TÉCNICO LISBOA

dDocs/Dashboards: Data Analytics for a Citizen-Centric Platform

Daiane Pereira de Oliveira

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisor: Prof. Alberto Manuel Rodrigues da Silva

Examination Committee

Chairperson: Prof. João António Madeiras Pereira
Supervisor: Prof. Alberto Manuel Rodrigues da Silva
Members of the Committee: Prof. Daniel Jorge Viegas Gonçalves

November 2017

Abstract

With the exponential access over the information and communication technology, and the interesting in investing in applications still being higher in different areas, the Government is one of these interested areas, which results in the creation of a well-known concept: e-Government. This evolution decreases the bureaucracy, corruption and costs, while, increasing the citizens' satisfaction since they do not need to go the physical locations and the end-process can be faster than the traditional way.

dDocs is a platform that was created having as a base the e-Government and, also, Record Management since documents needed to be changed from paper to digital. Having three user roles: citizens, curators and dDocs administrator, this platform provides the communication between citizens and curators without having the critical data.

These communications and, even, the user's management, results in events that could be transformed into something powerful, that is why it was decided to create a system to provide for each user role a dashboard to monitoring their activity, that is called dDocs/Dashboards.

In this dissertation, the dashboards were designed and developed after understanding the concepts behind it, analyzing the tools in the market that could be used during this work, understanding the best practices to reach a well-designed dashboard and collect the requirements. Also, it describes the process of evaluation to measure dDocs/Dashboard's quality.

Keywords

dDocs; dashboards; data analysis; visualizations.

Resumo

Com o acesso exponencial às tecnologias de comunicação e informação, o interesse em investir em aplicações mantém-se elevado em diferentes áreas. O governo é uma dessas áreas interessadas e com base nisso criou-se um conceito bem conhecido, denominado, e-Government. Essa evolução diminuiu a burocracia, a corrupção e os custos, enquant que aumentou a satisfação entre os cidadãos, uma vez que não necessitam de se deslocar fisicamente e o final do processo pode vir a ser mais rápido de se obter do que o modo tradicional.

dDocs é uma plataforma criada com base no e-Government e, também, na Gestão de Registos, pois os documentos passam a ser digitais e não de papéis. Tendo como três tipos de utilizadores: cidadãos, curadores e o administrador do dDocs, esta plataforma fornece a comunicação entre os cidadãos e os curadores sem conter no sistema os dados críticos.

Estas comunicações e, ainda, a gestão de documentos/templates resultam em eventos, que podem ser transformados em algo poderoso, tendo sido tomada a decisão de criar um sistema que fornece para cada tipo de utilizador uma dashboard para monitorização das respectivas actividades, denominado dDocs/Dashboards.

Nesta dissertação, as dashboards foram desenhadas e implementadas depois de compreender os conceitos que estão por trás, analisar as ferramentas existentes no mercado que poderiam ser utilizadas neste trabalho, compreender quais são as melhores práticas de desenho para alcançar uma boa dashboard e reunir os requisitos. Ainda, descreve o processo de avaliação para medir a qualidade do sistema dDocs/Dashboards.

Palavras Chave

dDocs; dashboards; análise de dados; visualizações.

Contents

List of Figures

List of Tables

Acronyms

BI	Business Intelligence
ETL	Extract, Transform, Load
IoT	Internet of Things
KPI	Key Performance Indicator
PDI	Pentaho Data Integration

1

Introduction

Contents

Being e-government an intensive use of information technologies with the purpose of services provision, effectiveness improvements and democratic values promotion [2], which brings advantages, such as, time savings, costs savings, improvement of the public administration competence, decrease in corruption and implies less bureaucracy and Records Management, the field of management responsible for the efficient and systematic control of the creation, receipt, maintenance, use and disposition of records, including processes for capturing and maintaining evidence and information about business activities and transactions in the form of records [3], these two areas are capable of showing the importance of data, since they must deal with a huge amount of it, it must be careful with the data showed to the users.

Data visualization is an easy way for the human brain to understand what is behind in large or small datasets. The information is presented in a graphical form and it is quantitative. It can be used to make better decisions or to tell stories about the surrounding environment. The last refereed one it is benefits mainly the marketers and the journalists. To accomplish a good data visualization, Edward Tufte [1] says that "complex ideas communicated with clarity, precision, and efficiency".

Between several data visualizations techniques, the most relevant one is a dashboard. It only has the most important elements to tell a story and these elements are shown based on metrics, previously, defined to reach some goal to the user. It provides a way to combine data from different sources.

1.1 Context

Government and citizens are constantly communicating with each other, so there was a need to develop a web based application known as dDocs where there is no necessity to be physically present on the entity's location and, even, leaves behind the traditional document way of digitalization. This system is capable of separating the document format from the document data, which implies some advantages like improve performance and reduce resources usage.

In dDocs, document templates are templates from the original document of the organizations without critical information and are managed by a specific curator, then, citizens can search them in dDocs.

A document, in dDocs, has an identifier and it is created based on a document template and can have life events, which can be, for instance, birth, death, education, health or professional life and/or as document types, for instance, identification documents, certificates and requests for certificates.

In this platform there are three different permissions where the users can perform specific actions. The users permission, in cause, are citizens, curators and the administrator.

The citizen is someone that is not on dDocs as curator or administrator, he can search for a specific template and after select the desired one, it turns into a document. He is able to perform four actions:

[1] https://www.edwardtufte.com/tufte/

create, edit, close and, share a document with some curator or others citizens.

A curator is a public or private organization that, responsibly, designs and manages the templates owned by it.

Meanwhile, the administrator is someone that manages the dDocs system.

In the figure 1.1 is possible to take an overall look about this platform.

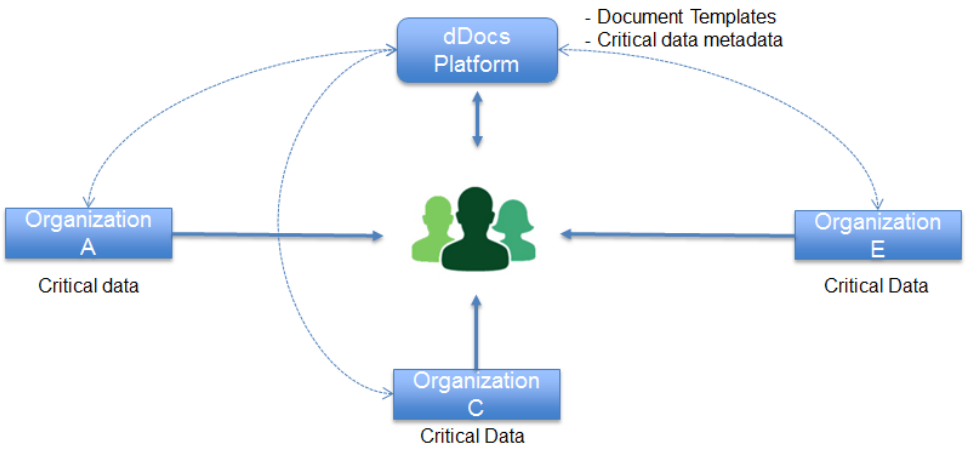


Figure 1.1: dDocs Architecture (Extracted From [1])

As shown above, it supports the citizen-centric approach, where the citizen is the center of all interactions. The critical data does not pass through dDocs, it is directly sent to citizens, dDocs only has the document templates and the critical data metadata [1]. This platform is accessible to any device and allows to design electronic documents through orchestration of services and data provided by the curators and they can be managed. Also, it is a collaborative and social 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. Any document can be searched by life events, by document types, by curators or by curator's classification [4].

The main features of citizens and curators, in the dDocs system [4], are:

Citizens

- Access and management of his own documents;
- Documents sorting and search features;
- Document sharing with other users;
- Citizen document filling, feeding Curator data collection, or forms;
- Integration with payment systems;

- Hands free authentication mechanism;
- Several levels of legally binding authentication;
- Authenticity validation system for critical identification documents.

Curators

- Curator-specific user management and role assignment;
- Management of protocols and contracts with other curators;
- Management and configuration of internal and external data services;
- Management and configuration of user interface objects;
- Management of document templates;
- Configuration and WYSIWYG interactive design of document templates.

dDocs was developed at the Information and Decision Support System Group of INESC-ID [2] and by the support of MDSS [3] that provide their fundamental concepts.

1.2 Problem Definition

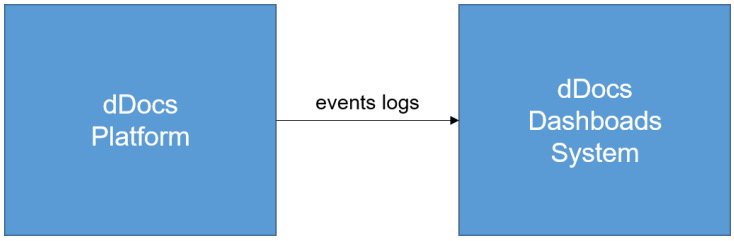


Figure 1.2: Relation Between dDocs And dDocs/Dashboards

Being dDocs a system with events monitoring, where an event provides historical information related to it, making it possible to do something useful with this events and show the result to the users, which can improve users satisfaction in the system and, even, give an opportunity to them to track their activity. In the figure 1.2 it is explicit the dependencies of dDocs/Dashboards on dDocs.

Having the notion that this platform, dDocs, has three distinct user roles and events logs, there was an opportunity to design and develop a system of dashboards.

[2] https://www.inesc-id.pt
[3] https://mdss.pt

1.3 Objectives

After contextualizing and defining the problem, there are objectives to achieve, which are:

- Research about big data, ETL tools, data visualization and dashboards;
- Design and implement dashboards based on data provided by dDocs, taking into account the different users roles;
- Evaluate the solution's quality.

