSQL[1], which is open-source object-relational database system that uses and extends the SQL language. The server uses the ASP.NET CORE framework and the client uses the Angular 4 framework. These two technologies are already described in Chapter 3 and are the ones already being used, since qDocs is already in an early stage of development before this work.

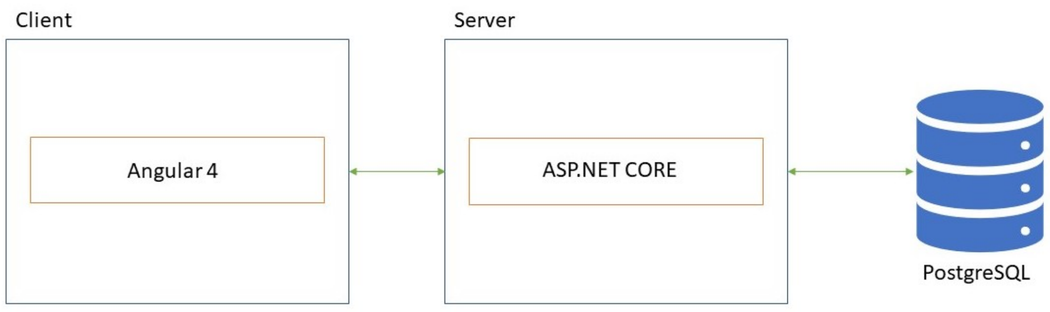


Figure 5.1: qDocs Technological Architecture

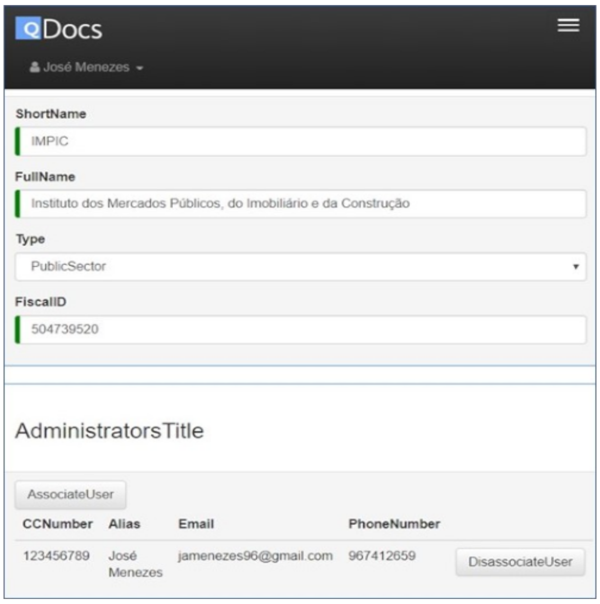
5.2 Design and Development

This section describes the technical implementation performed in the qDocs platform. Subsections 5.2.1, 5.2.2 and 5.2.3 present the implementations that are specific to each qDocs application and subsection 5.2.4 describe the implementations that are common to all applications.

[1] https://www.postgresql.org/about/, last accessed 30th July 2019

5.2.1 qDocs/Admin

The qDocs/Admin application provides administration features for the qDocs ecosystem. It was already possible to create new curators and edit their information before this work, but it was not possible to edit all information regarding a curator, so these minor changes were developed. Also the settings interface was implemented with this work. Figure 5.2 presents the Edit Curator Interface and figure 5.3 presents the Settings Interface.



ShortName: IMPIC

FullName: Instituto dos Mercados Públicos, do Imobiliário e da Construção

Type: PublicSector

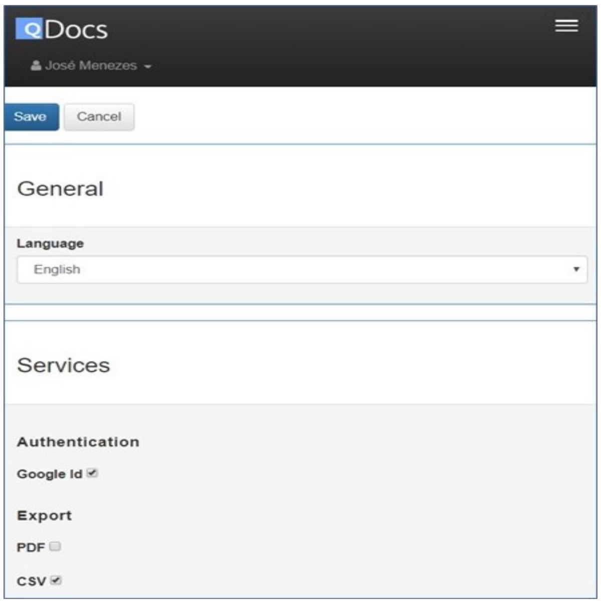
FiscalID: 504739520

AdministratorsTitle

AssociateUser

CCNumber	Alias	Email	PhoneNumber	
123456789	José Menezes	jamenezes96@gmail.com	967412659	DisassociateUser

Figure 5.2: Edit Curator Interface



General

Language: English

Services

Authentication

Google Id ✓

Export

PDF ☐

CSV ✓

Figure 5.3: Settings Interface

In figure 5.2 a user with admin privileges can edit a certain curator's information, including its short name, full name, type (e.g. public sector, education or other) and fiscal id. A user could not change the type and fiscal id values before this work. Also the user with access to this interface can see which citizens are associated to the curator with the Administrator general role.

In figure 5.3 a user can change the general settings of the qDocs platform (comprised only as the language) and activate/deactivate services that are going to be used by the curators, such as document export as pdf or as csv files.

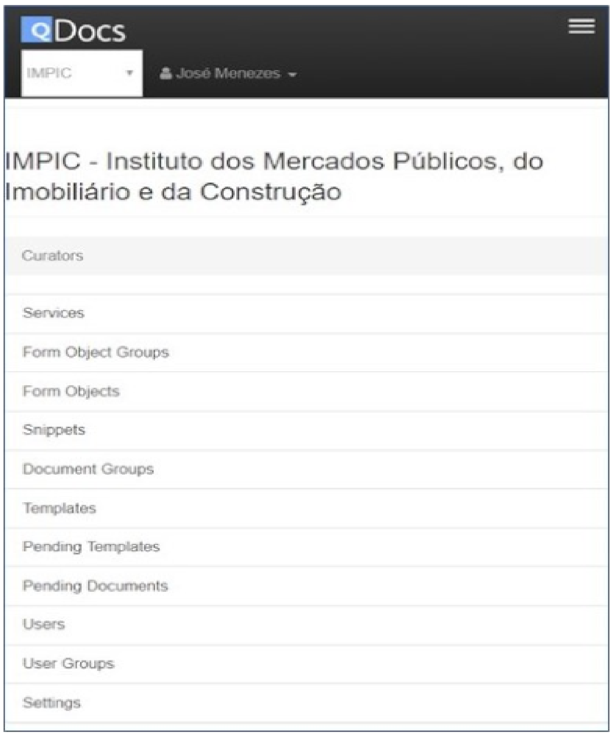
5.2.2 qDocs/Curator

The qDocs/Curator application allows curators to design and configure their document templates and make them available for citizens. Before this work, a user from the curator could:

- Associate users to the curator;
- Manage users associated to the curator;
- Manage curator-level entities;
- Manage data services;
- Manage form objects;
- Manage document templates;
- Design document templates, using form objects and snippets.

With this work, some features were enhanced (e.g. all managing interfaces have pagination and search bars) and others were created in order to provide a better experience for the users of the curator (e.g. associating users in bulk and configure features for document templates). Figures 5.4, 5.5, 5.6, 5.7, 5.8, 5.10, 5.11, 5.12, 5.13, 5.9 present the qDocs/Curator interfaces that included major technical implementation. These implementations include:

- Creation of curator General Roles;
- Settings interface development;
- Bulk association of users to the curator;
- Item deletion option (e.g. a Administrator could not delete a Form Object Group he created);
- Participant configuration;
- Template feature configuration;
- Search bars, pagination and filters to improve item search;
- Buttons to improve navigation between interfaces;



IMPIC - Instituto dos Mercados Públicos, do Imobiliário e da Construção

Curators

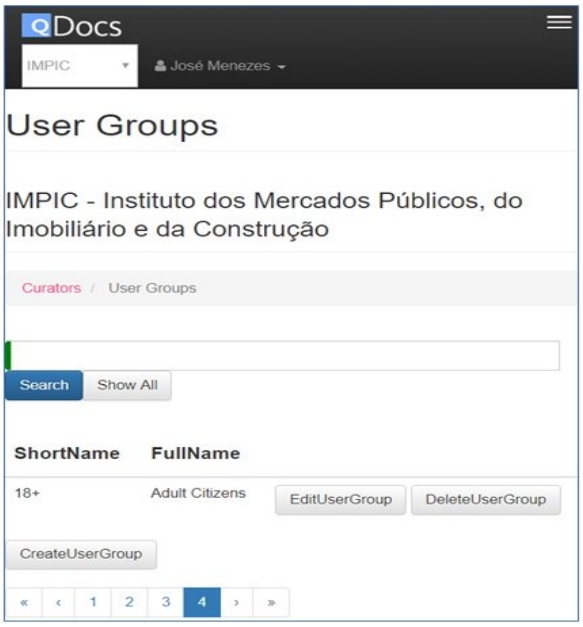
- Services
- Form Object Groups
- Form Objects
- Snippets
- Document Groups
- Templates
- Pending Templates
- Pending Documents
- Users
- User Groups
- Settings

Figure 5.4: qDocs/Curator Home Interface

Figure 5.4 presents the qDocs/Curator initial interface for a curator user. It has the name of the curator followed by several navigation options. The option User Groups is the one used to manage curator specific roles (User Groups was the name used before this work and was not changed). These navigation options are available to determined curator general roles:

- **Administrator** — can access the Form Object Groups, Document Groups, Users, User Groups and Settings options;
- **Data Manager** — can access the Services, Form Objects and Snippets options;
- **Templates Editor** — can access the Templates option;
- **Templates Manager** — can access the Pending Templates option (currently it does nothing, it is scheduled for future work, as stated in chapter 7);
- **Auditor** — does not have access to any option, since its functional requirements are scheduled for future work, as stated in chapter 7.
- **Documents Manager** — can access the Pending Documents option (currently it does nothing, it is scheduled for future work, as stated in chapter 7);

These curator general roles and curator specific roles are already described in Chapter 4.



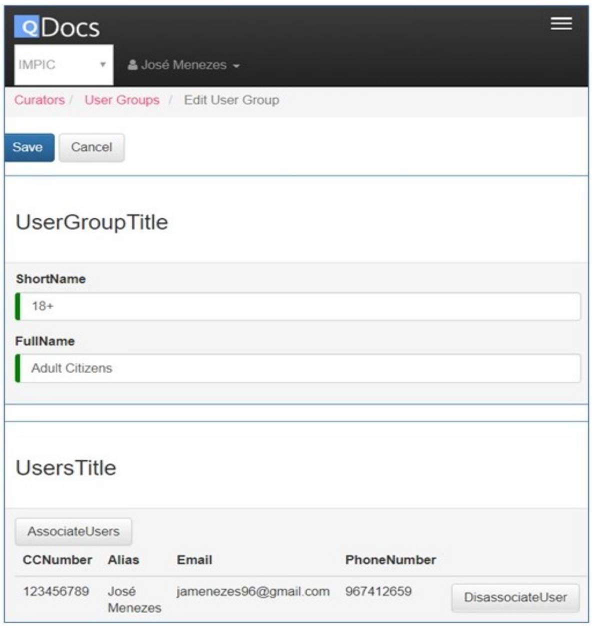
User Groups

IMPIC - Instituto dos Mercados Públicos, do Imobiliário e da Construção

Curators / User Groups

ShortName	FullName		
18+	Adult Citizens	EditUserGroup	DeleteUserGroup

Figure 5.5: List Specific Roles Interface



UserGroupTitle

ShortName: 18+

FullName: Adult Citizens

UsersTitle

CCNumber	Alias	Email	PhoneNumber	
123456789	José Menezes	jamenezes96@gmail.com	967412659	DisassociateUser

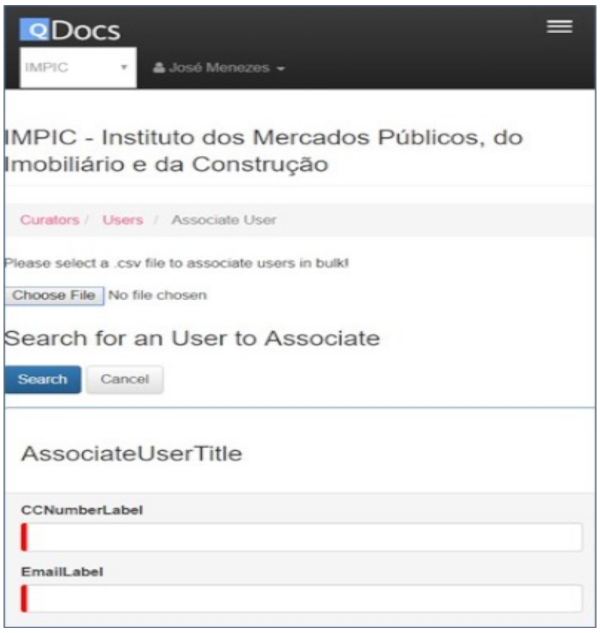
Figure 5.6: Edit Specific Role Interface

Figure 5.5 shows the UI presented after choosing the option ”User Groups”. It presents a list with all curator specific roles created in the presented curator. With this work, the pagination and search features were created so the curator user can access the desired ”user group” easily. Also the Delete User Group feature was created. If the curator user chose any other option, the UI would be similar, and was affected in the same manner by this work, therefore not all other options are presented.

Figure 5.6 is the UI presented after choosing ”Edit User Group”, presented in Figure 5.5. In this menu, a curator Administrator may change the short and full names of the curator specific role selected and may consult which citizens are associated to it and associate more (this interface was not affected by this work). If the curator wanted to edit a determined curator-level entity, the interface would be similar, therefore not all other options presented.

Figure 5.7 is the UI presented when a curator Administrator wants to associate users to the curator. It can associate a single user by providing its Email and Citizen Card Number (this feature was already available before this work) or associate users in bulk by providing a csv file with the citizens' Citizen Card Number, Email, Curator General Role and Curator Specific Role (this feature was created with this work).

Figure 5.8 is the interface presented when a curator Administrator wants to edit information regarding another curator user. It can consult the users information (CCNumber, Alias, Email and Phone Number) and assign General and Specific Roles to the user. Before this work, it was not possible to assign Specific Roles to a user through this interface. In Figure 5.8 there is a General Role named Citizen that was not presented. This General Role is used to associate a citizen to a curator, without any responsibility. This has to be done because for a citizen to request documents from a curator, it must be associated to with a general role, that was how the platform was designed before this work, and the architecture was kept.


qDocs

IMPIC ▾ José Menezes ▾

IMPIC - Instituto dos Mercados Públicos, do Imobiliário e da Construção

Curators / Users / Associate User

Please select a .csv file to associate users in bulk!