1.4 Document Outline

The remaining six chapters of this dissertation is organized as the following:

- **Chapter 2:** Background, describes the main concepts related to this work, which are: big data, ETL tools, data visualizations and dashboards and presents analysis related to technologies involved in this work and about related systems.
- **Chapter 3:** dDocs/Dashboards Requirements, that shows the different needs to apply during the design decision making.
- **Chapter 4:** dDocs/Dashboards Prototype Technical Description, that includes an overview over the architecture as well as the data model and describes the dashboards system.
- **Chapter 5:** Evaluation, describes two distinct evaluations: performance evaluation and usability evaluation.
- **Chapter 6:** Conclusions, discussion of the main development and what can be a must have in the future.

2

Background

Contents

This chapter overviews about the main technologies and concepts related to this research work. Therefore, it is separated into three sections: General Concepts, Technologies Analysis and Related Systems Analysis.

Section 2.1 presents the definition, characteristics and applications of Big Data, which leads to the description of NoSQL Databases and tools to extract, transform and load data to, then, embrace the data visualization and dashboards concepts.

Section 2.2 presents some NoSQL Databases, ETL tools and, last but not least, Data Visualization tools.

Section 2.3 presents a set of systems, related to this work, to be analyzed.

2.1 General Concepts

In this section will be described the fundamental concepts to comprehend the context which this work is inserted in.

2.1.1 Big Data

Big Data [5] is defined as a term for datasets that are complex and large, containing different data structures. A few years ago, when managing big data, big companies such as Google and Amazon, brought to the market solutions such as the Map-Reduce framework in order to deal with.

The characteristics of Big Data are known as ”5Vs” [6] which are: Volume, Variety, Velocity, Value and Veracity. Volume: is associated with the amount of data produced every second. Variety: indicates the different types of data that can be used. This diversity includes structured and unstructured data. In the past, there only existed structured data for relational databases but afterwards arose the need for photo and video storage, as well as social networking updates, which fall under unstructured data, which is not supported by the relational databases. Velocity: related with speed during the production of new data and during the interaction with data in real-time. Value: refers to making data valuable. Veracity: points to trustworthiness of the data.

Big Data technologies are being used in e-commerce and marketing, government and public health, transportation, industry and surveillance [5].

Qingshan Xu, Lihua Yang and Chunling Yu described the influences of big data to Chinese e-government in six aspects [7]:

1. **Technical aspect** - better use of resources, integration of all government entities information and e-government implementation based on data;

2. **Business aspect** - provide collaboration and efficiency increase;

3. **Service aspect** - integration of public data of different sectors;
4. **Management aspect** - enhance policies based on data, find trends and warnings, less costs, better efficiency and quality of services;
5. **System aspect** - improve e-government platform, laws, organization, services and regulations;
6. **Security aspect** - information security and national security;

US government, also, uses big data in areas, such as, national security, social services, taxation and data accessibility. National security to predict and prevent threats, social services to detect frauds, taxation to detect fraud and tax avoidance and data accessibility, in data.gov website, to integrate all the information and make it available to researches, agencies and citizens [8].

2.1.2 Non-Relational Database Management System

Through the last years, the volume of data has been increasing due increase in the number of sources that are, usually, producing data. The management of this amount of data was hard to do in traditional relational databases and was necessary to create a new model and technologies to deal with this problem.

Widely known as NoSQL (Not Only SQL), this term took progress in 2009 and, now, there are more than 225 NoSQL databases in the market. These databases have in common some fundamental characteristics like, schema-less storage, are distributed and fault-tolerant systems, have the ability to make replications and they are capable of making multiple versions of data.

Exist four NoSQL database models [5], that were developed to satisfy specific problems. They are: Key-value stores, Document-oriented databases, Wide-column stores and Graph databases.

Key-value stores consist in transforming the concept of data into a dictionary or a map of key-value pairs, where the keys are an unique identifier of the record and are performed like an index to access the respective value. Each record is a pair key-value and a value can be in strings or, even, arrays.

Key-value databases can easily do a query to find a value, through its specific key. Besides, some of this kind of databases have a search feature, which improves flexibility.

This model of NoSQL database is used to improve performance, by cache data from traditional databases, by stored images and audio, by stored users data in mobile applications and, even, stored configurations.

Document-oriented databases was invented to store documents, for example, documents from Microsoft Word but, then, was more beyond to define that a document is any kind of domain object.

It pairs each key with a document where it can have many key-value pairs or nested documents. These kind of systems have the advantage to make a full search, since it is possible to search by keys or values.

These documents include encoded data in XML, YAML, JSON, BSON, PDF and Microsoft Office documents.

Document-oriented databases are useful for: back-end part of websites with a huge amount of reads and writes, follow variable types of metadata, JSON data structures applications and store a vast collection of textual documents.

Wide-column stores instead of storing data in rows, it stores them in sections of columns. Each section contains key-value pairs. Keys represent rows that are primary keys and values are column families where each column family have a primary key of its columns.

These are used for applications that constantly are doing write operations to the database, also, applications that have dynamic fields and applications that enables some inconsistency in short time.

Wide-column stores are applicable in security analytics, big science, stock market analysis and web applications.

Graph databases have data in a network of nodes connected with edges. Each node represents an entity, while each edge represents the relationships between two nodes.

This model of NoSQL database is highly used in social networking but, also, they are important to identity and access management, business process management, recommendations of products and services, and to find patterns.

The table 2.1 exposes the differences between NoSQL models by performance, scalability, flexibility and complexity.

Table 2.1: Comparison Between The Four NoSQL Models

Model	Performance	Scalability	Flexibility	Complexity
Key-value	High	High	High	None
Document	High	High	Moderate	Low
Wide-column	High	Variable	High	Low
Graph	Variable	Variable	High	High

2.1.3 Extract, Transform, Load Tools

It is an important part of Business Intelligence (BI) systems, where, at least, 80% of the struggle is on ETL processes [9].

Through a tool, an ETL process is capable of transforming extracted data from different sources. During transformation, data can be computed to create new values or, even, records to accomplish a desirable structure and, furthermore, it is possible to perform data cleaning. Then, the resulted data is loaded in the appropriate flat file, database or data warehouse.

To a better understanding, the figure 2.1 represents an example of a ETL tool with a process implemented.

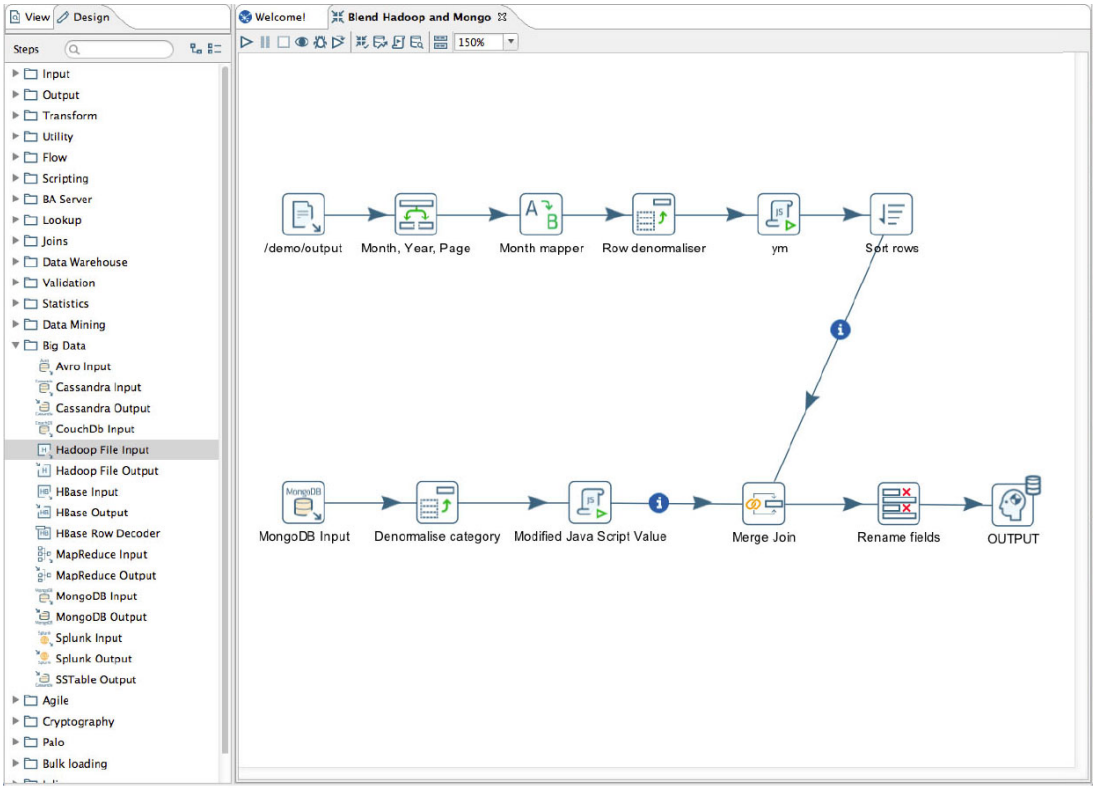


Figure 2.1: Overview Of An ETL Tool

2.1.4 Data Visualization

Before creating visualizations, there are some considerations to take into account. It is essential to understand who will be the audience, since this visualizations are a great way to communicate information, but if it does not meet the audience's expectation and needs, it can be a great disaster. Another consideration is about tools, first of all, it is important to know what it is in the market and understand the differences between them to make the right choice.

2.1.4.A Concept

Data visualization has been around for centuries in several areas of activity. In the late of 1700's, William Playfair, who is considered the "father" of data visualization, invented a few diagrams that are used often until today, like, bar charts, line charts and pie charts. Unfortunately, during his life, his work didn't have a major effect. The importance of representing information through visual was shown by John Snow when he decided to draw a map to determine the cause of the cholera epidemic of 1854. Later, Florence Nightingale, demonstrated, through pie charts, that the major cause of British soldiers deaths during the Crimean War was the lack of hygiene in the hospitals which improved the conditions of the hospitals. Then, in 1861, Charles Joseph Minard, mapped the march of Napoleon to Moscow to represent the massive loss of his army over the advance to Moscow and the retreat. Only in 1977, the Princeton University's professor, John Tukey, showed the data visualization techniques to analyze and present the data. Today, data visualization has an impact in BI and Business Analytics [10].

Visualization is defined as a communication of information through graphs. A single picture interpretation is processed at the same time as the human perception system, making information in pictures faster to understand than a page of words [11], for example. The speed of interpretation of a page of words depends of the sequence of reading. A graph or a map have the advantage of being understood by different cultures.

It is used on applications to provide different views of data and to detect some pattern or anomaly. Visualizations are an effective way for present information to people and enable to support some facts that happen, in real time, from global warming to economic trends [11]. Maps and charts are abstract representations of data that people see frequently and go beyond than being just for business and economic analysis. It could be used, for example, in MRIs and DNA sequencing results. In an interactive visualization it is possible to rearrange the information for preferences of users and an application can be fully driven through visualizations.

2.1.4.B Best Practices for Design

For constructing visualizations there are some guidance to think about before they are implemented:

1. Problem domain;
2. Data abstraction (What dataset will be used?, Where is it available?);
3. Task abstraction (What tasks?, What is the purpose?).

The tasks, described in [12], are:

Discover - Find new information;

Present - Transmit information, by telling a story with data, or guiding an audience through a series of cognitive operations;

Enjoy - Wake user curiosity that might be both stimulated and satisfied;

Annotate - New annotations in some visualization element that already exist;

Record - Save or capture visualization elements;

Derive - New elements based on existing ones;

Lookup - User search for information where s/he: knows what s/he is looking for and where it is;

Browse - User search for information where s/he: does not know what s/he is looking for, but know the location;

Locate - User search for information where s/he: knows what s/he is looking for and does not know the location;

Explore - User search for information where s/he does not know what and where it is.

Identify - Find information without prior knowledge;

Compare - Compare similar entities of data represented;

Summarize - Overview of everything.

2.1.5 Dashboards

This section states the main concepts related to the dashboards, where it is comprehend the definition, goals, characteristics, furthermore, the metrics and the design to materialize them with a good look and feel.

2.1.5.A Concept

In BI, dashboard is the most popular data visualization technique [10]. It is defined by Stephen Few in [13] as a "visual display of the most important information needed to achieve one or more objectives, consolidated and arranged on a single screen so the information can be monitored at a glance."

Dashboards have three functions to knowledge business performance: measure, monitor and manage.

As described in [14], it helps to decrease time to make decisions since it provides facts, through data. Also, concede the privilege to compare values since the past to grant new concrete objectives. It should focus on which data is more important to each role in the organization.

Dashboards can be classified in three categories:

Strategic - summarizes what is going on in the past performance of the organization.

Operational - tracks the organization's activities.

Analytical - allows interactions to explore the details of the data.

2.1.5.B Metrics

There are some challenges before the development of dashboards, one of them is to choose which metrics should be tracked. The dashboard design depends on the selected metrics. Furthermore, metrics helps to understand what will be achieved.

Basically, a metric can be defined as a quantitative data that may be counted, ordered or aggregated.

For instance, considering a CEO that wants to build a dashboard for his organization, what is the major metric? It could be net income. After that, consider what factors influence this metric, it could be sales, and, then, what factors take to sales? One reason could be because of the average sales price. As noticed, it is a good practice to think, first, at top-level and, then, lower the level.

2.1.5.C Best Practices for Design

Firstly, the placement, attention, cognitive load and interactivity are important to improve the effectiveness of a dashboard.

Before beginning the construction of the design, fundamental questions should be answered to understand the need of develop a dashboard and, also, to know which information is necessary to show. For that, typically, it is essential to make interviews to question the following:

1. Who are the users?
2. What are their goals?

Several operations can be use in the dashboards, and they are described as:

- **Drill Down** - allows to give more details.
- **Filter** - shows only what is selected.
- **Comparison** - compares several data.
- **Alert** - prompts attentiveness when a metric is above or below of the threshold.
- **Export** - exports the data from the dashboard.
- **Summary** - details the description of a visualization.

- **Tagging** - allows to custom the dashboard with the data the user wants.
- **Annotation** - brief comments about a visualization.
- **Chart** - organizes and represents quantitative data. For instance, line charts, bar charts and pie charts.

Important information should be quickly noticed and the data visualizations should, clearly, show the information and must be easy to interpret.

Line charts and bar charts are a good example to visualize quick comparisons.

One important tip is about avoiding distractions, the dashboards should not contain animations, motion, too bright colors or too many colors.

On analytical dashboards, the use of interactivity should be essential. Filters and drill down are good practice to be in.

Meanwhile, on operational dashboards, contains several Key Performance Indicator (KPI), which is a way to measure if an action is in the right direction for attending some goal, purpose by the organization and they are used to compare and to alert, typically through colors (red when some KPI is below the threshold and green to inform that KPIs are in a good performance), the tonality of the colors should be always the same. It is important to notice that KPI is different from a metric, since the first one is an important indicator to reach some goal of the organization, while the second one is something to be measured. A metric can turn to be a KPI if is relevant for the organization's strategy.

Texts should be carefully chosen to be in the dashboards, only the important ones can be inserted and the size of them must not be small so, it won't difficult the reading. Usually, to understand if the use of texts are effective in dashboards, it is resorted to usability tests.

2.2 Technologies Analysis

This section describes the study of several tools that are in the market that can be a possible candidate to complement the development of dDocs/Dashboards.

2.2.1 NoSQL Databases

NoSQL Databases are described by model's order and are mentioned to understand some solutions that are in the market. Although, they shared a set of features, each one was designed to serve a specific purpose.

Amazon DynamoDB [1] is a fully managed, proprietary, NoSQL database service on the cloud. It's

[1] https://aws.amazon.com/pt/dynamodb/

fast and flexible, since one of the advantages is single-digit millisecond latency, which is good, for instance, for web, Internet of Things (IoT), gaming and mobile. It is compatible with some models, such as, key-value and document-oriented.

Voldemort [2] is a key-value storage system, capable of eliminating a separate caching tier in a way that merges in-memory caching with the storage system. Also, Voldemort reads and writes scale horizontally. There's a mechanism to version data items, which implies more data integrity, in case of failure, it has a good performance, since each independent node handles 10-20k operations per second (it depends on: machines, network, disk system and data replication factor) and uphold pluggable placement strategies, in case of necessary distribution over data centers. Another advantage is about data replication that can be decided by an API to support a wide range of application specific strategies.

Cassandra [3] is an open source, wide-column storage system from Apache. It includes Cassandra Query Language (CQL) as an alternative to SQL but, for easy understanding, has a similar syntax to SQL, where any authorized user can access data. It offers the possibility to integrate with Hadoop giving support to MapReduce. Cassandra scales linearly, which means, the throughput increases as the number of nodes increases, without interruption to applications, which implies, fast responses. It supports all data formats (structured, semi-structured, unstructured) but does not supports JOINs between tables because it only allows one query per table.

HBase [4] is a distributed, open source, wide-column storage system from Apache, designed to accommodate a fast random access to a huge amount of structured data. HBase is part of Hadoop system, specifically, it is the database. It offers random real-time read and write access to data in the Hadoop Distributed File System (HDFS). Also, HBase is horizontally scalable.

MongoDB [5] is an open source, document-oriented database that manages concepts like collection and document. A collection is a set of documents that is similar to a table in RDBMS, while a document is like a row in RDMS, composed by a field and value pairs. Documents have dynamic schema, which means that documents grouped in one collection can have a different set of fields or structure and it is possible to have different types of data in common fields, in the collection. MongoDB stores data records as BSON documents, that is a binary representation of JSON documents and contains more data types than JSON.

CouchDB [6] is an open source, document-oriented database developed by Apache and consists in grasping the web without difficult use. Data (documents) are stored using JSON, for queries it uses

[2] http://www.project-voldemort.com/voldemort/
[3] http://cassandra.apache.org/
[4] https://hbase.apache.org/
[5] https://www.mongodb.com/
[6] http://couchdb.apache.org/

JavaScript, the data access is through an HTTP protocol. A document is identified with a unique ID and contains fields with different types, such as, strings, numbers, dates, ordered lists and associative maps. Data is stored in semi-structured documents and, to resolve a problem of adding structure, was necessary to create a view model. Views are a result of aggregation and reports on documents, in the database, and they are built dynamically without influence in the underlying documents. It's possible to have many different views of the common data and there's no need for full re-indexing views because of incremental documents updates.

AllegroGraph [7] is a proprietary graph database developed by Franz, Inc., designed to build semantic web applications, *i.e.*, are web applications that include standards such as Resource Description Framework (RFD) where data and meta-data are stored as triples. These triples can be query over SPARQL, *i.e.*, an RDF query language and Prolog.

Neo4j [8] is an open source, native graph database designed for optimizing storage of nodes and relationships. It contains an SQL-inspired language called Cypher (CQL) and fully supports ACID rules.

Discussion

For all NoSQL databases mentioned above, there are some features to take into account when there is comparisons between them. Those features are:

License - the cost of using each solution.

Model - model of NoSQL Database.

Complexity - how easy it is to use it.

Performance - how correctly and quickly the database performs all important functions.

Table 2.2: NoSQL Databases Comparison

	License	Model	Complexity	Performance
Amazon DynamoDB	Commercial	Key-value	None	High
Voldemort	Open Source	Key-value	None	High
Cassandra	Open Source	Wide-column	Low	High
HBase	Open Source	Wide-column	Low	High
MongoDB	Open Source	Document	Low	High
CouchDB	Open Source	Document	Low	High
AllegroGraph	Commercial	Graph	High	Variable
Neo4j	Open Source	Graph	High	Variable

[7] http://franz.com/agraph/allegrograph/
[8] https://neo4j.com/

2.2.2 ETL Tools

In this work it is necessary to use an ETL tool and before take any decision, a research was made to reunite a few tools and understand the advantages and disadvantages of each one to, then, make comparisons based on some features.