Choose File No file chosen

Search for an User to Associate

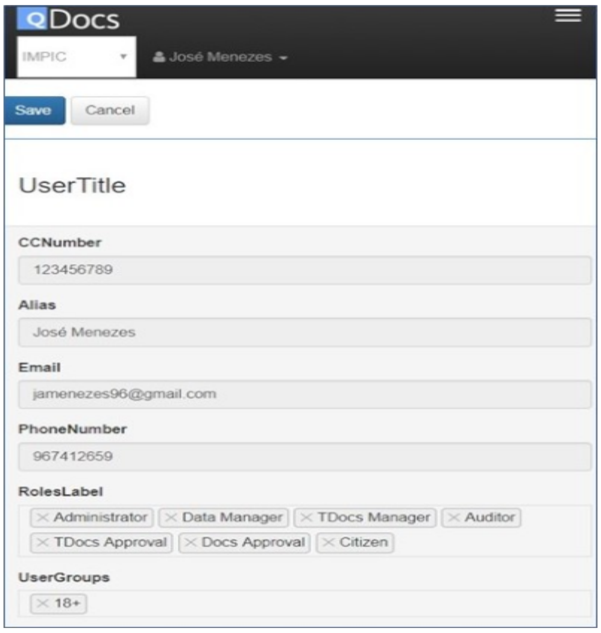
Search Cancel

AssociateUserTitle

CCNumberLabel

EmailLabel

Figure 5.7: Associate Users Interface


qDocs

IMPIC ▾ José Menezes ▾

Save Cancel

UserTitle

CCNumber: 123456789

Alias: José Menezes

Email: jamenezes96@gmail.com

PhoneNumber: 967412659

RolesLabel: × Administrator × Data Manager × TDocs Manager × Auditor × TDocs Approval × Docs Approval × Citizen

UserGroups: × 18+

Figure 5.8: Edit User Interface

Figure 5.9 presents the Settings Configuration interface divided into two parts. In figure 5.9(a) a curator Administrator can change its curator information (short name, full name, country, image url and fiscal id) and change its general settings (comprised only as the language). In figure 5.9(b) a curator Administrator can change the services available for template configuration and the default features for templates. Regarding services available, the ones that appear for a curator Data Manager to select are the ones enable by the Admin of the qDocs platform.

Figure 5.10 is the UI that lists all document templates of a curator. Besides a search bar and pagination, with this work filtering options were added. A user may search a document template using filters such as Life Events or Document Types and, if available, using sub-filters such as Birth or Taxes (for Life Events, for example). Through this interface a user can choose to edit a document template's metadata, content or participants or to view the active version of that document template, if available.

Figure 5.11 is the Edit Template Participants Interface. In this UI a curator Templates Editor can change the number of participants and their information. A participant's information is comprised as its role (a name given by the curator Templates Editor), its multiplicity and the features it has available, such as Sign a document or Export it. The multiplicity parameter may have four options with the following meaning:

- **Opcional** — a document may have one participant with this role (0..1);

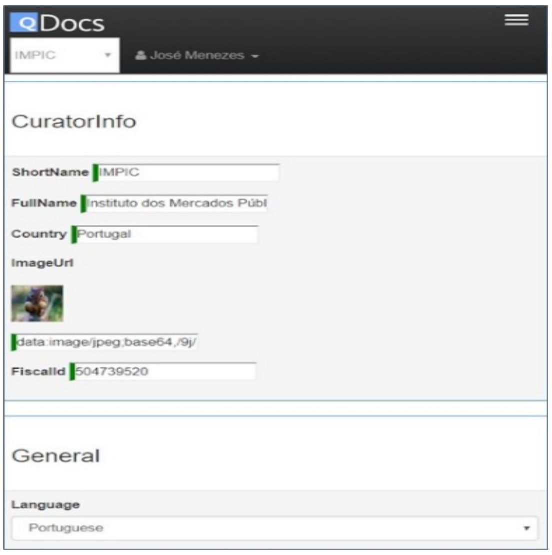
- **Mandatory** — a document must have one participant with this role (1);
- **ZeroMultiple** — a document may have zero or more participants with this role (*);
- **OneMultiple** — a document must have at least one participant with this role, but may have more (1..*).

Figure 5.12 presents the template Edit Content UI. Figure 5.12(a) is the initial template Edit Content UI. It provides all common text edition features such as text alignment and font size. In addition to these common features, it has a blue button with a *Q* letter that enables all features relative to document template edition. Figure 5.12(b) is the modal presented after pressing this button. Through this interface a curator user may insert snippets, form objects, conditions, formulas and IfElse conditions into the document template. These features were all available before this work.

Figure 5.13 presents the Template Edit Metadata interface divided in two parts. In figure 5.13(a) a curator user can consult the document template's short name, full name, version, creation date (the time when the template was created), publication date (the time when the template's version became active) and history, i.e. consult the document template's previous versions (this interface is not presented because it is similar to the other list interfaces). Regarding figure 5.13(a) a curator user can also enable the sharing types of a document, which can be:

- **Key** — a citizen receives a key and sends it to other citizens so they can access the document;
- **hyperlink** — a citizen receives an hyperlink and other citizens can access the document through it.

All of these features were already available before this work. In figure 5.13(b) a curator user can configure the document template's features such as Export, Strong Authentication, Payment and the Payment Value, if the previous is enabled. This was created with this work. A curator user can also assign the document template to document groups, comprised as Life Events, Document Types and Organization Groups (named document groups in the curator home interface). Both Life Events and Document Types are predetermined by the qDocs platform. In addition, it can assign document templates to specific roles and consult the scope and state of the template. This was already available before this work.



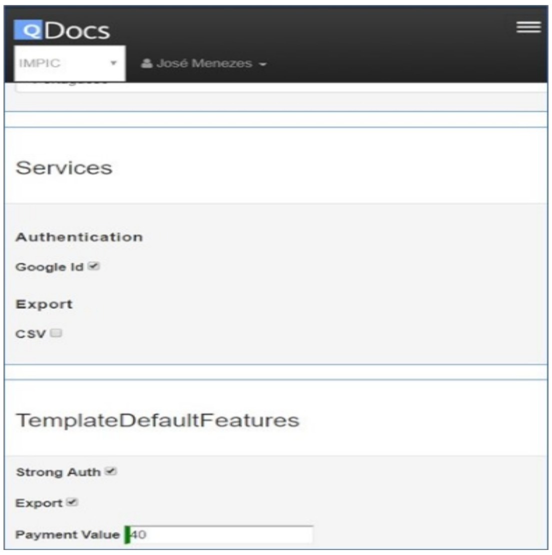
CuratorInfo

ShortName IMPIC
FullName Instituto dos Mercados Públ
Country Portugal
ImageUrl
data:image/jpeg;base64,/9j/
FiscalId 504739520

General

Language
Portuguese

(a)



Services

Authentication
Google Id ☑

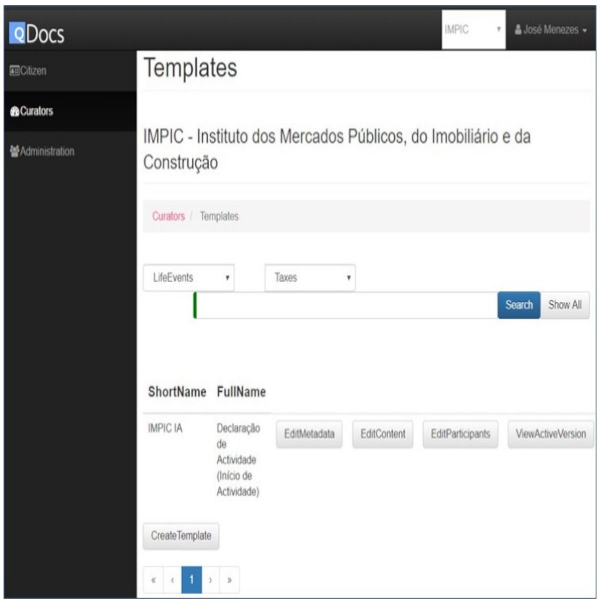
Export
CSV ☐

TemplateDefaultFeatures

Strong Auth ☑
Export ☑
Payment Value 40

(b)

Figure 5.9: Settings Configuration Interface



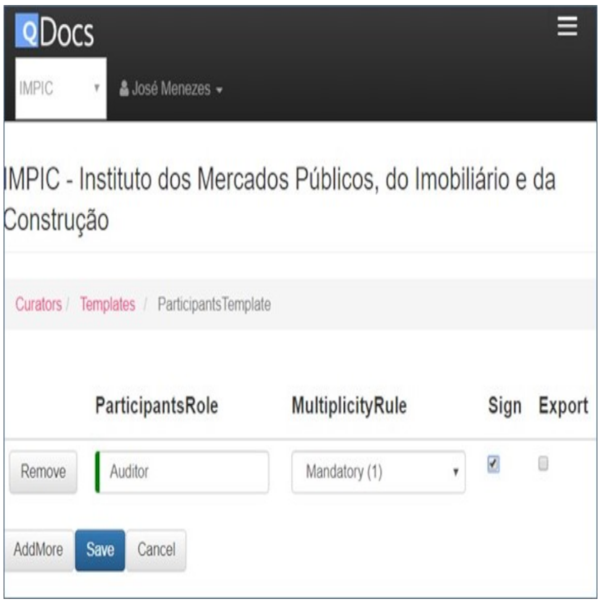
Templates

IMPIC - Instituto dos Mercados Públicos, do Imobiliário e da Construção

Curators / Templates

ShortName	FullName
IMPIC IA	Declaração de Actividade (Início de Actividade)

Figure 5.10: List Templates Interface

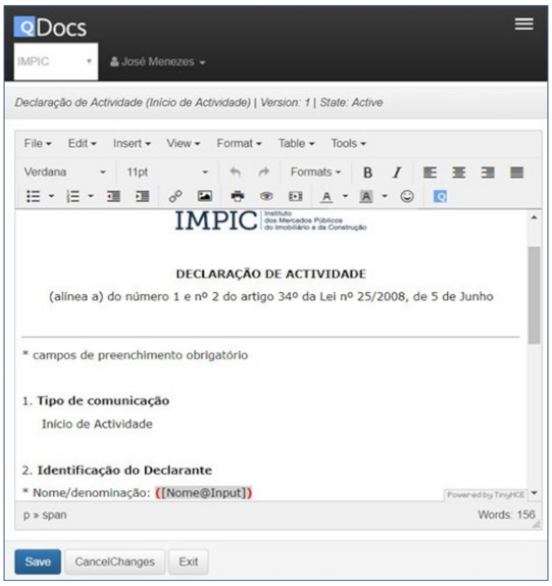


IMPIC - Instituto dos Mercados Públicos, do Imobiliário e da Construção

Curators / Templates / ParticipantsTemplate

ParticipantsRole	MultiplicityRule	Sign	Export
Auditor	Mandatory (1)	☑	☐

Figure 5.11: Edit Template Participants Interface



Declaração de Actividade (Início de Actividade) | Version: 1 | State: Active

IMPIC

DECLARAÇÃO DE ACTIVIDADE

(alínea a) do número 1 e nº 2 do artigo 34º da Lei nº 25/2008, de 5 de Junho

* campos de preenchimento obrigatório

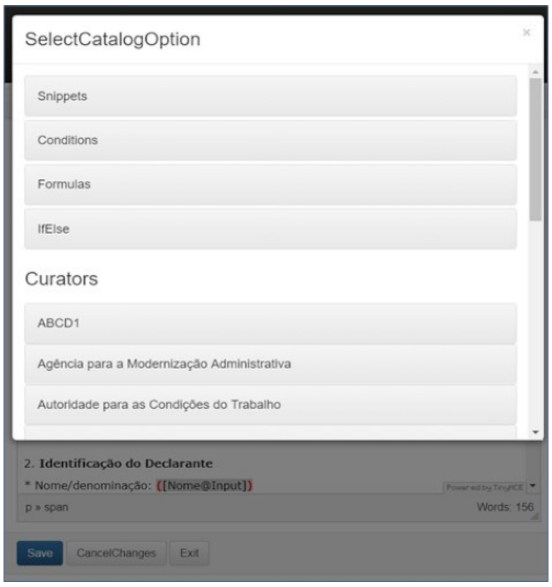
1. Tipo de comunicação

Início de Actividade

2. Identificação do Declarante

* Nome/denominação: {[Nome@Input]}

p » span Words: 156



2. Identificação do Declarante

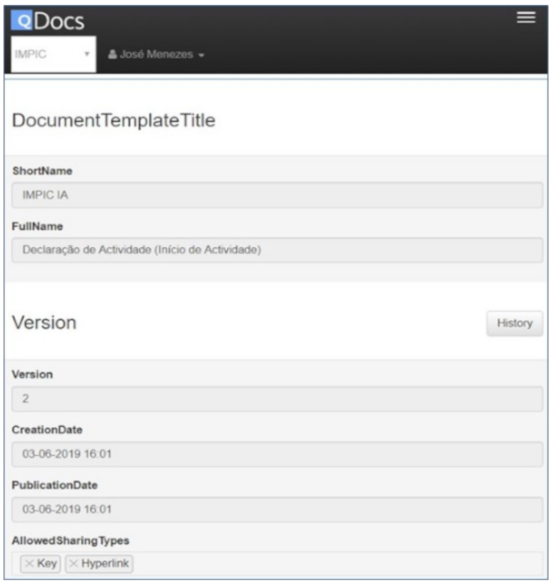
* Nome/denominação: {[Nome@Input]}

p » span Words: 156

(a) Edit Content Interface

(b) QDocs Edition Options Interface

Figure 5.12: Template Content Edition Interfaces



DocumentTemplateTitle

ShortName: IMPIC IA

FullName: Declaração de Actividade (Início de Actividade)

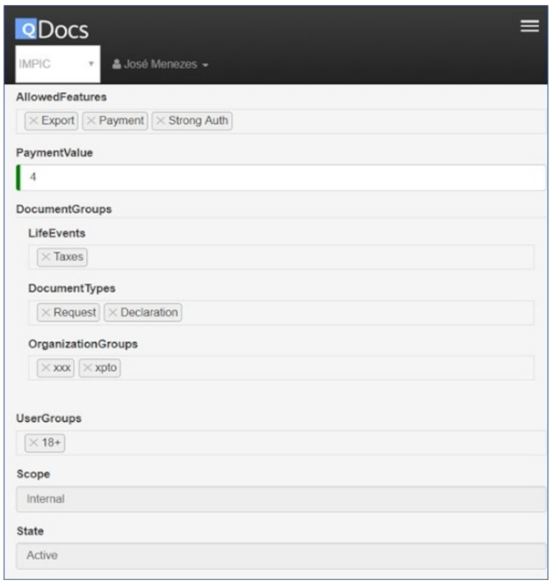
Version

Version: 2

CreationDate: 03-06-2019 16:01

PublicationDate: 03-06-2019 16:01

AllowedSharingTypes: Key, Hyperlink



AllowedFeatures: Export, Payment, Strong Auth

PaymentValue: 4

DocumentGroups

LifeEvents: Taxes

DocumentTypes: Request, Declaration

OrganizationGroups: xxx, xpto

UserGroups: 18+

Scope: Internal

State: Active

(a)

(b)

Figure 5.13: Template Metadata Edition Interface

5.2.3 qDocs/Citizen

The qDocs/Citizen application is aimed at any citizen looking for the benefits of the system. Before this work, a citizen user could:

- Create documents;
- Consult its documents;
- Use the share document feature;
- Access and edit its profile information.

The purpose of this work regarding the qDocs/Citizen application was to test and validate what was implemented in the qDocs/Curator application, e.g. the participant configuration and document feature availability. Figures 5.14, 5.15, 5.16, 5.17, 5.18 and 5.19 present the qDocs/Citizen interfaces that suffered technical implementation in the scope of this dissertation.

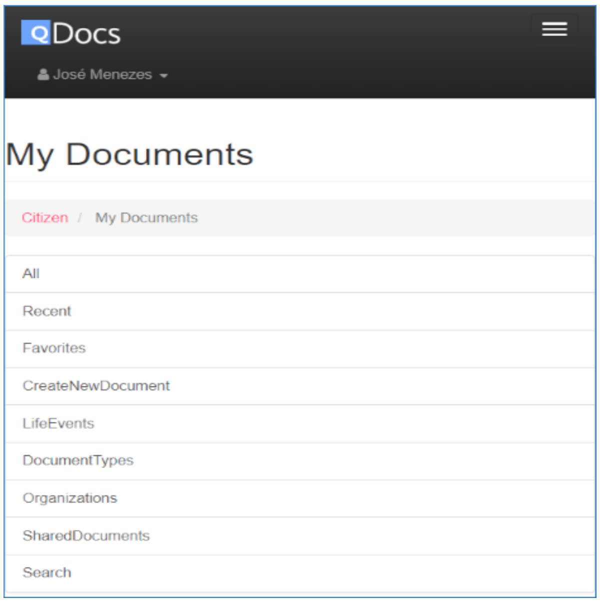
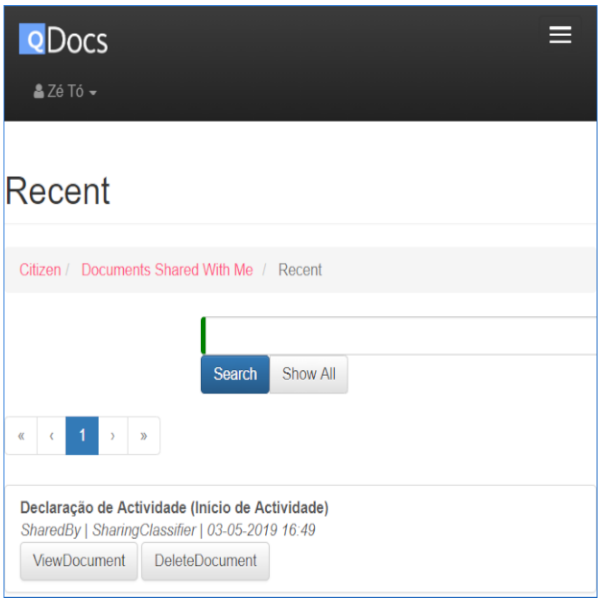


Figure 5.14: My Documents Interface

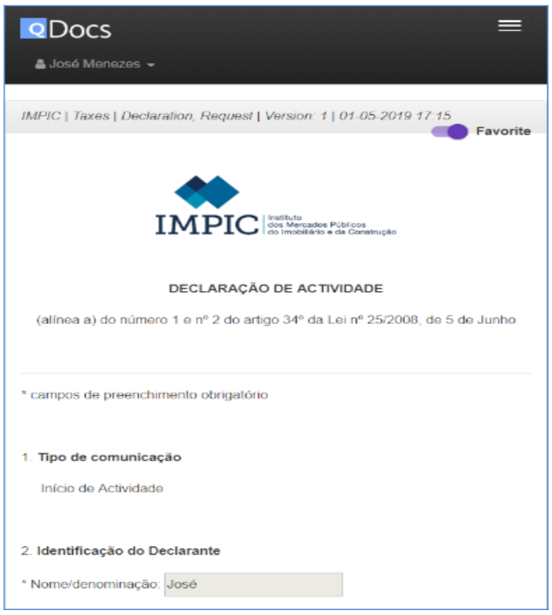


Declaração de Actividade (Inicio de Actividade)
SharedBy | SharingClassifier | 03-05-2019 16:49

Figure 5.15: Documents Shared With Me Interface

Figure 5.14 presents the My Documents interface. Through this interface a citizen create a new document or access its documents through different filters, including Life Events, Document Types and Organizations (i.e. curators). Before this work, accessing a document through a filter was not done properly, e.g. a citizen could select every kind of life event despite not having documents with life events associated with it.

Figure 5.15 presents the interface to access a document shared with the citizen, therefore there is an option to delete the document. Citizens can not delete their own created documents, so the interface to access a citizen's own documents is similar but without the delete document button. These interfaces did not have technical implementations but are shown to provide context.



IMPIC | Taxes | Declaration, Request | Version: 1 | 01-05-2019 17:15

Favorite

DECLARAÇÃO DE ACTIVIDADE

(alínea a) do número 1 e nº 2 do artigo 34º da Lei nº 25/2008, de 5 de Junho

* campos de preenchimento obrigatório

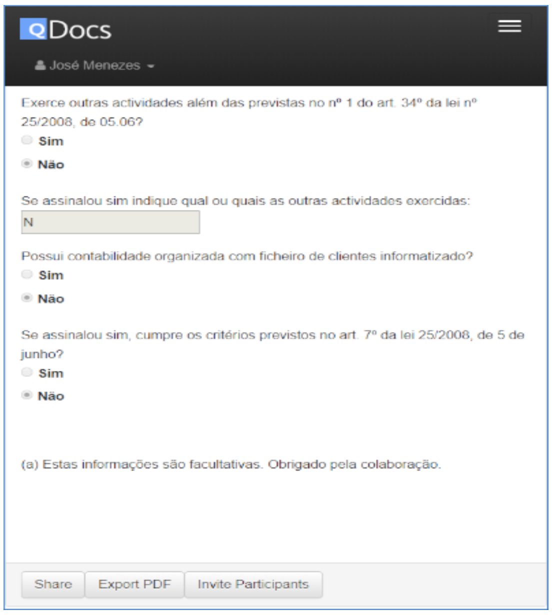
1. **Tipo de comunicação**

   Início de Actividade

2. **Identificação do Declarante**

* Nome/denominação: José

(a)



Exerce outras actividades além das previstas no nº 1 do art. 34º da lei nº 25/2008, de 05.06?

Sim [✓ Não]

Se assinalou sim indique qual ou quais as outras actividades exercidas: N

Possui contabilidade organizada com ficheiro de clientes informatizado?

Sim [✓ Não]

Se assinalou sim, cumpre os critérios previstos no art. 7º da lei 25/2008, de 5 de junho?

Sim [✓ Não]

(a) Estas informações são facultativas. Obrigado pela colaboração.

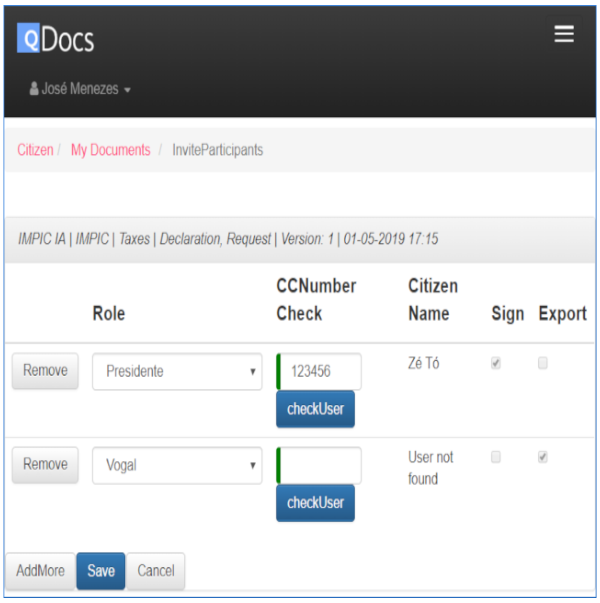
(b)

Figure 5.16: Document Interface

Figure 5.16 shows an example of a citizen's document. Figure 5.16(b) includes the buttons that provide the features available for the document. As explained before, the available features for a document are configured by the curator Templates Editor. For example, if the Templates Editor did not enable the share feature, that button would not appear in the document interface. Before this work, a citizen would always have access to all document features.

Figure 5.17 presents the Invite Participants interface. In this interface, a user can configure the participants of its document, i.e. assigns a role (e.g. President) and provides the CC Number of the citizen user that will have the assigned role. The citizen can see the features available for each participant role, e.g. in the figure the participant "Presidente" can sign the document and the "Vogal" can export it. For participants to have access to the document, the owner of the document must use the share feature and provide the key or hyperlink to the respective citizen users. Then these users can access the document and see an interface similar with the one presented in Figure 5.18, which has the Sign button available in this case. With this work, the described interfaces and application logic were created and implemented but the participants' features were not, only the button availability.

Figure 5.19 presents the user profile interface. In this interface a user can change its information (alias, email, phone number and password) and verify its associations with curators. A user can not change its CC Number because in order to create an account, its CC Number must be verified.



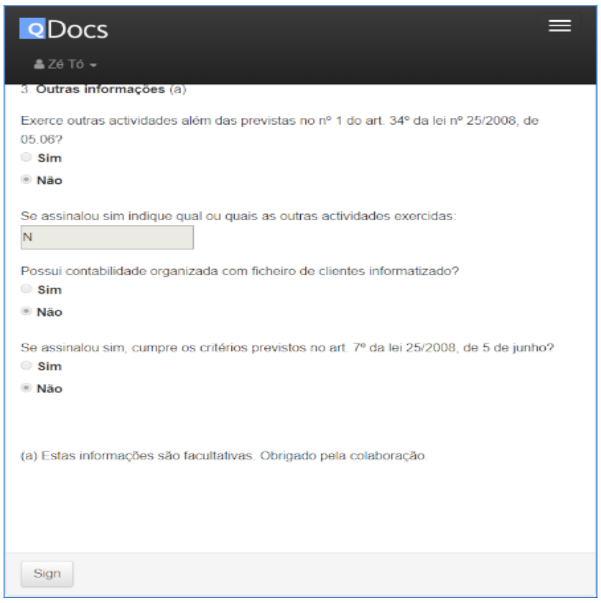
Citizen / My Documents / InviteParticipants

IMPIC IA | IMPIC | Taxes | Declaration, Request | Version: 1 | 01-05-2019 17:15

	Role	CCNumber Check	Citizen Name	Sign	Export
Remove	Presidente	123456 checkUser	Zé Tó	✓	
Remove	Vogal	checkUser	User not found		✓

AddMore Save Cancel

Figure 5.17: Invite Participants Interface



3. **Outras informações** (a)

Exerce outras actividades além das previstas no nº 1 do art. 34º da lei nº 25/2008, de 05.06?
Sim [✓ Não]

Se assinalou sim indique qual ou quais as outras actividades exercidas:
N

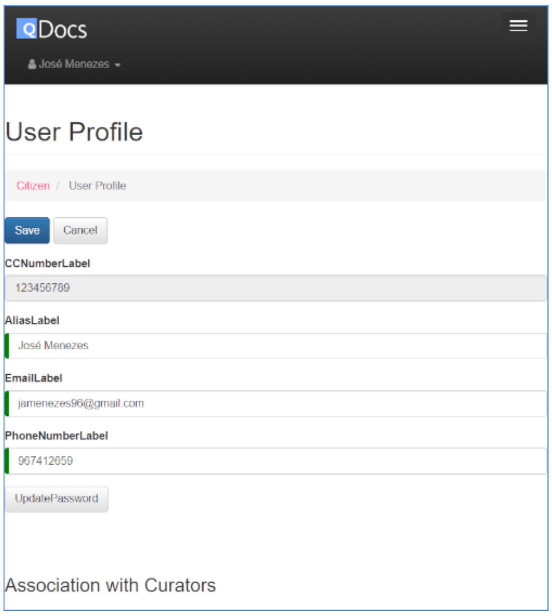
Possui contabilidade organizada com ficheiro de clientes informatizado?
Sim [✓ Não]

Se assinalou sim, cumpre os critérios previstos no art. 7º da lei 25/2008, de 5 de junho?
Sim [✓ Não]

(a) Estas informações são facultativas. Obrigado pela colaboração.

Sign

Figure 5.18: Shared Document Interface



User Profile

Citizen / User Profile

Save Cancel

CCNumberLabel
123456789

AliasLabel
José Menezes

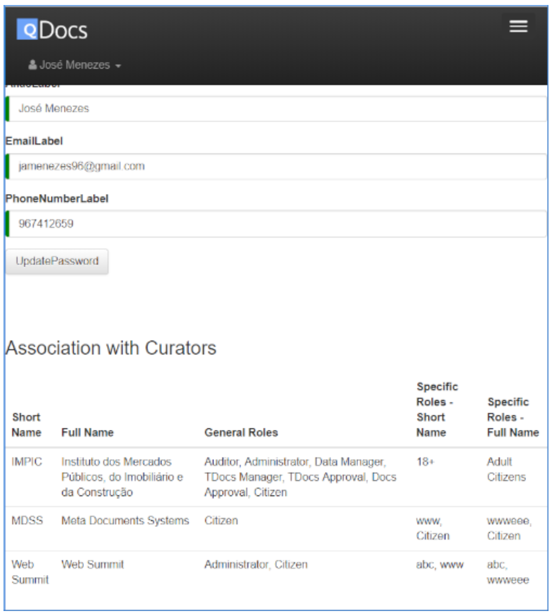
EmailLabel
jamenezes96@gmail.com

PhoneNumberLabel
967412659

UpdatePassword

Association with Curators

(a)



José Menezes

EmailLabel
jamenezes96@gmail.com

PhoneNumberLabel
967412659

UpdatePassword

Association with Curators

Short Name	Full Name	General Roles	Specific Roles - Short Name	Specific Roles - Full Name
IMPIC	Instituto dos Mercados Públicos, do Imobiliário e da Construção	Auditor, Administrator, Data Manager, TDocs Manager, TDocs Approval, Docs Approval, Citizen	18+	Adult Citizens
MDSS	Meta Documents Systems	Citizen	www, Citizen	wwweee, Citizen
Web Summit	Web Summit	Administrator, Citizen	abc, www	abc, wwweee

(b)

Figure 5.19: Citizen Profile Interface

5.2.4 Cross-Cutting

During this work, all interfaces newly created kept the visual style that was already implemented before this work. In addition the following cross-cutting technical implementations were made:

- **Search Bars** and **Pagination** — all list interfaces did not have these features; they make the search for items in a list easier;
- **Navigation** — in some interfaces navigation was not flexible or not properly done, e.g. it was not possible to navigate from the template edit metadata interface to the template edit content interface directly and vice-versa;
- **Curator Name** — before this work, all curator interfaces did not have the respective curator's name on them; this was implemented to provide context to the curator user;
- **Authorization** — before this work, when a citizen was associated to a curator it was allowed to access the curator's interface, however it could not do anything.

6 Evaluation

Contents

This chapter presents two scenarios used to test the implementations done in the qDocs platform. Both scenarios present the life-cycle of a document in a fictional academic context. Section 6.1 presents the Scenario – A focused on the following aspects:

- Template metadata configuration, manually use of payment and strong authentication features;
- Access to a created document through different filtering options;
- Access to the created document and check the available features.

Section 6.2 presents the Scenario – B focused on the following aspects of the qDocs application:

- Template design;
- Participant configuration;
- Document creation;
- Access to the created document;
- Participant invitation;
- Access to the document by a participant.

Finally, Section 6.3 discusses the presented scenarios and compares the qDocs application with the related work.

6.1 Scenario – A

Consider a master degree certificate of qualifications from a student of Instituto Superior Técnico, a certificate that proves the student completed a master degree.