Pentaho Data Integration [9], also known as Kettle, is a popular open source tool that is part of Pentaho and responsible for the extract, transform and load processes. It is mostly known because of the non existing complexity of use, since it is possible to use it without writing code and have a friendly graphical environment. Pentaho Data Integration implements a metadata-driven approach, which means that it specifies what to do and not how to do it. It can be used for data warehouse, migration of data from a database or application to another, integrate applications, load massive data into databases, data cleaning and export data from databases to flat files. It has the advantage of being prepared for receiving data from any source. Its components are Spoon, Pan, Kitchen and Carte. The first one is where the processes are developed, it allows transformations (basic data flow) and jobs (execute several transformations and other jobs in sequence). Pan executes the transformations from Spoon while Kitchen executes jobs. The last one, Carte, is a webserver for running and monitoring tasks.

Talend Open Studio [10] is an open source platform that is qualified to perform data warehouse, data profiling, data cleaning and data migration from a database or application to another. Although it is under open source license, it offers additional services, which requires a commercial license. Its components are Business Modeler, Job Designer and Metadata Manager, these components do not require any technical knowledge.

Jedox ETL [11] is a client-server application where Jedox ETL server provides all ETL functionalities and is part of Jedox that is a Business Intelligence solution, which has a proprietary license, although it offers a 14-day free trial for Jedox Cloud and 30-day free trial for Jedox Premium. The differences between these two is that, in Jedox Cloud every functionality is in the cloud, while, in Jedox Premium it must be downloaded and installed in order to access its full functionality. Some notable users are organizations such as Siemens and Fiat. In Jedox ETL, the data can be extracted from different sources, transformation and loading of master and transactions into Jedox models. It, easily and automatically, imports data and allows modeling complex aggregations and transformations. Also, provides logs and reports over ETL tasks. It can be used from command line or from web based ETL Manager.

[9] http://community.pentaho.com/projects/data-integration/
[10] https://www.talend.com/products/talend-open-studio/
[11] https://www.jedox.com/en/

SAP Data Services [12] is a proprietary application which allows the creation and execution of workflows with data from data stores as databases, flat files, applications or web services, for example. It offers transformation functions such as joins, lookups, aggregations, filters and many others. This application has functionalities to perform data cleansing, data profiling, text analysis, data quality and batch processing. The last one, batch processing, allows query Data Services to get an instant response based on a workflow. SAP Data Services is integrated with other SAP solutions.

IBM InfoSphere Information Server [13] is a proprietary product of IBM to accomplish data warehousing and data integration. It operates in four areas: data profiling, data quality, data transformation and data to end-users. It has a component named Datastage and Qualitystate for integrating data stored in data warehouses and develop ETL processes. It is prepared for a wide amount of data and can easily integrate with other applications.

Discussion

After understanding the above tools for extract, transform and load processes, some features must be analyzed to consider which one will be used in this work. Those features are:

License - the cost of using each solution.

Easy to Use - the level of complexity to use the tool.

NoSQL Connections - understanding if the tool supports NoSQL connections.

Large Volume Performance - comprehending if it supports large amount of data.

Table 2.3: ETL Tools Comparison

	License	Easy to Use	NoSQL Connections	Large Volume Performance
Pentaho Data Integration	Open Source	Yes	Yes	Yes
Talend Open Studio	Open Source	Yes	Yes	Yes
Jedox ETL	Commercial	Yes	No	Yes
SAP Data Services	Commercial	No	No	Yes
IBM InfoSphere Information Server	Commercial	No	Yes	Yes

2.2.3 Data Visualization Tools

Currently, there are many tools for developing visualizations in the market and so, it is essential to analyze them and make an evaluation to know which one could fit this purpose. There are some tools that require coding which implies that they are for developers while other tools do not need any coding.

[12] https://www.sap.com/products/data-services.html
[13] https://www.ibm.com/analytics/us/en/technology/information-server/

amCharts [14] is a JavaScript library for charts and maps that has the capability of being responsive and customizable. This library offers charts like column, bar, line, area, pie, funnel, bubble and gauges. It provides a clean visualization and it is used by well-known companies such as NASA, Microsoft, IBM, Amazon and PayPal.

D3.js [15] short for Data-Driven Documents which is a JavaScript library for create dynamic, interactive data visualizations in web browsers using HTML5, SVG and CSS technologies. Contains various plugins and components and it has advantages of being free and open source.

Tableau [16] is a Business Intelligence tool where users can drag and drop data to, then, analyze in real-time, offers an easy way of combining multiple databases and even, provide the option for combining with other users. It may interact with files, RDBMS or NoSQL databases. This tool has a wide quantity of charts with great quality.

Google Charts [17] is a JavaScript library that renders charts using HTML5 and SVG standards and is meant to improve web applications by adding interactive charting capability. Google Charts provides a variety of charts, such as, bar, area, pie and gauges.

Infogram [18] is a tool which starts with choosing a template, followed by the possibility of adding charts and, at lastly, sharing that visualization. It has a free version for non-profit and educational organizations but with a few restrictions, such as, not importing CSV, XLS files, lack of API and no possibility to connect live data.

Plotly [19] is a tool to create charts and dashboards. The visualizations can be embedded in web applications and it has the ability to support social shares. About integration with databases, it offers compatibility with systems like MySQL, Microsoft SQL Server, PostgreSQL and can support csv files.

Chart.js [20] is an open source library with few charts, such as, line, bar, radar, polar area, pie, doughnut and bubble. Uses HTML5 to render charts and all charts are responsive. It is a great solution for small projects.

ZingChart [21] consists in a JavaScript library which builds responsive charts in HTML5 Canvas, SVG, VML or as an image. It has more than 100 types of charts and can render, in a fast way, a wide quantity of data. It is compatible with most well-known browsers and devices. It is possible to make

[14] https://www.amcharts.com/
[15] https://d3js.org
[16] http://www.tableau.com
[17] https://developers.google.com/chart
[18] https://infogr.am
[19] https://plot.ly
[20] http://www.chartjs.org/
[21] https://www.zingchart.com/

an integration with AngularJS, PHP, JQuery, Ember, React and Backbone and has the option for live data.

FusionCharts [22] is a tool that has more than 90 charts using JavaScript, SVG and VML technologies and supports JSON and XML data formats. Fusion Charts is compatible with Internet Explorer 6+, Firefox 2+, Safari 5.0 +, Opera 10 and 11, and Chrome. About exporting charts, it offers the option to do so in JPG, PDF, SVG, PNG. The charts are customizable and support interactions. It also supports data export of CSV, XML files.

HighCharts [23] is another JavaScript library. Charts can be exported in PNG, JPG, SVG and PDF and supports, for instance, line, spline, area, column, bar, pie, scatter, angular gauges, bubble, box plot, error bars, funnel, waterfall and polar chart types. About compatibilities, it can perform in all modern browsers (mobile and desktop) where rendering is done through SVG in modern browsers but IE 6/7/8 is through VNL and Android 2.x through HTML5 Canvas technology. In order to create stock or timeline charts there is a package called Highstock.

Discussion

These data visualization tools will be analyzed in the next table based on a few features. Those features are:

License - the cost of use of each solution.

Easy to Use - the level of complexity to use the tool.

Customization - understanding if the tool allows custom graphs.

JavaScript Library - understanding if the tool usage is through code or through a graphical environment.

Responsive - understanding if the tool adapts to any device.

Zooming - understanding if charts can offer zoom functionality.

Table 2.4: Data Visualization Tools Comparison

	License	Easy to Use	Customization	JavaScript Library	Responsive	Zooming
amCharts	Open Source	Yes	Yes	Yes	Yes	Yes
D3.js	Open Source	Yes	Yes	Yes	Yes	Yes
Tableau	Commercial	Yes	Yes	No	Yes	Yes
Google Charts	Open Source	Yes	Yes	Yes	Yes	No
Infogram	Open Source	Yes	Yes	No	Yes	No
Plotly	Open Source	Yes	Yes	No	Yes	Yes
Chart.js	Open Source	Yes	Yes	Yes	Yes	No
ZingChart	Open Source	Yes	Yes	Yes	Yes	Yes
FusionCharts	Open Source	Yes	Yes	Yes	Yes	Yes
HighCharts	Open Source	Yes	Yes	Yes	Yes	Yes

[22] http://www.fusioncharts.com/
[23] http://www.highcharts.com/

2.3 Related Systems Analysis

This section allows to discovering, analyzing and comparing similar systems, that is, they contain a module about insight to understand what is being used in this area.

One of the first steps is to explore systems with analytics module to understand the importance of the metrics, to understand what turns a simple and clean dashboard and what dashboard's operations are typically chosen.

In the following, four systems are briefly described and, each specifies how each corresponding insight module represents data.

Facebook is one of the most well-known social network services with millions of people interacting daily. Here, someone can create a page to promote a local business or place, a company, organization or institution, a brand or a product, an artist, band or public figure, entrainment, a cause or a community. Since the page's creation, the administrator(s) have access to a tab called "Insights" where it shows useful information about the page in some tabs, which will be named and detailed in the following.

- Overview - During a selected date range, it shows the page summary that includes: actions on page, page views, likes, followers, reach, post engagements and videos.
- Promotions - Shows the most recent promotions created by the administrator.
- Followers - Total followers, net followers and where follows happened.
- Likes - This tab presents the total page likes, net likes, where the likes happened. All through a chart that makes a comparison between time and average performance of the page.
- Reach - Has charts where it shows the evolution of post reach, likes, comments, shares, hide, report as Spam, unlikes and total reach through time.
- Page views - Is a tab that has information about who viewed which section of the page.
- Page previews - Contains a line chart with the number of page previews by time, it has another line chart with the total people who previewed by time and one more chart with information filtered by age and gender.
- Actions on page - Presents total actions on page, and then, it goes to a more detailed information, about the clicks on some actions (Get Directions, Phone Number, Website) and here it is possible to have a general idea through time or specify the information by age and gender, by country, by city, by device.
- Posts - Has information about when fans of the page are online, post types and all posts published.