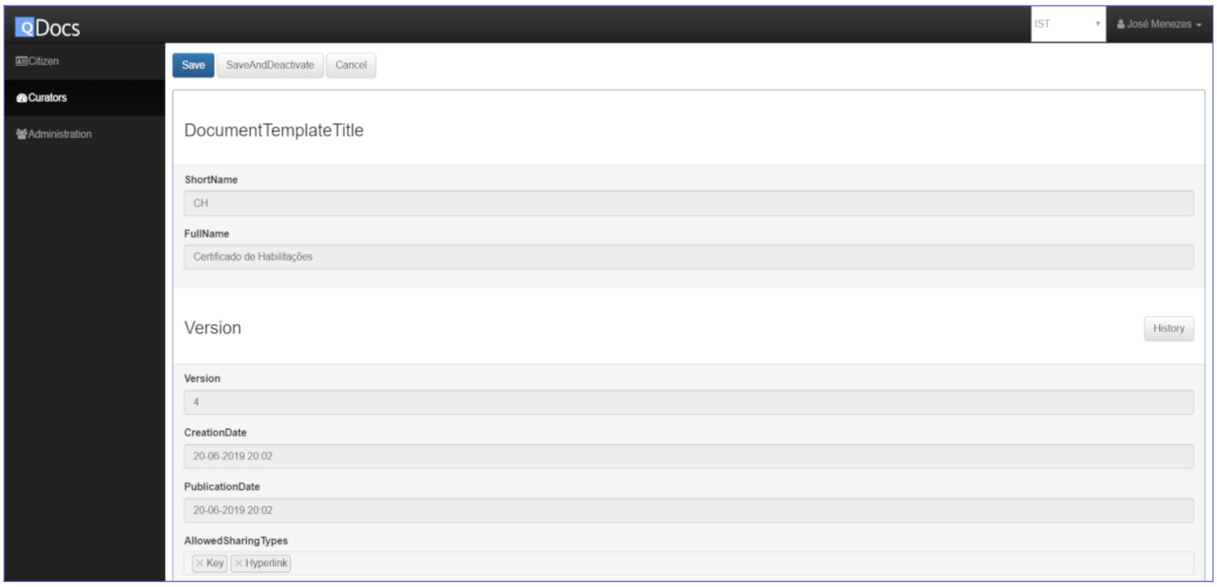
6.1.1 Curator-level

Before the certificate is available in the qDocs/Citizen application, a Templates Editor must design the corresponding template, configure its metadata. Figure 6.1 presents the form to configure and define the metadata for the ”Certificado de Habilitações” template. From the properties presented in Figure 6.1(a) only the ”AllowSharingTypes” can be edited. This parameter allows the Templates Editor to configure if the template can be shared via hyperlink or key. In this scenario, the certificate can be shared via hyperlink and key. Since the template is on version 4, the Templates Editor can check the template's history if he wants. The ”CreationDate” refers to the date of creation of the current template's version and the ”PublicationDate” refers to the date of activation of the current template's version. Figure 6.1(b) shows the rest of the metadata form. The ”AllowedFeatures” of a template are:

- Export — the citizen can export the document, e.g. as a PDF file format;
- Strong Authentication — to access the document, a citizen must go through a Strong Authentication process (the implementation of this process is scheduled for future work);
- Payment — to access the document, a citizen must pay the value that is defined in the ”Payment-Value” parameter (the implementation of this process is scheduled for future work).

The parameters inside ”DocumentGroups” (i.e. ”LifeEvents”, ”DocumentTypes” and ”Organization-Groups”) are classifiers that may help citizens to search their documents. In this scenario, the values for these parameters are the following: (a) Education (Life Event); (b) Declaration (Document Type); (c) Certificado (Organization Group, also named document group in the curator interface). A “User-Group” (also known as Curator Specific Role) can also be selected and if that is the case, only citizens with these roles can access that document. “UserGroups” are also set by the Curator Administrator, as stated before. The value for this parameter is ”Finalista”, which means only citizen students who have finished their degrees may create this document. Templates can be defined as an internal or external scope: if it is internal, only users associated with the Curator can access its documents, e.g. a student from university X can not access documents from university Y; if it external, the document is available for any citizen user.

After designing the template and defining its metadata, it is activated, therefore it is ready to be created in the qDocs/Citizen application.



DocumentTemplateTitle

ShortName: CH

FullName: Certificado de Habilitações

Version

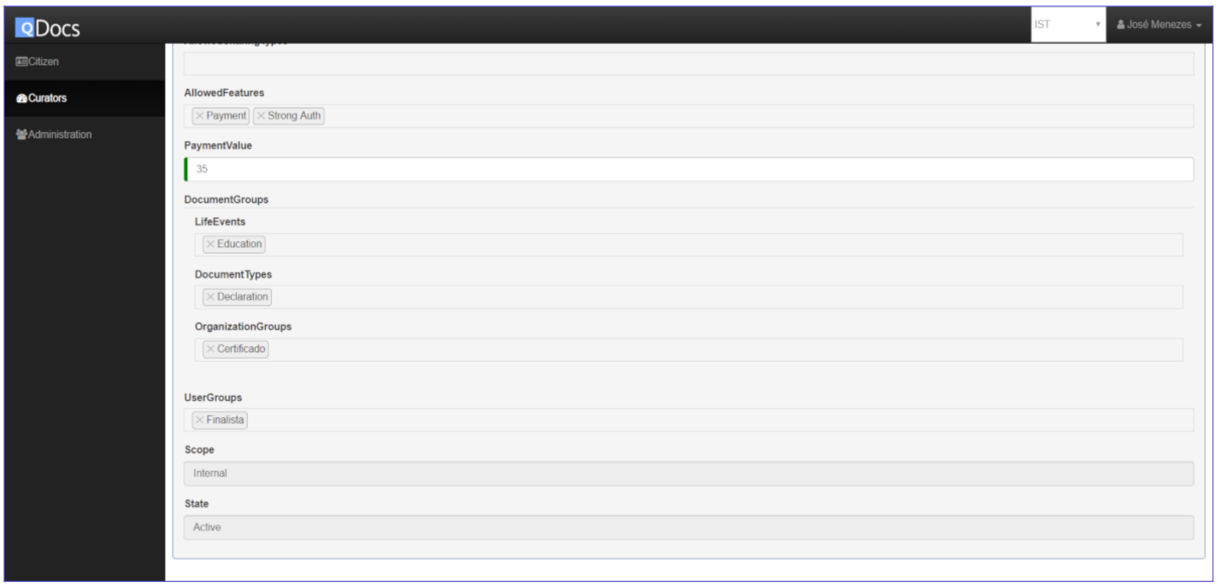
Version: 4

CreationDate: 20-06-2019 20:02

PublicationDate: 20-06-2019 20:02

AllowedSharingTypes: Key, Hyperlink

(a)



AllowedFeatures: Payment, Strong Auth

PaymentValue: 35

DocumentGroups

LifeEvents: Education

DocumentTypes: Declaration

OrganizationGroups: Certificado

UserGroups: Finalista

Scope: Internal

State: Active

(b)

Figure 6.1: Template Metadata

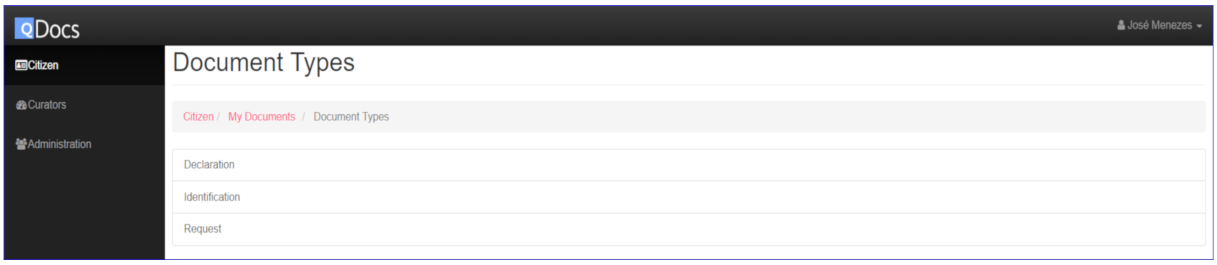
6.1.2 Citizen-level

For this scenario, the following would happen: (a) The student requests the creation of his ”Certificado de Habilitações”; (b) that request is approved by the Documents Manager of Instituto Superior Técnico; (c) the student now has access to the document in his qDocs/Citizen application. This workflow is scheduled for future work. With this work, the student would be the one creating his own ”Certificado de Habilitações”.

After the document is created, the student can access it using the classifiers that were set by the Templates Editor when he configured the template's metadata. Figures 6.2, 6.3 and 6.4 present the sequences of interfaces the student sees when he wants to access its ”Certificado de Habilitações” using the different classifiers. Figure 6.2 presents the access to the document through the ”Document Type” classifier, Figure 6.3 presents the access to the document through the ”Life Event” classifier and Figure 6.4 presents the access to the document through the ”Organization Groups” classifier.

After accessing the document through any of the specified classifiers, the student must pay for the document and proceed through a Strong Authentication process before visualizing the document, because that was specified in the template's metadata. This process is represented in Figures 6.5(a), 6.5(b) and 6.5(c). The implementation of the Payment and Strong Authentication features are scheduled for future work, since the focus of this work was to implement the functionality in the curator side of the qDocs platform. Figure 6.5(d) presents the end of the visualization of the ”Certificado de Habilitações” with the ”Share” button visible, because the Templates Editor only enabled that feature.

The student only has to pay for the document for the first time he wants to access it, but he has to go through the Strong Authentication process every time he tries to access the document.



Document Types

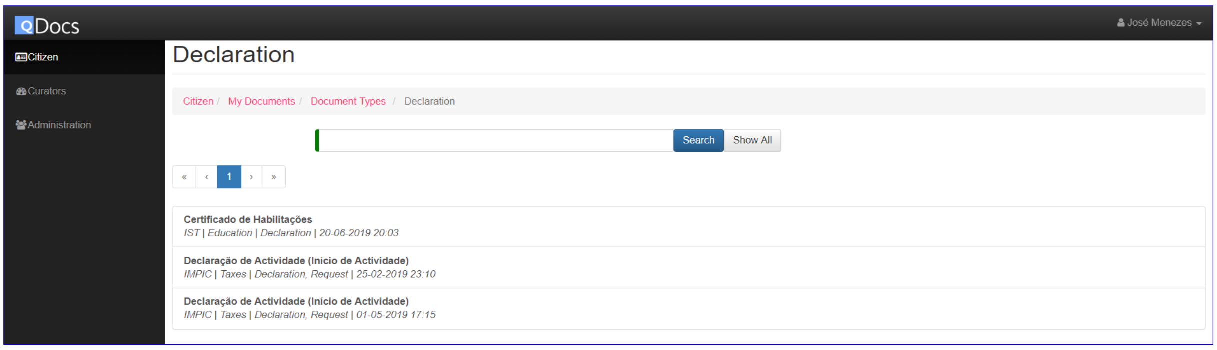
Citizen / My Documents / Document Types

Declaration

Identification

Request

(a)



Declaration

Citizen / My Documents / Document Types / Declaration

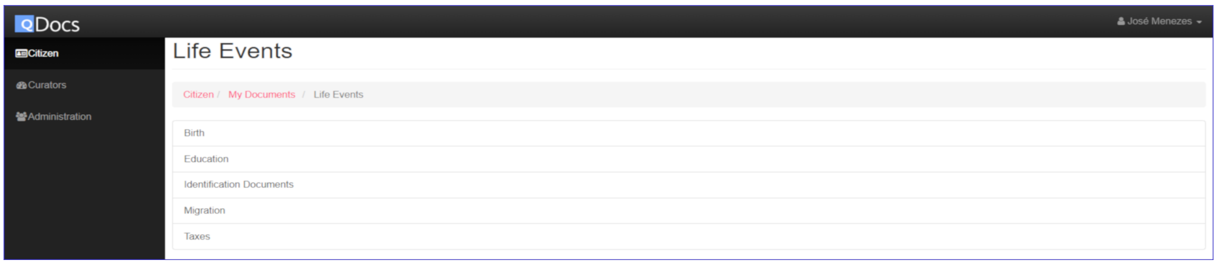
Certificado de Habilitações
IST | Education | Declaration | 20-06-2019 20:03

Declaração de Actividade (Inicio de Actividade)
IMPIC | Taxes | Declaration, Request | 25-02-2019 23:10

Declaração de Actividade (Inicio de Actividade)
IMPIC | Taxes | Declaration, Request | 01-05-2019 17:15

(b)

Figure 6.2: Document Access — Document Types



Life Events

Citizen / My Documents / Life Events

Birth

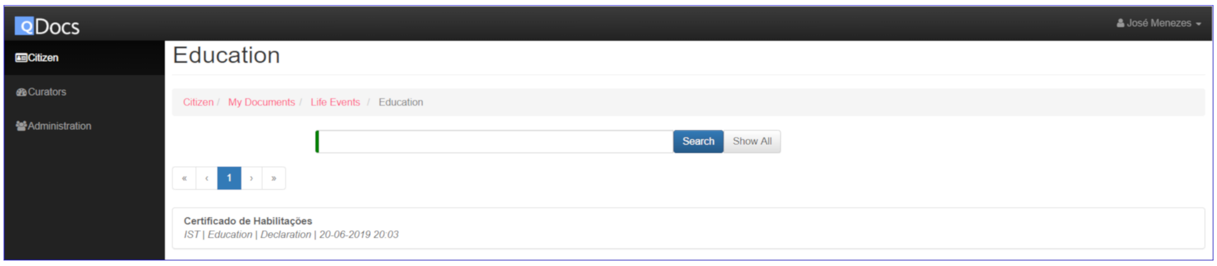
Education

Identification Documents

Migration

Taxes

(a)



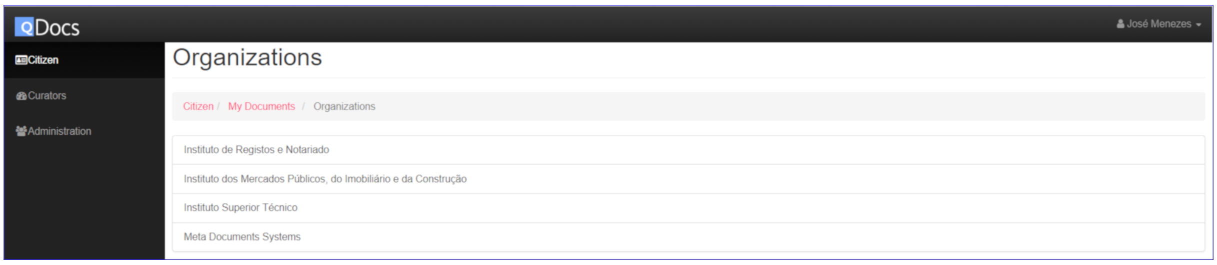
Education

Citizen / My Documents / Life Events / Education

Certificado de Habilitações
IST | Education | Declaration | 20-06-2019 20:03

(b)