- Events - Describes awareness (people reached, number of views), engagement (people who responded, event actions), tickets (clicks on buy tickets), and audience (demographics) and shows the upcoming and past events.
- Videos - Shows metrics, like video views, for all videos of the page through time and top videos during a selected date range.
- Messages - Informs about the number of conversations between the page and users.

Summary: Facebook groups the information in tabs which uses mostly line charts to show data, bar charts, comparisons, alerts, annotations to show an overview data, as well, drill down, filters and a button to export data.

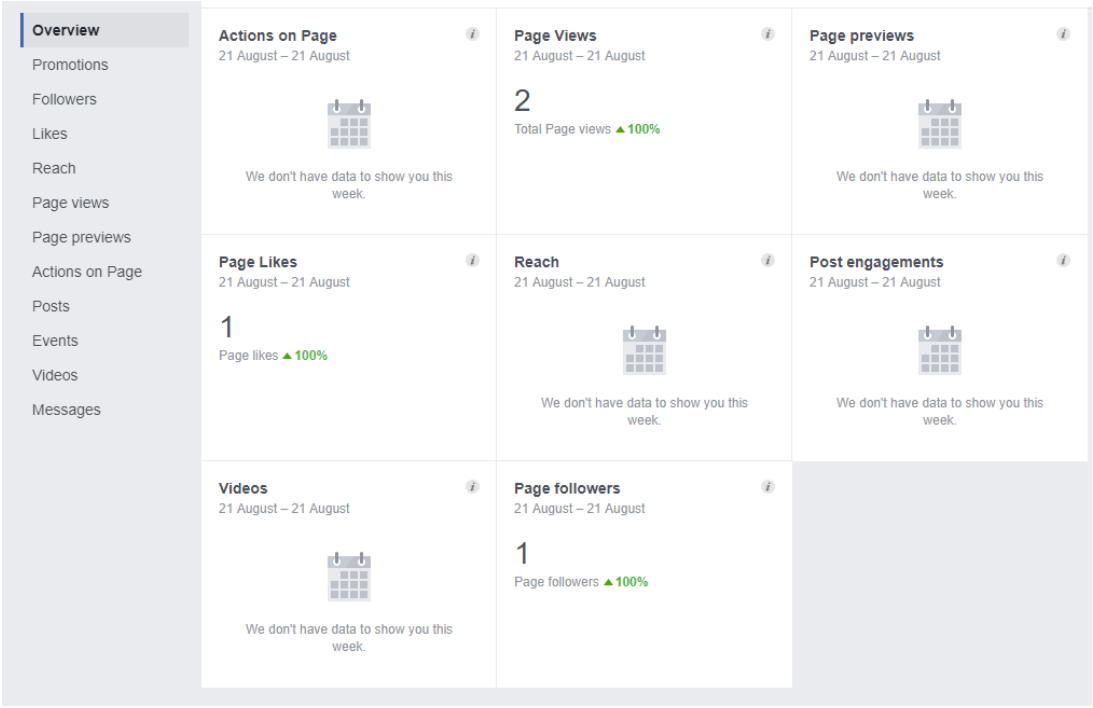


Figure 2.2: Facebook Insights

WordPress is a popular content management system (CMS) where someone can easily start a blog, a website, a portfolio or an online store. Next, the statistical part of one website created on WordPress will be analyzed.

After clicking on ”Stats” option, it gives information grouped in:

- Insights - Here it is possible to take a look on the posting activity during the last twelve months. Also, there is a content for number of views during time, separated by months and years and the

average per day. When there is a new post on the website, information about the number of views, likes and comments corresponding to it and shows quantity information about all posts, views, visitors and the best day of views. The owner knows the most popular day and hour of the website, followers and comments.

- Days, Weeks, Months, Years - For each one, there is a bar chart with the number of views, visitors, likes, comments during time (can be days, weeks, months or even years). It is possible to know from where the visitors access the website, the number of views for each page of the website, search terms, videos, clicks and referrers.

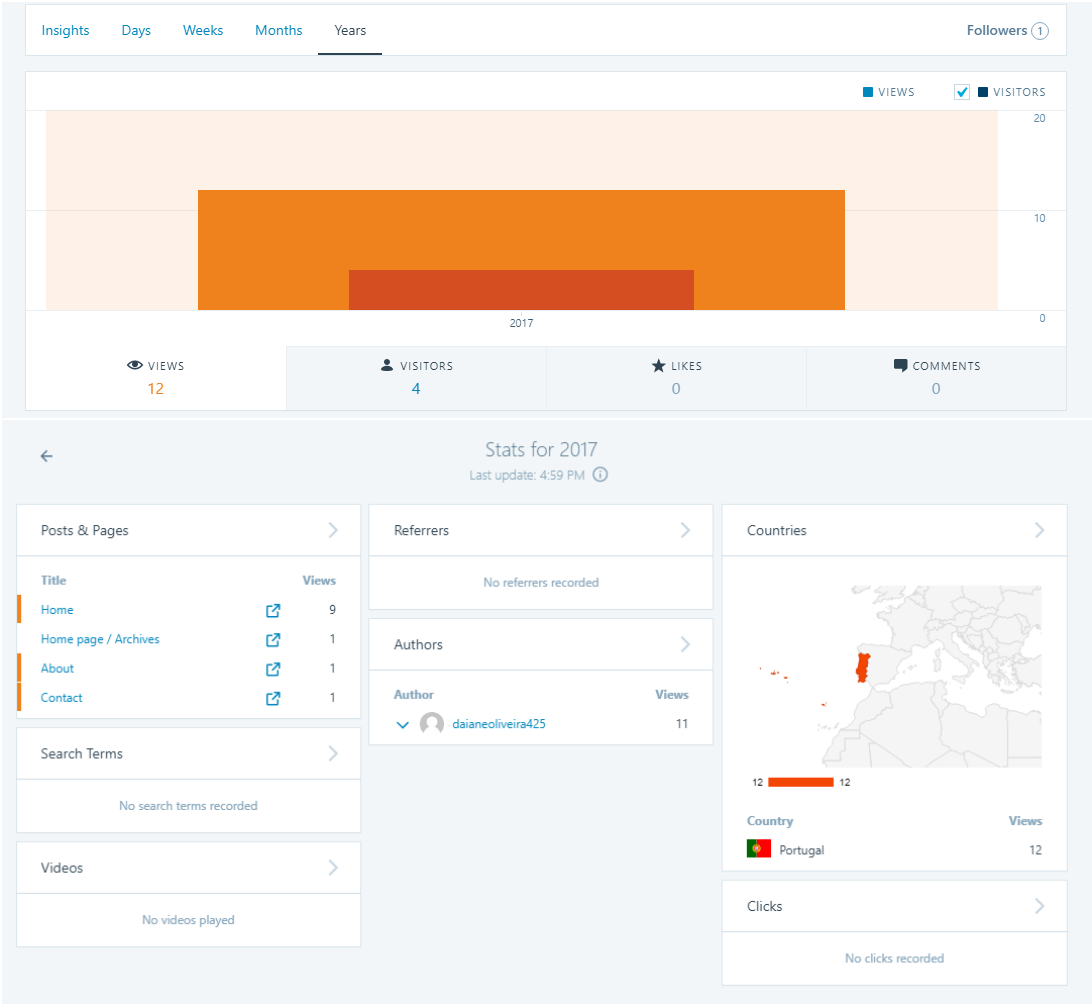


Figure 2.3: Wordpress Statistics

Summary: WordPress uses a lot of annotations, drill down, comparisons, filters, bar charts and a map.

ResearchGate is a social network for researchers and scientists where they can share papers,

formulate and answer questions, find a job and find collaborations.

The statistics are seen on home and profile pages.

- On home, there is content with a summary of the most important metrics since ever, the previous week and from the current week.
- On profile, there is a button called ”Stats”, which displays a set of overview indicators, a line chart with the number of views by time, filter by day or by week.

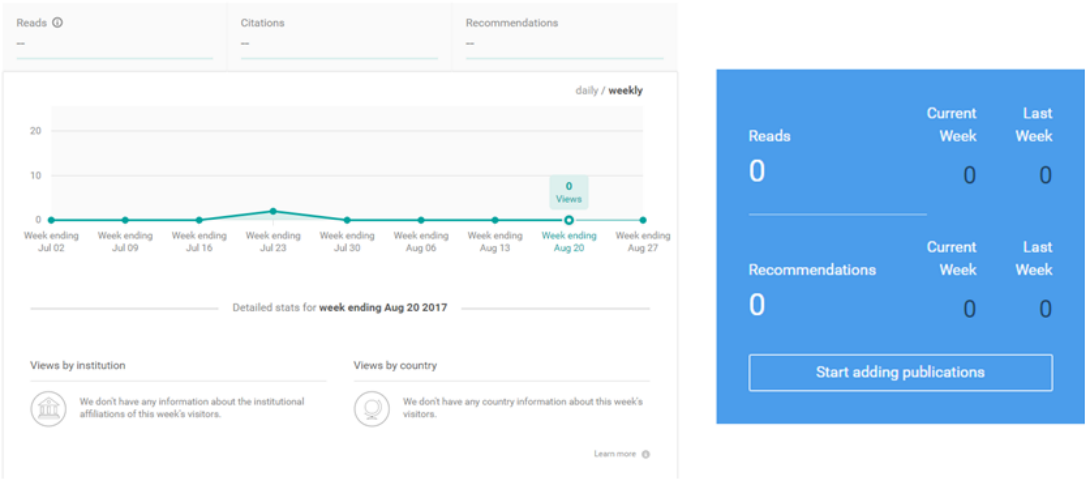


Figure 2.4: ResearchGate Statistics

Summary: This system uses a line chart and annotations.

Twitter is another popular social network where users communicate through messages called ”tweets”.