Figure 6.3: Document Access — Life Events



Organizations

Citizen / My Documents / Organizations

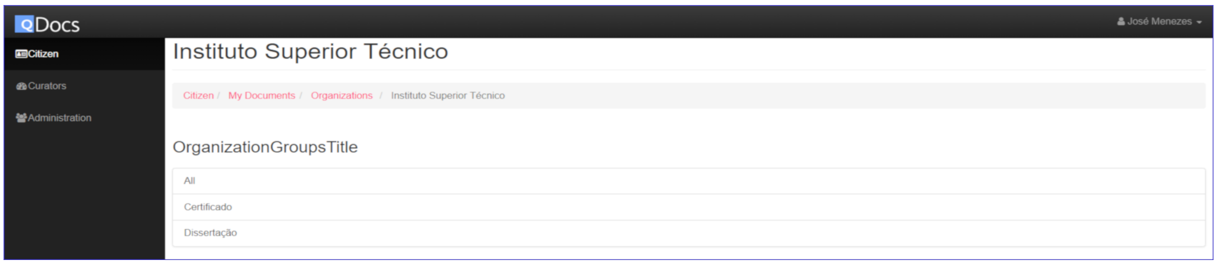
Instituto de Registos e Notariado

Instituto dos Mercados Públicos, do Imobiliário e da Construção

Instituto Superior Técnico

Meta Documents Systems

(a)



Instituto Superior Técnico

Citizen / My Documents / Organizations / Instituto Superior Técnico

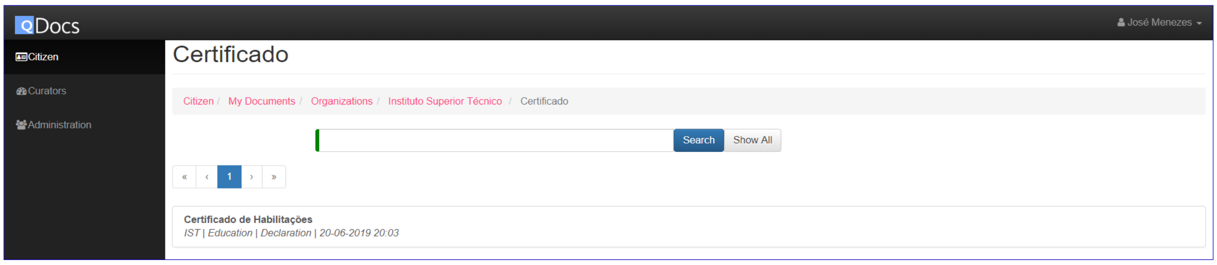
OrganizationGroupsTitle

All

Certificado

Dissertação

(b)



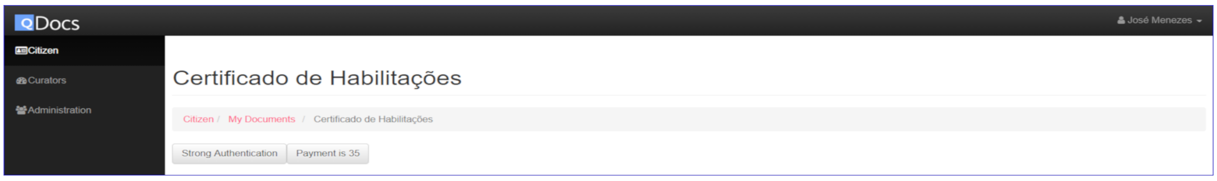
Certificado

Citizen / My Documents / Organizations / Instituto Superior Técnico / Certificado

Certificado de Habilitações
IST | Education | Declaration | 20-06-2019 20:03

(c)

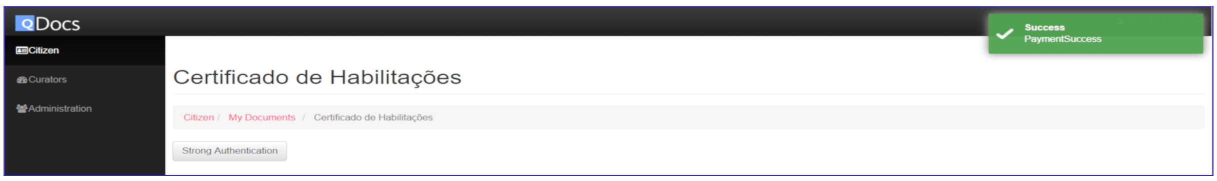
Figure 6.4: Document Access — Organizations



Certificado de Habilitações

Citizen / My Documents / Certificado de Habilitações

(a)

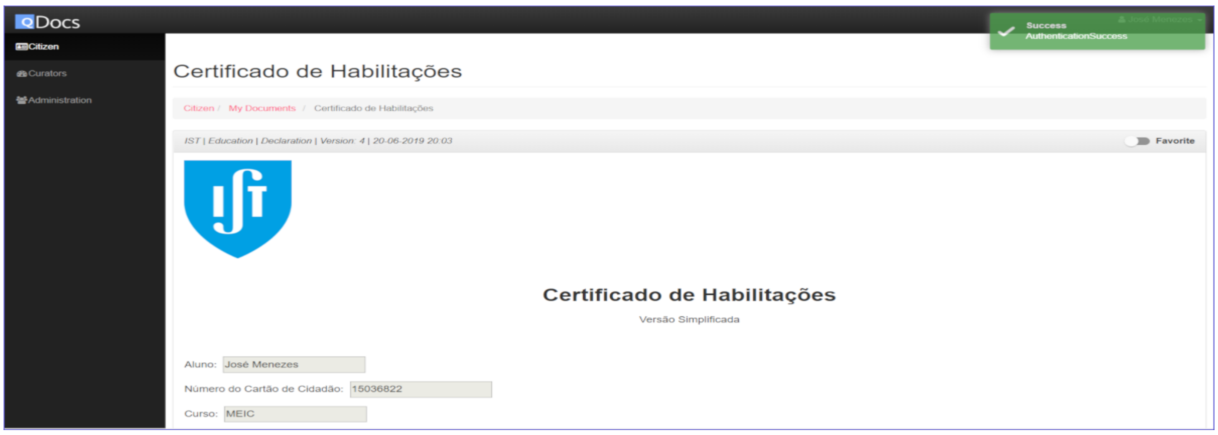


Success
PaymentSuccess

Certificado de Habilitações

Citizen / My Documents / Certificado de Habilitações

(b)



Success
AuthenticationSuccess

Certificado de Habilitações

Citizen / My Documents / Certificado de Habilitações

IST | Education | Declaration | Version: 4 | 20-06-2019 20:03

Certificado de Habilitações

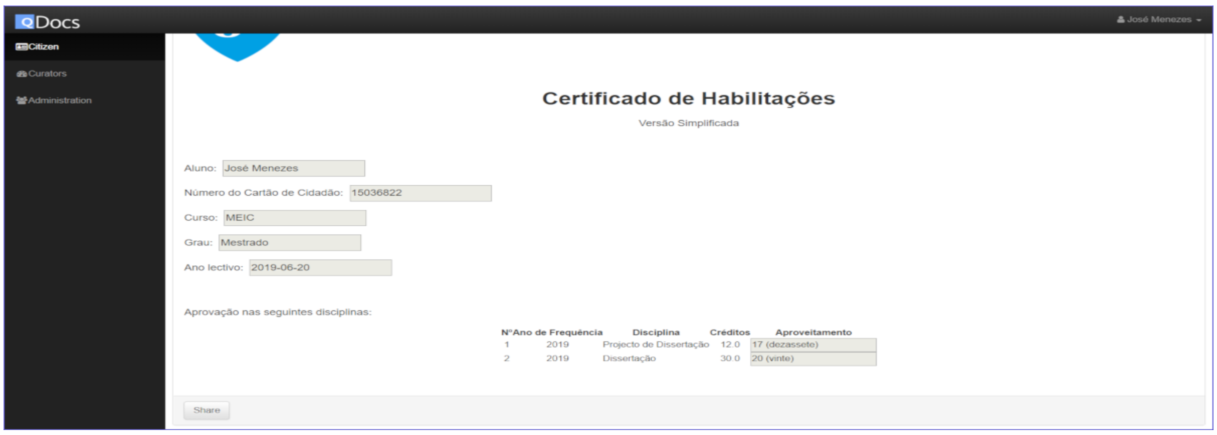
Versão Simplificada

Aluno: José Menezes

Número do Cartão de Cidadão: 15036822

Curso: MEIC

(c)



Certificado de Habilitações

Versão Simplificada

Aluno: José Menezes

Número do Cartão de Cidadão: 15036822

Curso: MEIC

Grau: Mestrado

Ano lectivo: 2019-06-20

Aprovação nas seguintes disciplinas:

Nº	Ano de Frequência	Disciplina	Créditos	Aproveitamento
1	2019	Projecto de Dissertação	12.0	17 (dezassete)
2	2019	Dissertação	30.0	20 (vinte)

(d)

Figure 6.5: Document Visualization

6.2 Scenario – B

Consider a minute of the dissertation defense meeting of Instituto Superior Técnico, a document used to evaluate a student's dissertation after its defense meeting.

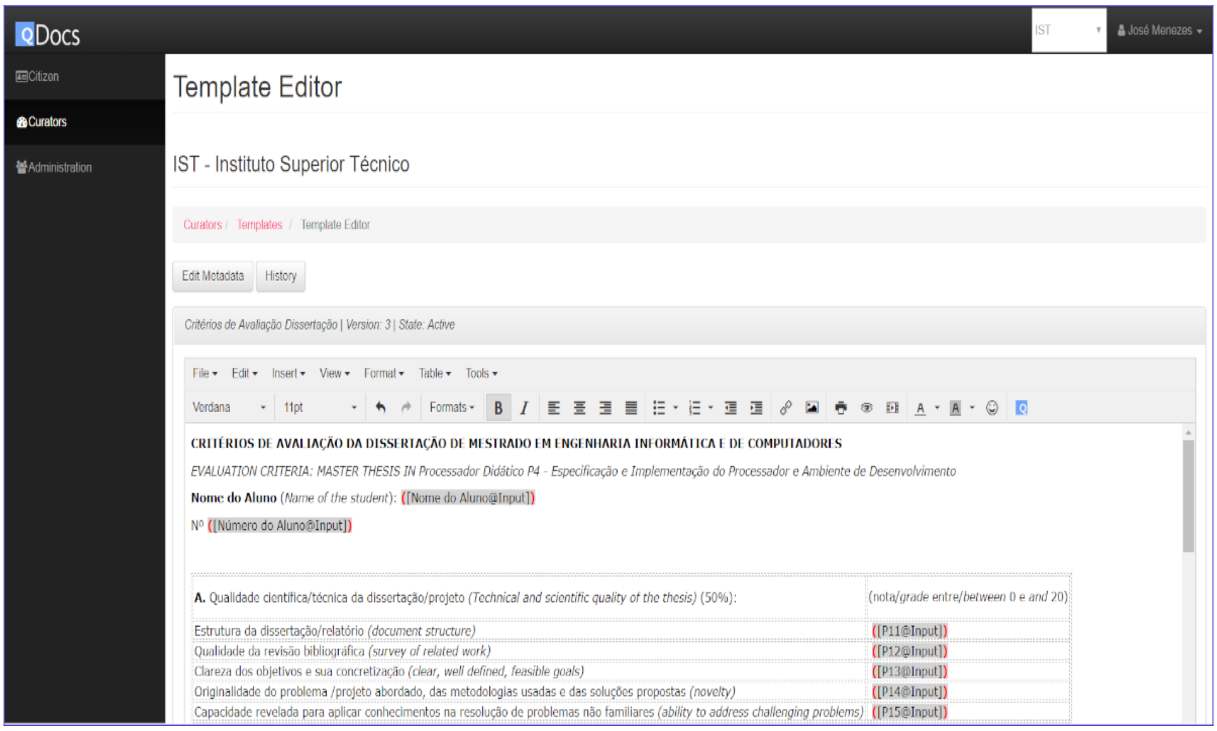
6.2.1 Curator-level

Before it is available in the qDocs/Citizen application, a Templates Editor has to design the template for the minute. Figures 6.6 and 6.7 show how the template editor of the qDocs/Curator application looks just before saving the template. In the editor, **form objects** are inserted using the blue button with the letter ”Q” and are represented by their name followed by the ”@” symbol and by the ”Input” or ”Output” words, meaning the form object is of type ”Input” (i.e. the citizen must insert some information when creating the document and then it will be stored in the Curator's database when the document is generated) or ”Output” (i.e. the created document is already filled with information from the respective Curator), respectively. This information is between brackets and red parentheses as shown in the figures. In this minute the form objects are:

- ([Nome do Aluno@Input]) — the evaluated student's name;
- ([Número do Aluno@Input]) — the evaluated student's faculty number;
- ([P11@Input]) to ([P28@Input]) — the evaluated student's score in each parameter; the score may have the values **NA** (Not applicable), **Mau** (Poor), **Medíocre** (Mediocre), **Suficiente** (Fair), **Bom** (Good) or **Muito Bom** (Very Good);
- ([Nota@Input]) — the evaluated student's dissertation grade;
- ([Justificacao@Input]) — the justification of the student's grade;
- ([Unanimidade@Input]) — the unanimity, i.e. if the dissertation's grade was a unanimous decision; the values of this form object are **Yes** or **No**;
- ([Data@Input]) — the date of the dissertation's defense meeting.

After designing the template, the Templates Editor configures the participants of the document as shown in Figure 6.8. In this case, there are 4 participants and they all can Sign and Export the document, since it is a minute from the faculty. The participants of this document have different multiplicity rules, because to evaluate a dissertation it is mandatory to have one ”Presidente”, one ”Orientador”, zero or one ”Co-Orientador” and zero or more ”Vogal”. When the Templates Editor finishes the configuration it saves it.

After designing the document template and configuring its participants, the template's metadata is set and then the template is activated, making the document ready to be created in the qDocs/Citizen platform.



Template Editor

IST - Instituto Superior Técnico

Critérios de Avaliação Dissertação | Version: 3 | State: Active

CRITÉRIOS DE AVALIAÇÃO DA DISSERTAÇÃO DE MESTRADO EM ENGENHARIA INFORMÁTICA E DE COMPUTADORES

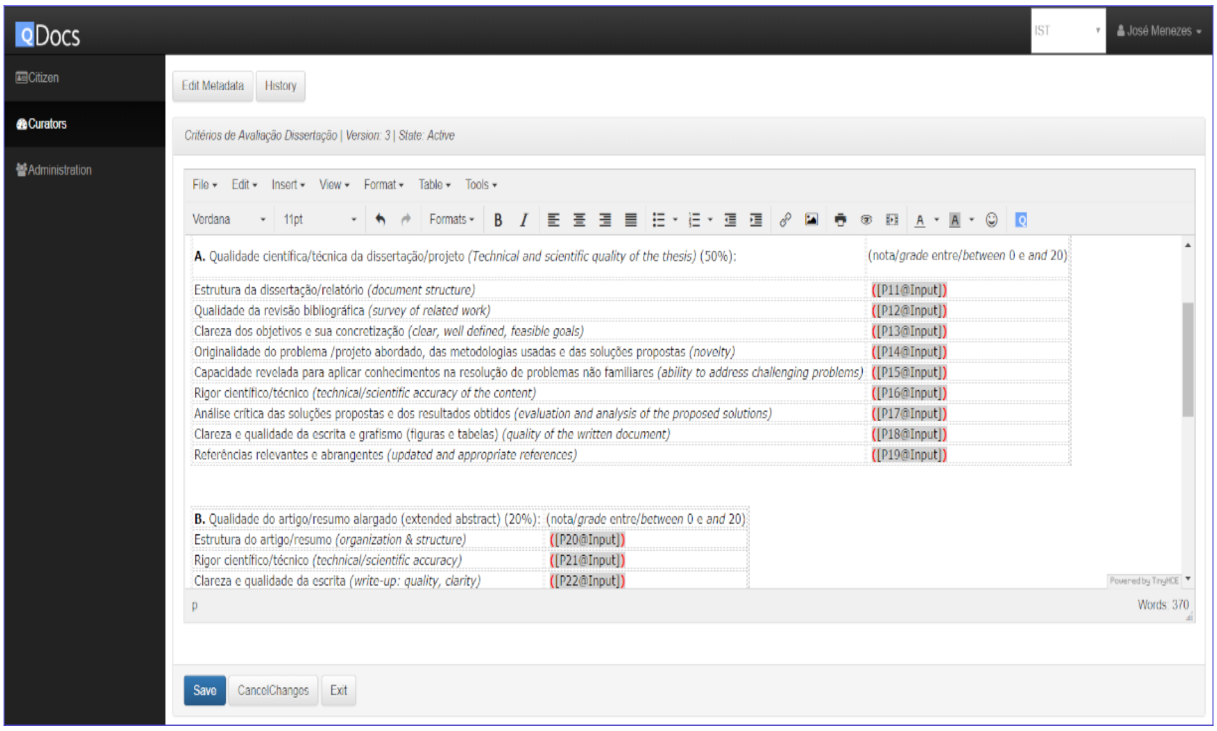
EVALUATION CRITERIA: MASTER THESIS IN Processador Didático P4 - Especificação e Implementação do Processador e Ambiente de Desenvolvimento

Nome do Aluno (*Name of the student*): ([Nome do Aluno@Input])

Nº ([Número do Aluno@Input])

A. Qualidade científica/técnica da dissertação/projeto *(Technical and scientific quality of the thesis)* (50%):	(nota/*grade* entre/*between* 0 e *and* 20)
Estrutura da dissertação/relatório *(document structure)*	([P11@Input])
Qualidade da revisão bibliográfica *(survey of related work)*	([P12@Input])
Clareza dos objetivos e sua concretização *(clear, well defined, feasible goals)*	([P13@Input])
Originalidade do problema /projeto abordado, das metodologias usadas e das soluções propostas *(novelty)*	([P14@Input])
Capacidade revelada para aplicar conhecimentos na resolução de problemas não familiares *(ability to address challenging problems)*	([P15@Input])

(a)



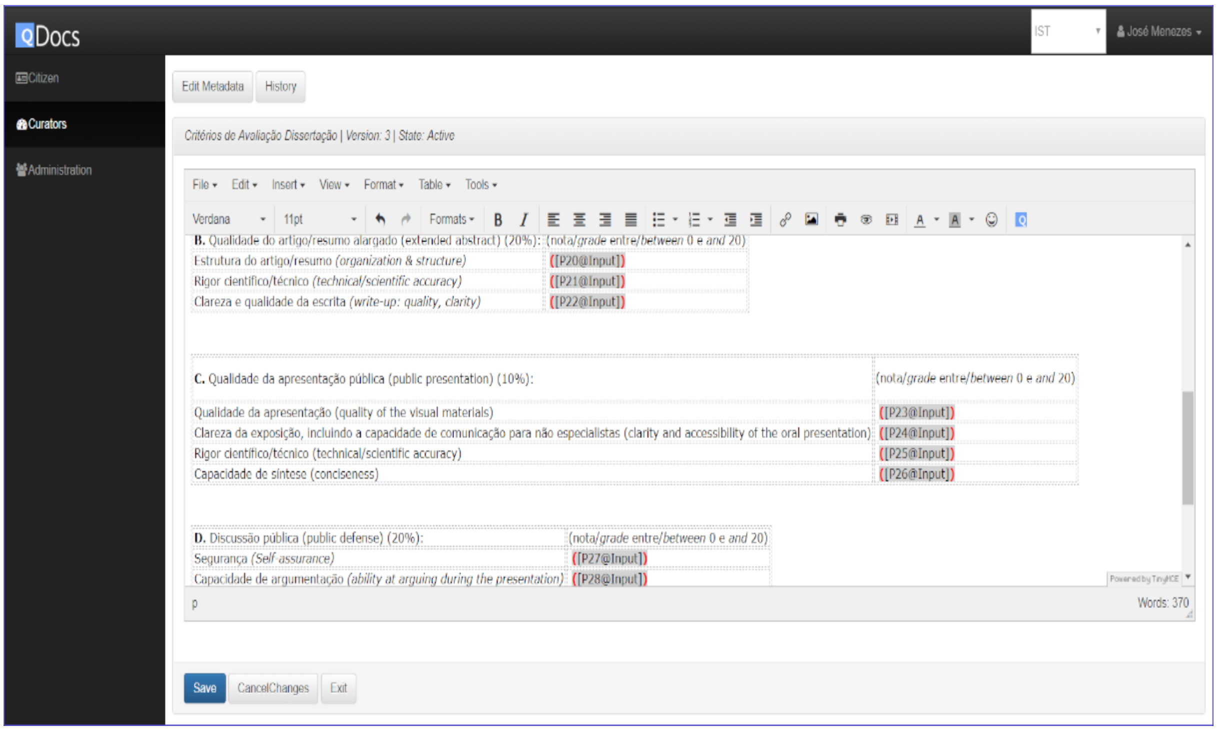
Critérios de Avaliação Dissertação | Version: 3 | State: Active

A. Qualidade científica/técnica da dissertação/projeto *(Technical and scientific quality of the thesis)* (50%):	(nota/*grade* entre/*between* 0 e *and* 20)
Estrutura da dissertação/relatório *(document structure)*	([P11@Input])
Qualidade da revisão bibliográfica *(survey of related work)*	([P12@Input])
Clareza dos objetivos e sua concretização *(clear, well defined, feasible goals)*	([P13@Input])
Originalidade do problema /projeto abordado, das metodologias usadas e das soluções propostas *(novelty)*	([P14@Input])
Capacidade revelada para aplicar conhecimentos na resolução de problemas não familiares *(ability to address challenging problems)*	([P15@Input])
Rigor científico/técnico *(technical/scientific accuracy of the content)*	([P16@Input])
Análise crítica das soluções propostas e dos resultados obtidos *(evaluation and analysis of the proposed solutions)*	([P17@Input])
Clareza e qualidade da escrita e grafismo (figuras e tabelas) *(quality of the written document)*	([P18@Input])
Referências relevantes e abrangentes *(updated and appropriate references)*	([P19@Input])

B. Qualidade do artigo/resumo alargado (extended abstract) (20%):	(nota/*grade* entre/*between* 0 e *and* 20)
Estrutura do artigo/resumo *(organization & structure)*	([P20@Input])
Rigor científico/técnico *(technical/scientific accuracy)*	([P21@Input])
Clareza e qualidade da escrita *(write-up: quality, clarity)*	([P22@Input])

(b)

Figure 6.6: Template Creation Part 1



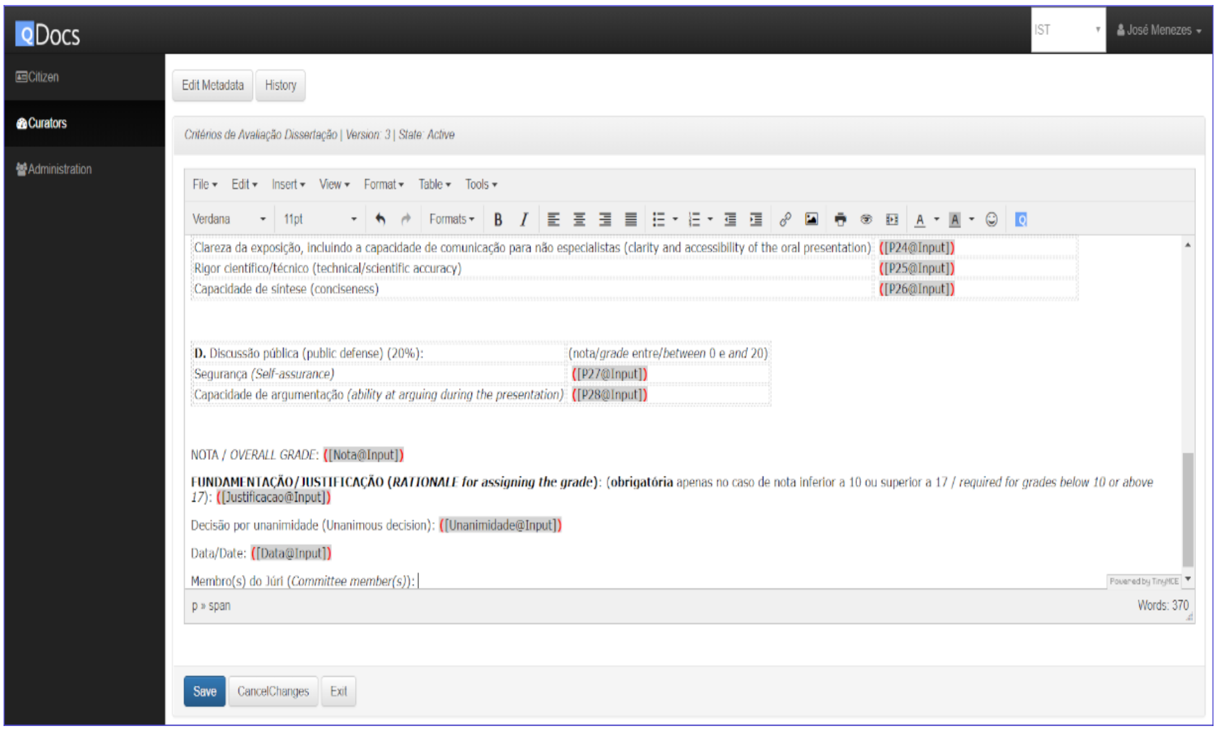
Critérios de Avaliação Dissertação | Version: 3 | State: Active

B. Qualidade do artigo/resumo alargado (extended abstract) (20%):	(nota/*grade* entre/*between* 0 e *and* 20)
Estrutura do artigo/resumo *(organization & structure)*	([P20@Input])
Rigor científico/técnico *(technical/scientific accuracy)*	([P21@Input])
Clareza e qualidade da escrita *(write-up: quality, clarity)*	([P22@Input])

C. Qualidade da apresentação pública (public presentation) (10%):	(nota/*grade* entre/*between* 0 e *and* 20)
Qualidade da apresentação (quality of the visual materials)	([P23@Input])
Clareza da exposição, incluindo a capacidade de comunicação para não especialistas (clarity and accessibility of the oral presentation)	([P24@Input])
Rigor científico/técnico (technical/scientific accuracy)	([P25@Input])
Capacidade de síntese (conciseness)	([P26@Input])

D. Discussão pública (public defense) (20%):	(nota/*grade* entre/*between* 0 e *and* 20)
Segurança *(Self-assurance)*	([P27@Input])
Capacidade de argumentação *(ability at arguing during the presentation)*	([P28@Input])

(a)



Critérios de Avaliação Dissertação | Version: 3 | State: Active

Clareza da exposição, incluindo a capacidade de comunicação para não especialistas (clarity and accessibility of the oral presentation)	([P24@Input])
Rigor científico/técnico (technical/scientific accuracy)	([P25@Input])
Capacidade de síntese (conciseness)	([P26@Input])

D. Discussão pública (public defense) (20%):	(nota/*grade* entre/*between* 0 e *and* 20)
Segurança *(Self-assurance)*	([P27@Input])
Capacidade de argumentação *(ability at arguing during the presentation)*	([P28@Input])

NOTA / *OVERALL GRADE*: ([Nota@Input])

FUNDAMENTAÇÃO/JUSTIFICAÇÃO (*RATIONALE for assigning the grade*): (**obrigatória** apenas no caso de nota inferior a 10 ou superior a 17 / *required for grades below 10 or above 17*): ([Justificacao@Input])

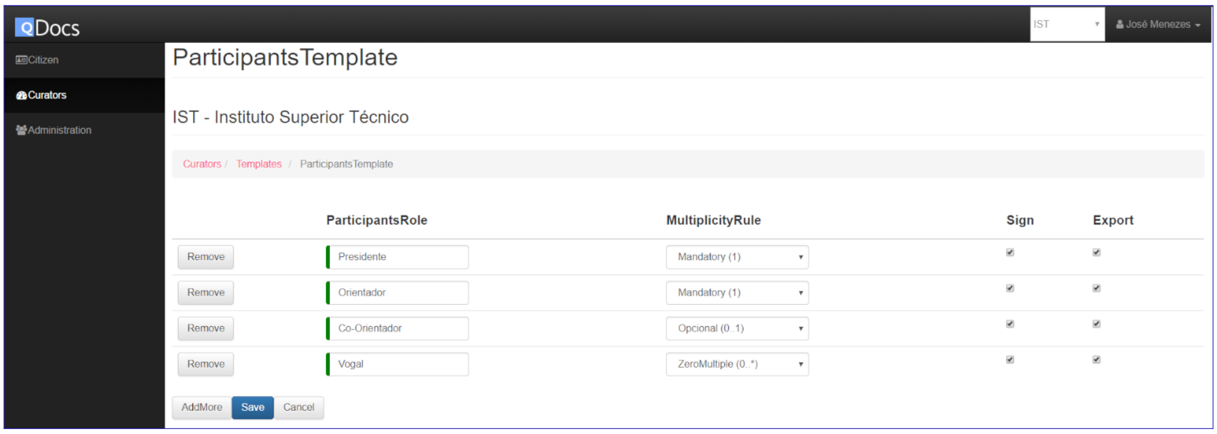
Decisão por unanimidade (Unanimous decision): ([Unanimidade@Input])

Data/Date: ([Data@Input])

Membro(s) do Júri (*Committee member(s)*):

(b)

Figure 6.7: Template Creation Part 2



ParticipantsTemplate

IST - Instituto Superior Técnico

Curators / Templates / ParticipantsTemplate

	ParticipantsRole	MultiplicityRule	Sign	Export
Remove	Presidente	Mandatory (1)	✓	✓
Remove	Orientador	Mandatory (1)	✓	✓
Remove	Co-Orientador	Opcional (0..1)	✓	✓
Remove	Vogal	ZeroMultiple (0..*)	✓	✓

Figure 6.8: Participant Configuration

6.2.2 Citizen-level

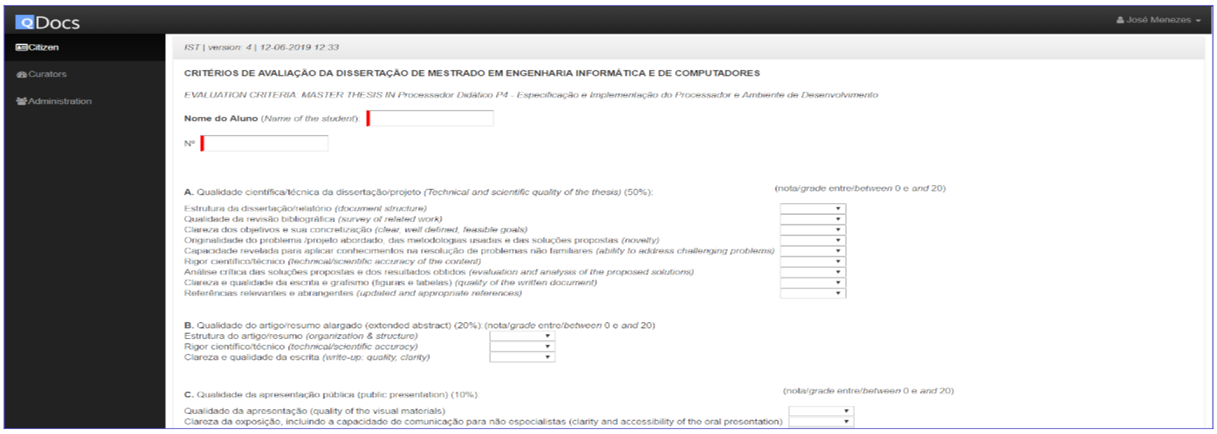
Figure 6.9 shows the interface presented to the user that will generate the minute of the dissertation's defense meeting. The boxes to be filled refer to the form objects inserted in the template. Theoretically, in this case, the user that would create the document would not fill it, since it is the participant with the role ”Presidente” the responsible for doing it, but with this work the user who creates the document is the one who should fill it. The filling of documents by participants is scheduled for future work.