When activating the statistics, the user has access to a analytics part which has several tabs:

- Home - shows a summary of tweets, impressions, visits, mentions and followers over some date.
- Tweets - shows the tweet activity on a dashboard which contains information related to a selected date range and where it is possible to export data. The information is about the number of impressions over a date as well as the number of tweets published on each date. The first one is represented with indicators and in a bar chart while the second is represented in the bar chart only. Then, there is an area with four filters: Tweets, Top Tweets, Tweets and replies and Promoted. Each one counts the number of impressions, engagements and the percentage of engagement. Also, there exists a set of charts and indicators which detailed the engagements, a line chart for showing the number of engagement rate over time, a bar chart for linking clicks over time and other bar charts for the number of retweets, likes and replies with their totals, simultaneously.

- Audiences - is divided in all twitter users, user's followers and user's organic audience. In the first one there's five tabs (Overview, Demographics, Lifestyle, Consumer Behavior and Mobile Footprint) with information about United States users. The second shows through time the number of followers until now while the third gives information about the interests, demographics, lifestyle and mobile footprint of user's audience

- Events - is about the browser events on Twitter.

- Videos - shows user's published videos over a selected date range.

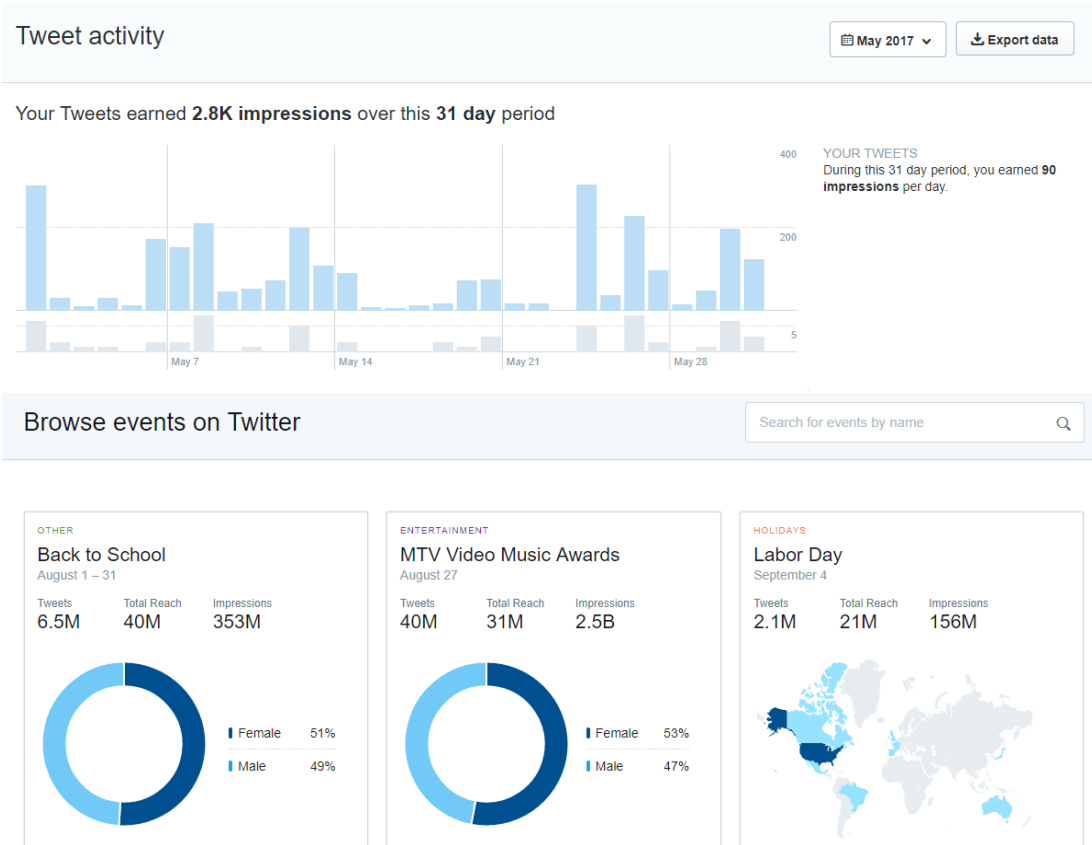


Figure 2.5: Twitter Analytics

Summary: Twitter, taking in account their metrics, create dashboards with charts (donut, line, bar), maps, annotations, summary, filters and export data.

Discussion

After describing the related systems, namely, Facebook, WordPress, ResearchGate and Twitter, the analysis will be present in table 2.5, based on the metrics and dashboard's operations.

Table 2.5: Related Systems Comparison

	Metrics	**Operations**
Facebook	# Actions # Views # Previews # Likes # Reach # Post Engagements # Videos # Page Followers	Filters Line charts Bar charts Comparison Alerts Drill Down Annotations Export
WordPress	# Views # Visitors # Likes # Comments # Followers # Videos # Search terms # Clicks # Referrers	Bar charts Filters Drill Down Comparison Annotations Map
ResearchGate	# Reads # Citations # Recommendations # Views	Line chart Annotations
Twitter	# Tweets # Impressions # Visits # Mentions # Followers # Tweets Engagements Average Tweets Engagements # Tweets Clicks # Retweets # Likes # Replies # Videos	Donut charts Export Filters Bar charts Line chart Annotations Comparison Maps Comparison Summary

As noticed, the operation that all the four systems have in common is the annotation. Then, the most used operations are line charts, bar charts, drill downs and filters.

3

dDocs/Dashboards Requirements

Contents

Before designing and developing the data analysis for dDocs, there are some considerations to take into account, like technical constraints, which represent any of a number of technical issues and obstacles that impact the design and metrics, that are a set of quantitative measures to make comparisons, measurement or track performance.

3.1 Technical Constraints

To ensure that the final result of this work is compatible to the expectations of the involved people, it is important to determine the constraints. They are:

Web-based solution: The client application should be accessed on a web browser.

Responsive: The client application should be compatible with resolutions of different devices.

Angular 2: The client application development should be on Angular 2.

MongoDB: The data system should be kept in MongoDB database during the development.

3.2 Proposed Metrics

Based on the dDocs concepts described in section 1.1 and remembering that dDocs platform has three user roles: citizens, administrator and curators, it is important to highlight the relevant metrics for each of them to start designing the dashboards. These relevant metrics are:

Citizens

- **Number of documents by action over time (last month/last year/since ever):** Total of active documents by action from the citizen in time (last month/last year/since ever).

- **Number of documents over time (month/year):** Total of documents active from the citizen during the time (months/years).

- **Number of documents by document group (life event/document type):** Total of documents active related to a document group from the citizen.

- **Number of documents by curator:** Total of documents active related to a curator from the citizen.

- **Number of documents shared by the citizen over time:** Total of documents active that are shared to others over time.

- **Number of documents shared with the citizen over time:** Total of active documents that were shared to the citizen over time.

- **Number of documents shared with the citizen by a document group (life event/document type):** Total of active documents that were shared, having a document group, to the citizen.

- **Number of documents shared with the citizen by curator:** Total of active documents that were shared, related to a curator, with the citizen.

- **Number of documents shared by the citizen, by a document group (life event/document type):** Total of active documents that were shared by a citizen, having a document group (life event/document type).

- **Number of documents shared by the citizen, by curator:** Total of active documents that were shared by the citizen, related to a curator's document template.

- **Cumulative number of documents over time (month/year):** Total of cumulative documents active from the citizen during the time (months/years).

- **Cumulative number of documents shared by the citizen over time:** Total of cumulative documents active shared by the citizen over time.

- **Cumulative number of documents shared with the citizen over time:** Total of cumulative documents active shared with the citizen over time.

Administrator

- **Number of citizens:** Total of citizens that are active in the system.

- **Number of curators:** Total of curators that are active in the system.

- **Number of inactive curators:** Total of curators that are inactive in the system.
- **Number of templates:** Total of documents templates active in the system.
- **Number of inactive templates:** Total of documents templates that are inactive in the system.
- **Number of documents:** Total of documents that are active in the system.
- **Number of citizens over time:** Total of active citizens that were created over time, in the system.
- **Number of documents over time:** Total of documents that are active over time, in the system.
- **Number of templates over time:** Total of documents templates active over time, in the system.
- **Number of curators over time:** Total of active curators that were created over time, in the system.
- **Number of documents by action:** Total of active documents by action in the system.
- **Number of templates per curator in the current year:** Total of documents templates per curator, in the current year, that are in the system.
- **Number of templates per curator:** Total of documents templates per curator in the system.
- **Number of templates, per curator, over time:** Total of documents templates, per curator, over time, in the system.
- **Number of templates, per document group, per curator:** Total of documents templates, per document group (life event/document type), per curator, in the system.
- **Cumulative number of citizens over time:** Total of cumulative active citizens that were created over time, in the system.
- **Cumulative number of documents over time:** Total of cumulative active documents over time, in the system.

- **Cumulative number of curators over time:** Total of cumulative active curators over time, in the system.

Curators

- **Number of templates:** Total of active document templates that the curator has in the system.

- **Number of documents:** Total of documents created based on the curator's document templates.

- **Number of documents shares with citizens:** Total of documents that are shared in the system with other citizens using the curator's templates.

- **Number of documents shares with curators:** Total of documents that are shared in the system with other curators using the curator's templates.

- **Number of citizens served over time:** Total of citizens that used the curator's document templates over time.

- **Number of templates over time:** Total of created templates that are active, by the curator, over time.

- **Number of accesses by template:** Total of accesses of a curator's template, that is, the number of created documents, by the citizens, based on curator's document templates.

- **Cumulative number of citizens served over time:** Total of cumulative citizens that used the curator's templates over time.

- **Cumulative number of templates over time:** Total of cumulative templates that were created by the curator over time.

4

dDocs/Dashboards Prototype Technical Description

Contents

After understanding the background and knowing which technical constraints and metrics are important to successfully accomplish this work, the current chapter reveals in section 4.1 the architecture, while section 4.2 describes the data model and, last but not least, section 4.3 presents the design and implementation of the dDocs/Dashboards.