After filling the boxes, the user then saves the document and it is generated and ready to be consulted. When the user wants to consult the generated document, an interface similar to Figure 6.10 will appear, and now the user has the option to invite the respective participants to the document using the button ”Invite Participants”. This button is available because there are participants that were configured previously.

Figure 6.11 presents the interface where the user invites the respective participants to the document. For citizens to be invited has participants of a document, a user must know their CC Number and check if it correct using the ”checkUser” button. After all participants are set, the user clicks the ”Save” button and then he must share the document with the participants in order for them to access the document. For future work, after the user clicks the ”Save” button, a notification with a key or hyperlink should be sent automatically to the participants of the document.

When participants receive the sharing key or hyperlink to access the document to which they were invited to, they see an interface similar to the one in Figure 6.12. It is similar to the interface of a user accessing its generated document, but the buttons that appear are the ones that were set by the Templates Editor in the participant configuration. These buttons allow the participant to perform actions such as Sign the document or Export it. The figure only shows pop-ups in green with a success message for both Sign and Export because with this work the functionality of these features was not done, but it is scheduled for future work.

After all participants signed the document, the grade of the student's dissertation can be published. The workflow relative to document manipulation by participants (e.g. signing a document) is also scheduled for future work.



IST | version: 4 | 12-06-2019 12:33

CRITÉRIOS DE AVALIAÇÃO DA DISSERTAÇÃO DE MESTRADO EM ENGENHARIA INFORMÁTICA E DE COMPUTADORES

EVALUATION CRITERIA: MASTER THESIS IN Processador Didático P4 - Especificação e Implementação do Processador e Ambiente de Desenvolvimento

Nome do Aluno (*Name of the student*):

Nº

A. Qualidade científica/técnica da dissertação/projeto *(Technical and scientific quality of the thesis)* (50%): (nota/*grade* entre/*between* 0 e *and* 20)

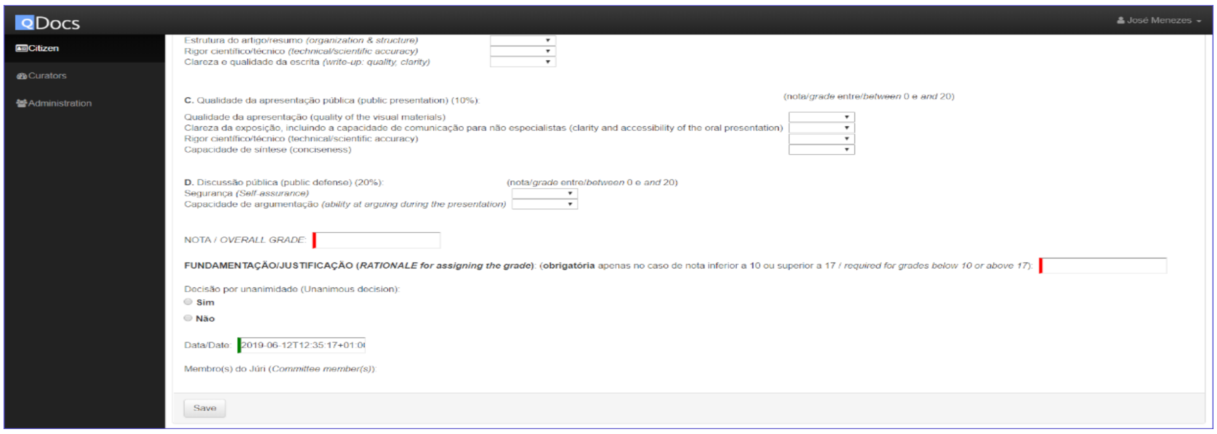
Estrutura da dissertação/relatório *(document structure)*
Qualidade da revisão bibliográfica *(survey of related work)*
Clareza dos objetivos e sua concretização *(clear, well defined, feasible goals)*
Originalidade do problema /projeto abordado, das metodologias usadas e das soluções propostas *(novelty)*
Capacidade revelada para aplicar conhecimentos na resolução de problemas não familiares *(ability to address challenging problems)*
Rigor científico/técnico *(technical/scientific accuracy of the content)*
Análise crítica das soluções propostas e dos resultados obtidos *(evaluation and analysis of the proposed solutions)*
Clareza e qualidade da escrita e grafismo (figuras e tabelas) *(quality of the written document)*
Referências relevantes e abrangentes *(updated and appropriate references)*

B. Qualidade do artigo/resumo alargado (extended abstract) (20%): (nota/*grade* entre/*between* 0 e *and* 20)
Estrutura do artigo/resumo *(organization & structure)*
Rigor científico/técnico *(technical/scientific accuracy)*
Clareza e qualidade da escrita *(write-up: quality, clarity)*

C. Qualidade da apresentação pública (public presentation) (10%): (nota/*grade* entre/*between* 0 e *and* 20)

Qualidade da apresentação (quality of the visual materials)
Clareza da exposição, incluindo a capacidade de comunicação para não especialistas (clarity and accessibility of the oral presentation)

(a)



Estrutura do artigo/resumo *(organization & structure)*
Rigor científico/técnico *(technical/scientific accuracy)*
Clareza e qualidade da escrita *(write-up: quality, clarity)*

C. Qualidade da apresentação pública (public presentation) (10%): (nota/*grade* entre/*between* 0 e *and* 20)

Qualidade da apresentação (quality of the visual materials)
Clareza da exposição, incluindo a capacidade de comunicação para não especialistas (clarity and accessibility of the oral presentation)
Rigor científico/técnico (technical/scientific accuracy)
Capacidade de síntese (conciseness)

D. Discussão pública (public defense) (20%): (nota/*grade* entre/*between* 0 e *and* 20)
Segurança *(Self-assurance)*
Capacidade de argumentação *(ability at arguing during the presentation)*

NOTA / *OVERALL GRADE*:

FUNDAMENTAÇÃO/JUSTIFICAÇÃO (*RATIONALE for assigning the grade*): (**obrigatória** apenas no caso de nota inferior a 10 ou superior a 17 / *required for grades below 10 or above 17*):

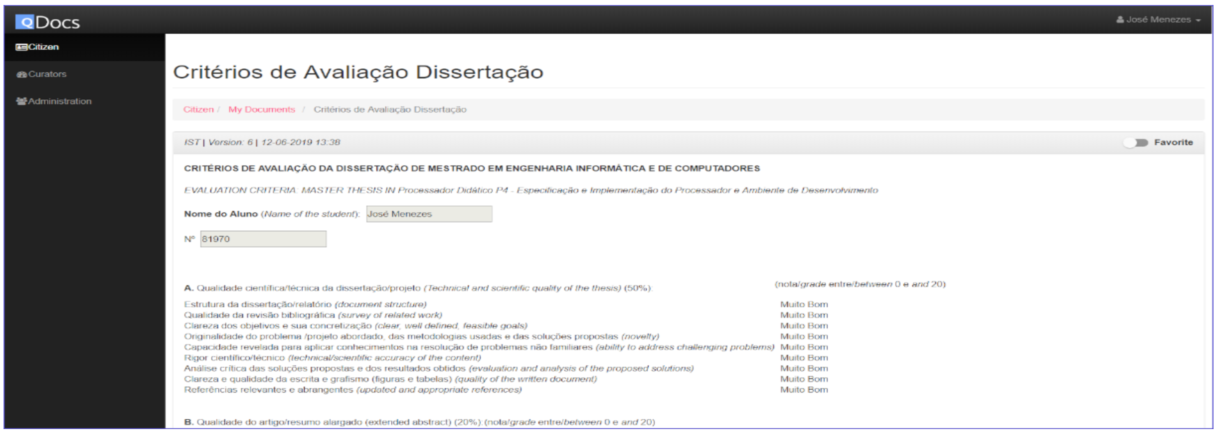
Decisão por unanimidade (Unanimous decision):
○ **Sim**
○ **Não**

Data/Date: 2019-06-12T12:35:17+01:0

Membro(s) do Júri (*Committee member(s)*)

(b)

Figure 6.9: Document Creation



Critérios de Avaliação Dissertação

Citizen / My Documents / Critérios de Avaliação Dissertação

IST | Version: 6 | 12-06-2019 13:38

CRITÉRIOS DE AVALIAÇÃO DA DISSERTAÇÃO DE MESTRADO EM ENGENHARIA INFORMÁTICA E DE COMPUTADORES

EVALUATION CRITERIA: MASTER THESIS IN Processador Didático P4 - Especificação e Implementação do Processador e Ambiente de Desenvolvimento

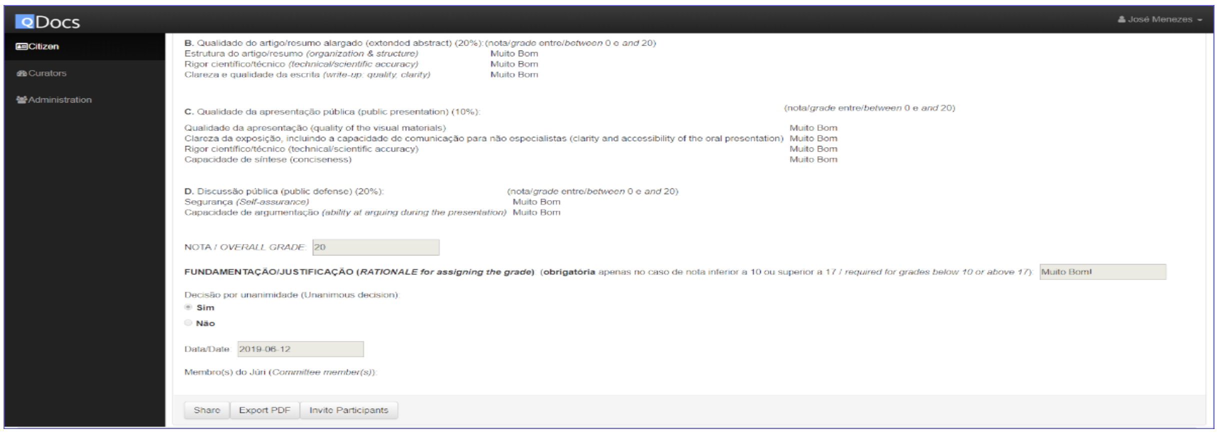
Nome do Aluno (*Name of the student*): José Menezes

Nº 81970

A. Qualidade científica/técnica da dissertação/projeto (*Technical and scientific quality of the thesis*) (50%):	(nota/*grade* entre/*between* 0 e *and* 20)
Estrutura da dissertação/relatório (*document structure*)	Muito Bom
Qualidade da revisão bibliográfica (*survey of related work*)	Muito Bom
Clareza dos objetivos e sua concretização (*clear, well defined, feasible goals*)	Muito Bom
Originalidade do problema /projeto abordado, das metodologias usadas e das soluções propostas (*novelty*)	Muito Bom
Capacidade revelada para aplicar conhecimentos na resolução de problemas não familiares (*ability to address challenging problems*)	Muito Bom
Rigor científico/técnico (*technical/scientific accuracy of the content*)	Muito Bom
Análise crítica das soluções propostas e dos resultados obtidos (*evaluation and analysis of the proposed solutions*)	Muito Bom
Clareza e qualidade da escrita e grafismo (figuras e tabelas) (*quality of the written document*)	Muito Bom
Referências relevantes e abrangentes (*updated and appropriate references*)	Muito Bom

B. Qualidade do artigo/resumo alargado (extended abstract) (20%): (nota/*grade* entre/*between* 0 e *and* 20)

(a)



B. Qualidade do artigo/resumo alargado (extended abstract) (20%): (nota/*grade* entre/*between* 0 e *and* 20)	
Estrutura do artigo/resumo (*organization & structure*)	Muito Bom
Rigor científico/técnico (*technical/scientific accuracy*)	Muito Bom
Clareza e qualidade da escrita (*write-up: quality, clarity*)	Muito Bom

C. Qualidade da apresentação pública (public presentation) (10%):	(nota/*grade* entre/*between* 0 e *and* 20)
Qualidade da apresentação (quality of the visual materials)	Muito Bom
Clareza da exposição, incluindo a capacidade de comunicação para não especialistas (clarity and accessibility of the oral presentation)	Muito Bom
Rigor científico/técnico (technical/scientific accuracy)	Muito Bom
Capacidade de síntese (conciseness)	Muito Bom

D. Discussão pública (public defense) (20%):	(nota/*grade* entre/*between* 0 e *and* 20)
Segurança (*Self-assurance*)	Muito Bom
Capacidade de argumentação (*ability at arguing during the presentation*)	Muito Bom

NOTA / *OVERALL GRADE*: 20

FUNDAMENTAÇÃO/JUSTIFICAÇÃO (*RATIONALE for assigning the grade*) (**obrigatória** apenas no caso de nota inferior a 10 ou superior a 17 / *required for grades below 10 or above 17*): Muito Bom!