4.1 Architecture Overview

Since it must be a web based solution, the architecture will be a client-server style. The figure 4.1 shows the overall architecture with the technologies used in each side of the application and even the database.

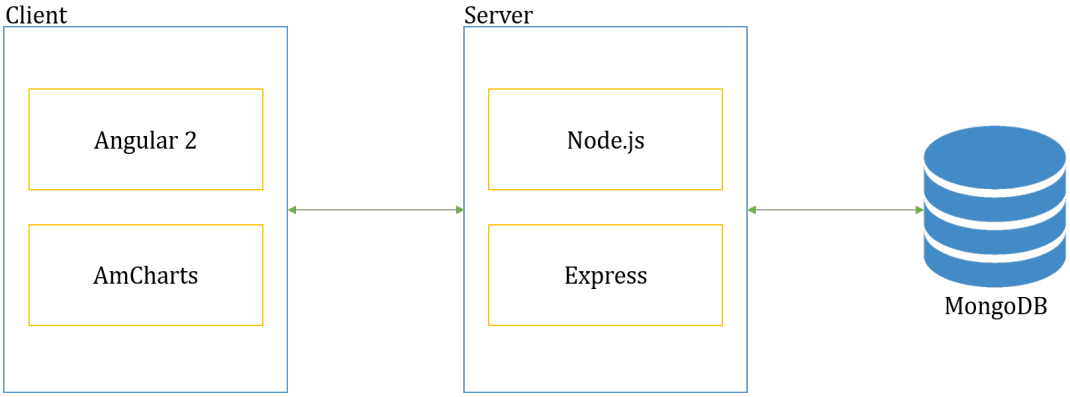


Figure 4.1: dDocs/Dashboards Architecture

Developed on MEAN [1], which is a JavaScript full-stack framework that includes the following technologies:

MongoDB [2] - the database, which is the one that is required.

Express [3] - is a framework for Node.js that simplifies the way to build the server application. Here, it is possible to easily create a robust API since it provides HTTP methods and middleware.

Angular 2 [4] - is an open source framework to write client-side web applications, using HTML where it is possible to extend its syntax and JavaScript MVC to be able to automatically synchronize data from a view. This framework was one of the requirements for this work.

Node.js [5] - asynchronous event driven JavaScript runtime for developing the server-side application.

[1] http://mean.io/
[2] https://www.mongodb.com/
[3] https://expressjs.com/
[4] https://v2.angular.io/docs/ts/latest/
[5] https://nodejs.org/en/

Based on the research and comparison about all data visualization tools mentioned in Background, section 2.2.3, one was picked to be used to develop this work.

The first to be rejected was Tableau, since it has a commercial license. Then, Infogram and Plotly by the reason that they are not recommended for developers, that is, they do not allow the use of visualizations by coding.

Since the amount of data may increase, the zooming feature is necessary, and for this reason, Google Charts and Chart.js are excluded. Also, Chart.js has a limited number of charts, six to be exactly. Data visualization tools like ZingChart, FusionChart, HighCharts are great but they are more indicated for a commercial activity, mostly because they provide a cleaner appearance in the charts.

Although D3.js is well-known for having a substantial number of charts and provide high customization, amCharts is easier to use and offers a limitless customization, interaction and the charts have a clean look. The data visualization tool used in this prototype is amCharts JavaScript library.

Meanwhile, taking into account the comparison of the ETL tools made in Related Work, specifically in the section 2.2.2, Jedox ETL, SAP Data Services and IBM InfoSphere Information Server were excluded because they have a commercial license. Then, Pentaho Data Integration and Talend Open Studio remained.

Since Pentaho Data Integration is more intuitive to use and it is used in other occasions than the last one refereed, it is the chosen one to be part of this work.

4.2 Data Model

In the beginning of the process to accomplish dDocs/Dashboards there is a JSON file, here, each event has the following attributes:

- **Timestamp** - type: datetime. Shows the date and time of the event.
- **Object** - type: string. It describes the type of the object on each event was created. The options are: Document, User and Document Template.
- **ObjectID** - type: number. It is the identifier of the object and it is unique.
- **LifeEvent** - type: string. Represents the name of the life event, can be empty or have one or more events separated by comma, related to the object.
- **DocumentType** - type: string. It is described as the LifeEvent attribute but instead of representing the name of the life event, it has the name of the document type.
- **Action** - type: string. It is the action performed in the object. The options are, Created, Edited, Closed and Shared, when the object is a Document. Otherwise, the only option is to be empty.

- **OwnerID** - type: number. It is the identifier of the user that shared a document with the user who receives it. If there is no shared document by other citizen, this attribute is null.
- **OwnerName** - type: string. It is the name of the user that shared the document with the user who receives it. If there is no shared document, this attribute is empty.
- **User** - type: number. It is the user identifier on each the object suffered a change.
- **IsAdmin** - type: boolean. Informs if the user is the dDocs administrator or not.
- **IsCurator** - type: boolean. Informs if the user represents a curator or not.
- **OrganizationName** - type: string. Describes the name of the curator.
- **OrganizationId** - type: number. Identifies the number of the curator.
- **Name** - type: string. Informs about the name of the user who created the event.
- **Password** - type: string. It is an encrypted password for logging into the system and is empty when the object is not a User.
- **State** - type: string. Represents the current state of the object in the event. The options are: Active or Inactive.
- **TemplateName** - type: string. It is the name of the template. It is not empty except when the object is a User.
- **DocumentName** - type: string. It is the name of the document. Only appears when the object is a Document.

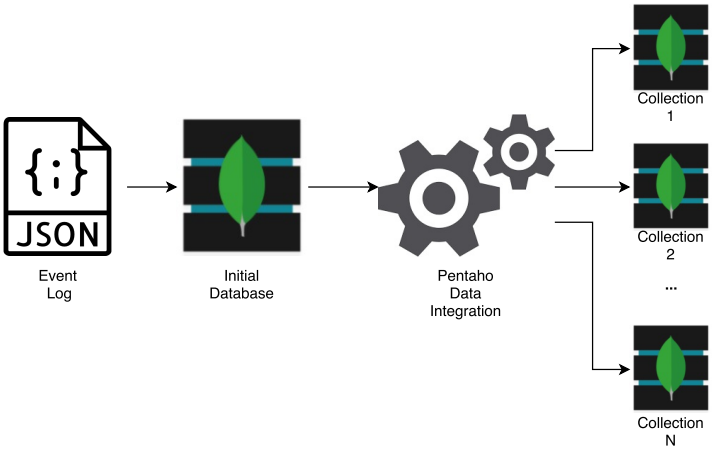


Figure 4.2: Data Processing

As represented in the figure 4.2, the data processing starts by using a script, the JSON file data was inserted on a MongoDB collection. After that, through Pentaho Data Integration (PDI), data suffered the necessary transformations and was inserted in other MongoDB database. Specifically, in the following collections:

- **DocumentsActions** { _id, User, Year, Month, Action, NDocs }
- **LifeEvent** { _id, LifeEventName, User, NDocs }
- **DocumentType** { _id, DocumentTypeName, User, NDocs }
- **Curators** { _id, User, OrganizationName, NDocs }
- **MyShares** { _id, OwnerID, Date, Name, NDocs }
- **LifeEventShared** { _id, OwnerID, LifeEventName, NDocs }
- **DocumentTypeShared** { _id, OwnerID, DocumentTypeName, NDocs }
- **CuratorsShared** { _id, OwnerID, OrganizationName, NOrgDocs }
- **SharesReceived** { _id, User, Date, NDocs }
- **LifeEventSharedWithMe** { _id, User, LifeEventName, NDocs }
- **DocumentTypeSharedWithMe** { _id, User, DocumentTypeName, NDocs }
- **CuratorsSharedWithMe** { _id, User, OrganizationName, NOrgDocs }
- **Citizens** { _id, Date, NCitizens }
- **TotalCurators** { _id, Date, NCurators, State }
- **Documents** { _id, Date, User, Name, IsAdmin, IsCurator, NDocs }
- **Templates** { _id, Date, OrganizationName, State, LifeEventName, DocumentTypeName, NDocTemplates }
- **Totals** { _id, NCreatedDocs, NEditedDocs, NClosedDocs, NSharedDocs, LastAddedCurator, LastAddedDate, LastRemovedCurator, LastRemovedDate }
- **NTemplates** { _id, Date, TemplateName, OrganizationID, NTemplates }
- **AccessedTemplates** { _id, Date, OrganizationID, TemplateName, NAccesses }
- **NCitizensShares** { _id, Date, OrganizationID, NCitizensShares }
- **SharedWithCurators** { _id, Date, OrganizationID, NShares }

- **CitizensServed** { _id, Date, OrganizationID, IsCurator, Action, State, NCitizens }

After all that, these collections were used for constructing several dashboards for different user permission. The table 4.1 presents which collections compose the dashboard for a citizen, the administrator and a curator.

Table 4.1: MongoDB Collections Used In Each Dashboard

	Collections
Citizen	DocumentsActions LifeEvent DocumentType CuratorsMyShares LifeEventShared DocumentTypeShared CuratorsShared SharesReceived LifeEventSharedWithMe DocumentTypeSharedWithMe CuratorsSharedWithMe
Administrator	Citizens TotalCurators Documents Templates Totals
Curator	NTemplates AccessedTemplates NCitizensShares SharedWithCurators CitizensServed

4.3 Design and Development

dDocs events is in a JSON file, then, consequentially, it is maintained in MongoDB NoSQL database and, after that, it is processed and clean up to be kept in MongoDB collections. Each collection has key-values about the data that were needed to show in the dashboards.

The data processing and cleaning is made on Pentaho Data Integration, which is a drag and drop environment that is responsible for Extract, Transform and Load (ETL) processes. It has some features like the capability of migrating data from different data sources, it is able to integrate applications, perform data cleaning and load a considerable amount of data.

In the server-side application, with Mongoose (MongoDB object modeling for Node.js) it is possible to query what is needed and then creating the API with the result of each query with Express.