Decisão por unanimidade (Unanimous decision):
[✓ **Sim**]
Não

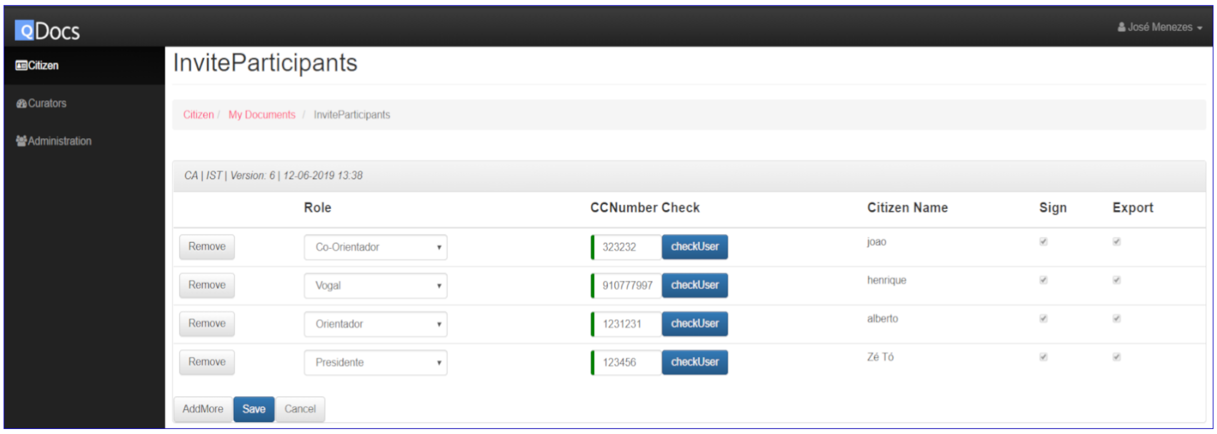
Data/Date: 2019-06-12

Membro(s) do Júri (*Committee member(s)*):

Share | Export PDF | Invite Participants

(b)

Figure 6.10: Document Access



InviteParticipants

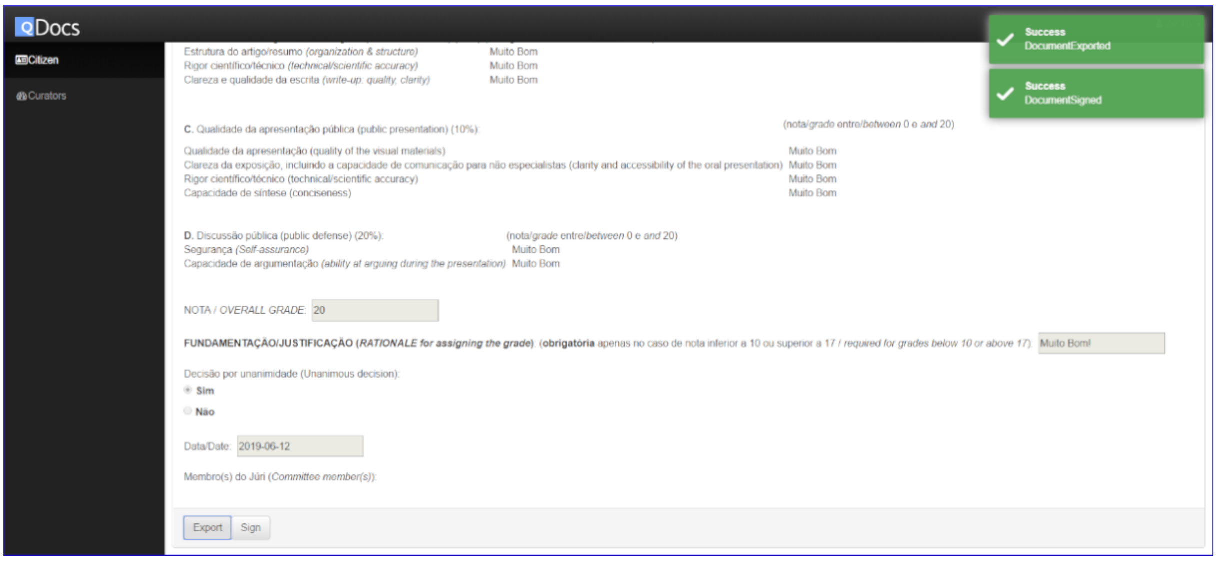
Citizen / My Documents / InviteParticipants

CA | IST | Version: 6 | 12-06-2019 13:38

	Role	CCNumber Check		Citizen Name	Sign	Export
Remove	Co-Orientador	323232	checkUser	joao	✓	✓
Remove	Vogal	910777997	checkUser	henrique	✓	✓
Remove	Orientador	1231231	checkUser	alberto	✓	✓
Remove	Presidente	123456	checkUser	Zé Tó	✓	✓

AddMore | Save | Cancel

Figure 6.11: Participant Invitation



Estrutura do artigo/resumo *(organization & structure)* Muito Bom
Rigor científico/técnico *(technical/scientific accuracy)* Muito Bom
Clareza e qualidade da escrita *(write-up: quality, clarity)* Muito Bom

C. Qualidade da apresentação pública (public presentation) (10%): (nota/*grado* entro/*between* 0 e *and* 20)

Qualidade da apresentação (quality of the visual materials) Muito Bom
Clareza da exposição, incluindo a capacidade de comunicação para não especialistas (clarity and accessibility of the oral presentation) Muito Bom
Rigor científico/técnico (technical/scientific accuracy) Muito Bom
Capacidade de síntese (conciseness) Muito Bom

D. Discussão pública (public defense) (20%): (nota/*grade* entre/*between* 0 e *and* 20)
Segurança *(Self-assurance)* Muito Bom
Capacidade de argumentação *(ability at arguing during the presentation)* Muito Bom

NOTA / *OVERALL GRADE*: 20

FUNDAMENTAÇÃO/JUSTIFICAÇÃO (*RATIONALE for assigning the grade*): (**obrigatória** apenas no caso de nota inferior a 10 ou superior a 17 / *required for grades below 10 or above 17*): Muito Bom!

Decisão por unanimidade (Unanimous decision):
[✓ **Sim**]
Não

Data/Date: 2019-06-12

Membro(s) do Júri (*Committee member(s)*):

Figure 6.12: Participant Document

6.3 Discussion

The scenarios presented in this chapter were tested in a controlled environment, supervised by both advisors of this work, which are participants in the qDocs project. As stated in the beginning of this chapter, both scenarios presented the life-cycle of documents in the academic context. The first scenario was focused on the configuration of a template and its consequences for a citizen who wants to access a document created with that template. The second scenario was focused on the design of a template and its participant configuration. This scenario included the interaction of a citizen with a document created with the designed template and the interaction of the invited participants with that same document.

After this work, qDocs provides more functionalities to all kinds of users of the platform (administrators, curators and citizens): administrators now manage users, curators and services available for the curators; curator users have more options to configure their templates and the different curator general roles provide better management flexibility; citizens now can access shared documents as participants and have specific features available when accessing their documents.

Comparing to what was introduced and discussed in Chapter 2, qDocs differs from the platforms analyzed but has some of the suggested features for a citizen-centered platform.

On the curator side, qDocs gathers information from associated curators through the qBox application, allowing the platform to integrate with entities that use different systems. Regarding template definition, although it is not possible to import documents, qDocs provides an HTML text editor within the qDocs/Curator application, making template definition possible through any device with internet connection. This editor provides all elements present in Table 2.2. With the definition of Data Services and Form Objects it is possible to gather (and store) data directly from curators' databases and use it in different templates, so there is no need to define a Form Object or Data Service twice for the same purpose.

On the citizen side, the process of document creation is similar in every platform, because it is a simple, user-friendly method. With this project, qDocs now has the participant configuration and invitation feature which is different from the platforms evaluated. It is scheduled for future work that citizens must make a request for document creation, and when the request is approved by a Documents Manager they may have access to the document. Regarding digital signature, despite being presented in the second scenario as something a participant can do in a document, it is scheduled for future work.

Regarding content management, which is common to both citizens and curators, it is done within the qDocs platform. When a citizen requests the generation of a document the information needed to do it comes directly from qBox, which is stored in the curator's system. This makes document generation secure, because the information stored in qDocs is relative to the information it needs to request to qBox, so sensitive information is never stored directly in the qDocs platform. Also qDocs is a cloud platform, so all its features are available to users by accessing the qDocs website.

Concluding, if qDocs is evaluated according to the framework presented in Table 2.1 it will have its values oriented to a citizen-centered platform as it should be. Comparing it to A2J Author, which is the other citizen-centered platform evaluated, qDocs may support a broader range of users, since every curator may join the platform, meaning every kind of document will be accessible to citizens.

7 Conclusion

Contents

This chapter presents the conclusion of this work (Section 7.1) and recommends implementations for the future of the work (Section 7.2).

7.1 Conclusion

Document automation is an approach that supports the creation of electronic documents in a flexible and efficient way, which provides several benefits for organizations. Document automation platforms are popular in several areas and keep evolving over time. However, these platforms are most common designed to be used by private organizations, therefore not addressing the purpose to help citizens in general. qDocs is a platform designed for that purpose, it provides a single point of access for citizens to access and manage their own documents (e.g. id cards, forms, certificates). These documents are provided by curators that use the qDocs platform to both manage templates and data. This work proposes an improvement to the qDocs platform, so that every citizen and curator may benefit with the use of the platform.

After analyzing the state-of-art, no other platform seems to match what qDocs offers. Being able to connect with any kind of curator, which has specific requirements regarding documents would be a major benefit for citizens because they would have a single point of access to all their documents. This work was evaluated in a controlled environment by applying qDocs to two scenarios that had different requirements.

7.2 Future Work

This section describes a few implementations to the be done in the future to this work. These suggestions emerged during the development of this work but due to reasons as time constraints or complexity they were not considered. The suggestions for future work are the following: **Curator General Roles Use Cases** — the Templates Manager, Documents Manager and Auditor curator general roles exist in the qDocs platform but they do not do yet anything; in particular for future work the use cases presented for these roles (in Section 4.2) should be implemented; **Strong Authentication** — this feature was scheduled for future work due to its complexity of implementation and since many curators have documents that require citizen authentication in order to be accessed; with this work only the general logic regarding this feature was considered; **Payment** — this feature was scheduled for future work due to its complexity of implementation and since many curators have documents that need to be payed in order to be accessed; with this work only the general logic regarding this feature was considered; **Participant Notification** — when a user invites other users to have access and participate in his document, they should be notified with an access key or hyperlink automatically; **Document Manipulation by Partici-**

pants — participants can only access a document and see buttons relative to the features available for them; however these features should be implemented (such as Sign and Export) and interactions with the document like filling specific fields should be implemented as well.

Bibliography

[1] Google, “Angular - Architecture overview.” [Online]. Available: https://angular.io/guide/architecture

[2] M. Kilger, “A shadow handler in a video-based real-time traffic monitoring system,” *Proceedings of IEEE Workshop on Applications of Computer Vision*, vol. 1992-Novem, pp. 11–18, 1992.

[3] J. A. Menezes, A. Silva, Rodrigues da Silva, and J. Saraiva, “Citizen-Centric and Multi-Curator Document Automation Platform : the Curator Perspective,” in *Proceedings of ISD’2019, AIS*, 2019.

[4] MDSS, “qDocs , Citizen centric document technology,” *White Paper*, 2018.

[5] R. Lankester, “Implementing Document Automation: Benefits and Considerations for the Knowledge Professional,” *Legal Information Management*, vol. 18, no. 2, pp. 93–97, 2018.

[6] M. Lauritsen, “Knowing documents,” *Proceedings of the International Conference on Artificial Intelligence and Law*, vol. Part F1271, pp. 184–191, 1993.

[7] V. Mital and A. D. Elliman, “Document assembly and evidence analysis - Two approaches to hypertext,” pp. 149–175, 1994.

[8] D. R. Mountain, “Disrupting conventional law firm business models using document assembly,” *International Journal of Law and Information Technology*, vol. 15, no. 2, pp. 170–191, 2007.

[9] M. Lehtonen, R. Petit, O. Heinonen, and G. Lindén, “A Dynamic User Interface for Document Assembly,” *Proceedings of the 2002 ACM Symposium on Document Engineering*, pp. 134–141, 2002.

[10] T. F. Gordon, “A Theory Construction Approach to Legal Document Assembly,” *Expert Systems in Law*, pp. 211–225, 1992.

[11] N. Loutas, F. Narducci, A. Ojo, M. Palmonari, C. Paris, and G. Semeraro, “PEGOV 2014: 2nd international workshop on personalization in eGovernment services and applications,” *CEUR Workshop Proceedings*, vol. 1181, pp. 1–9, 2014.

[12] N. Colineau, C. Paris, and K. V. Linden, “Automatically generating citizen-focused brochures for public administration,” *ACM International Conference Proceeding Series*, pp. 10–19, 2011.

[13] A. Kamphuis, “Revolutionary Technology,” *Courts Today*, 2012.

[14] R. J. Glushko and T. McGrath, “Document Engineering for e-Business,” *Proceedings of the 2002 ACM Symposium on Document Engineering*, pp. 42–48, 2002.

[15] R. Eito-Brun and A. Amescua-Seco, “Automation of Quality Reports in the Aerospace Industry,” *IEEE Transactions on Professional Communication*, vol. 61, no. 2, pp. 166–177, 2018.

[16] N. J. Petro Jr, “DOCUMENT AUTOMATION: Using Technology to Improve Your Practice,” *GPSolo*, vol. 32, no. 5, pp. 56–62, 2015.

[17] M. W. Wong, H. Haapio, S. Deckers, and S. Dhir, “Computational Contract Collaboration and Construction,” *Proceedings of the 18th International Legal Informatics Symposium IRIS 2015.*, vol. 512, no. February, pp. 505–512, 2015.

[18] R. Smith and A. Paterson, “Face to Face Legal Services and their Alternatives : Global Lessons from the Digital Revolution,” *White Report*, 2014.

[19] R. Klempner, “The case for court-based document assembly programs: A review of the New York state court system’s ”DIY” forms,” *Fordham Urban Law Journal*, vol. 41, no. 4, pp. 1189–1226, 2015.

[20] J. Frank, “A2J Author, Legal Aid Organizations, and Courts: Bridging the Civil Justice Gap Using Document Assembly,” *Western New England Law Review*, vol. 39, no. 2, 2017.

[21] M. C. Penadés, P. Martí, J. H. Canós, A. Gómez, M. C. Penadés, P. Martí, J. H. Canós, A. Gómez, P. Line-based, N. Loutas, F. Narducci, A. Ojo, and M. Pal, “Product Line-based customization of e-Government documents,” *In PEGOV 2014: Personalization in e-Government Services, Data and Applications (Vol. 1181). CEUR-WS*, 2014.

[22] K. Betts and K. Jaep, “The Dawn of Fully Automated Contract Drafting: Machine Learning Breathes New Life Into a Decades-Old Promise,” *Duke Law & Technology Review*, vol. 15, no. 1, pp. 216–233, 2017.

[23] S. Passera, H. Haapio, and M. Curtotti, “Making the Meaning of Contracts Visible – Automating Contract Visualization,” *IRIS Conference on Legal Informatics 2014, in Salzburg*, vol. 450, no. February, pp. 443–450, 2014.

[24] M. E. Key, “The Universalization of Data Interaction Technology Research between Database and Spreadsheet of WORD Based on OLE Qing-zhong JIA and Bin XIAO,” *DEStech Transactions on Environment, Energy and Earth Sciences (SEEIE 2016)*, 2016.

[25] W. He and L. D. Xu, “Integration of distributed enterprise applications: A survey,” *IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, pp. 35–42, 2014.

[26] R. Iqbal, N. Shah, A. James, and T. Cichowicz, “Integration, optimization and usability of enterprise applications,” *Journal of Network and Computer Applications*, vol. 36, no. 6, pp. 1480–1488, 2013.

[27] Y. Zhai, M. Liu, J. Zhai, X. Ma, and W. Chen, “Cloud versus in-house cluster: Evaluating amazon cluster compute instances for running MPI applications,” *State of the Practice Reports, SC’11*, 2011.

[28] D. Molnar and S. Schechter, “Self Hosting vs . Cloud Hosting : Accounting for the security impact of hosting in the cloud,” *9th Workshop on the Economics of Information Security (WEIS 2010)*, 2010.

[29] R.L.Grossman, “The case for cloud computing,” *IT Professional*, vol. 11, no. 2, pp. 23–27, 2009.

[30] F. Monteiro, *Learning Single-page Web Application Development*, packt publ ed., 2014.

[31] A. Mesbah and A. V. Deursen, “Migrating Multi-page Web Applications to Single-page A JAX Interfaces,” *Mesbah, Ali, and Arie Van Deursen. ”Migrating multi-page web applications to single-page Ajax interfaces.” 11th European Conference on Software Maintenance and Reengineering (CSMR’07). IEEE*, 2007.