In the client-side application, there is component that acts as a service that receives the content of HTTP page with the related REST API to create a function. Afterwards, these functions will be subscribed on the component of the application that will use the data. The components resort Underscore.js, jQuery, amCharts and they are constructed under Bootstrap.

4.3.1 Citizen Interface

This interface has three tabs:

- My Documents/"Meus Documentos" - contains information related to his own documents.
- My Shares/"Minhas Partilhas" - shows information about the documents that have been shared by the citizen.
- Documents Shared With Me/"Documentos Partilhados Comigo" - displays information about the documents that are shared with the citizen.

Inside the My Documents/"Meus Documentos" tab, the citizen can get information about his number of documents by action - Created, Edited, Closed, Shared (with others) - from last month, last year and since ever. Below, there is a line chart that describes the number of documents owned by him through time, which gives two option: by year or by month. Each chart gives two informations: the number of documents at each date and the cumulative documents that exist till that date. The Citizen can be able to disable the information. Following, there is a set of donut charts where it is possible to explore the information related to each document group (life event, document type) and curators. In the document group the citizen selects if he wants to take a look at information related to life events or document types which shows the life event/document type name and the number of documents that the citizen has about this event or type. The curators donut chart means that the documents owned by the citizen were from some curator template. So, it contains the name of each curator with the related number of documents. The figure 4.3 presents this page of a citizen's dashboard.

After selecting the My Shares/"Minhas Partilhas" tab, a line chart related to his shares is shown. So, a visualization of the number of documents shared by him at some date, as well as the cumulative documents that were shared by him until that date is shown. The citizen may disable each information and zoom it. Then, there are three donut charts where he can select which document group he would like to observe and another about curators, which count the number of documents shared by him related to some curator. After that, there is a table to show the top three citizens that received more documents shared by him. The figure 4.4 presents this page of a citizen's dashboard.

The last tab, Documents Shared With Me/"Documentos Partilhados Comigo", has the same structure as the My Shares/"Minhas Partilhas" tab. The difference is that, instead of showing information related to his documents shares, it is about the documents that he received by others citizens. The figure 4.5 overviews this page.

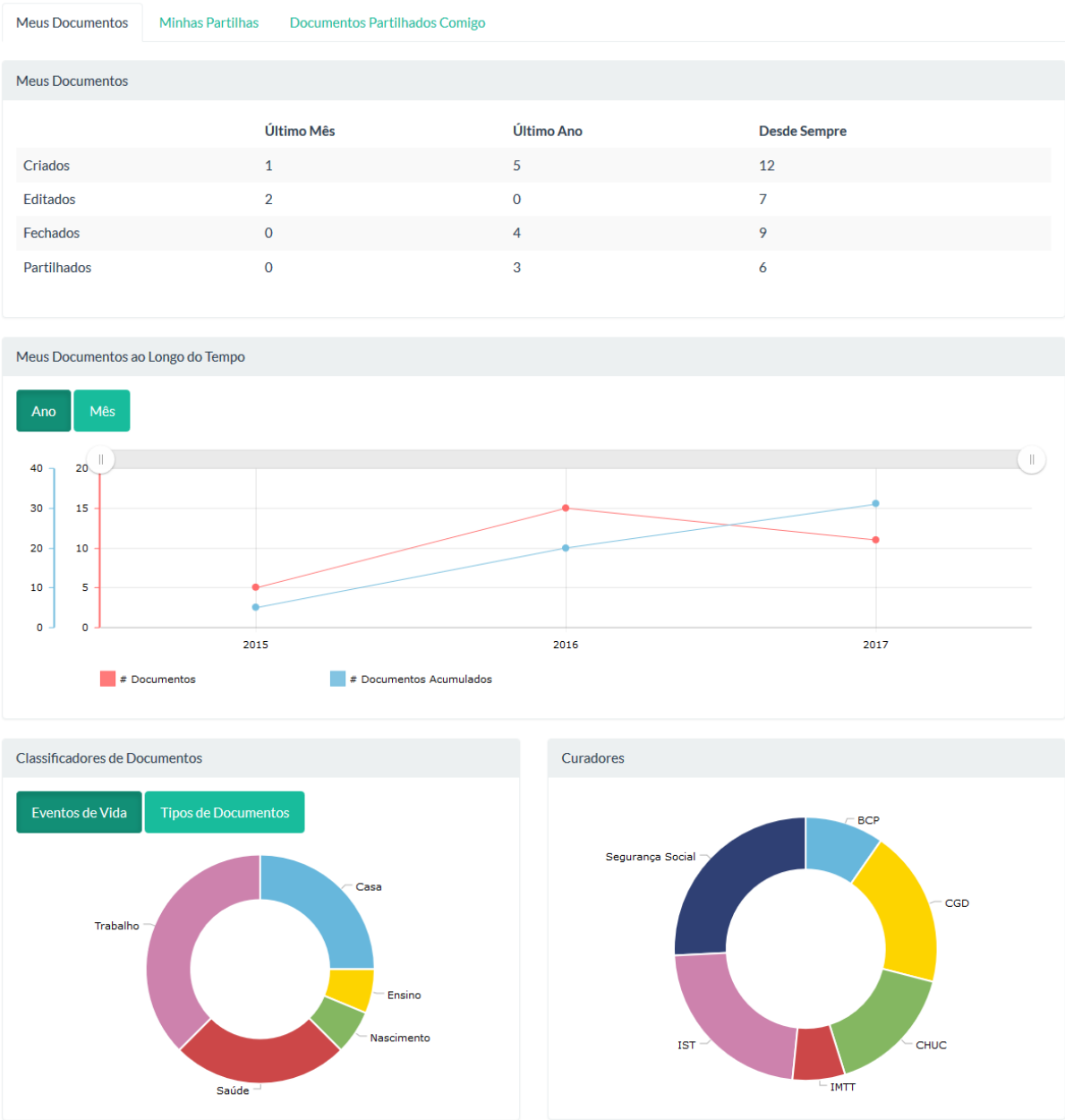


Figure 4.3: Citizen: My Documents Page

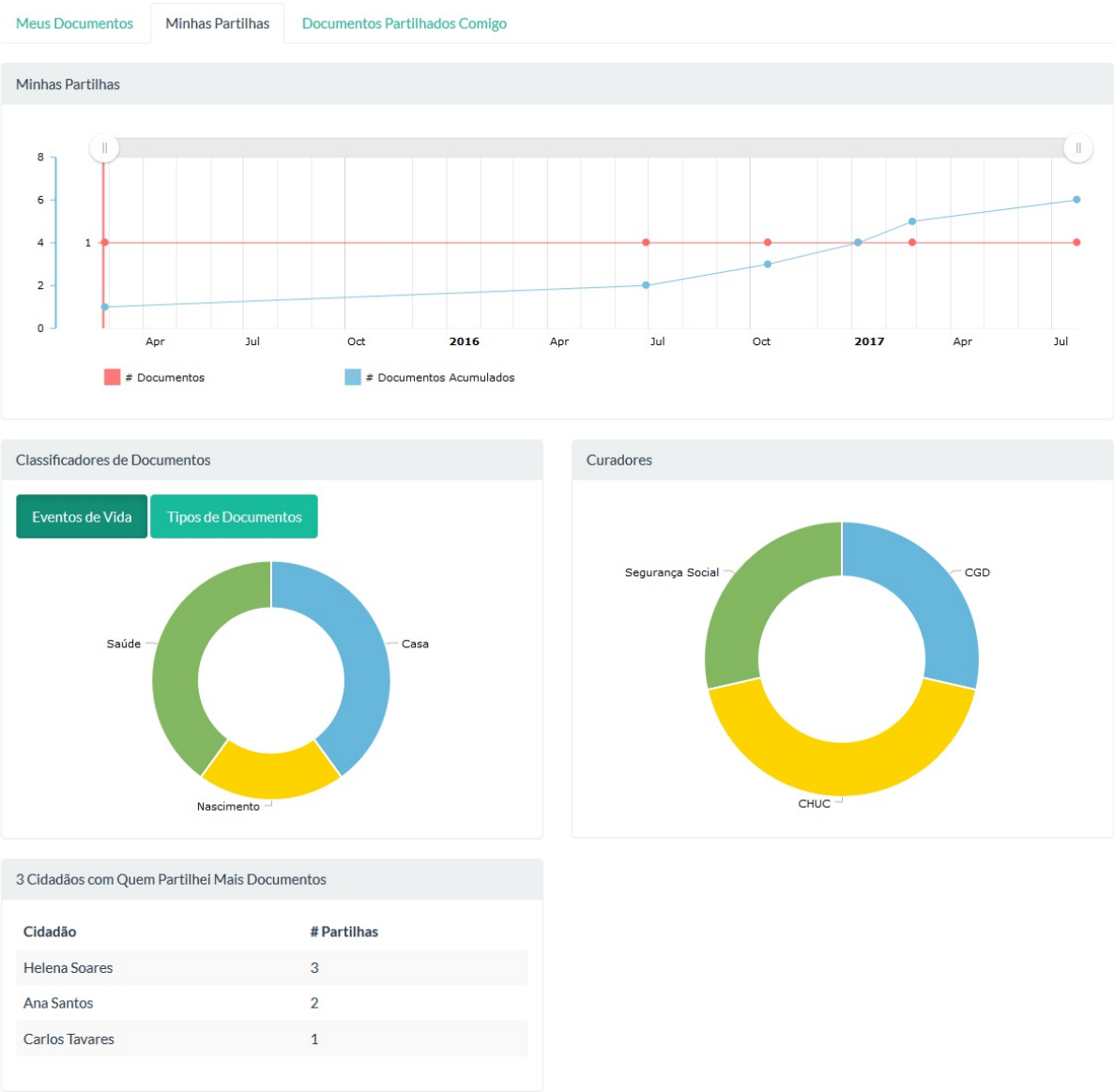


Figure 4.4: Citizen: My Shares Page

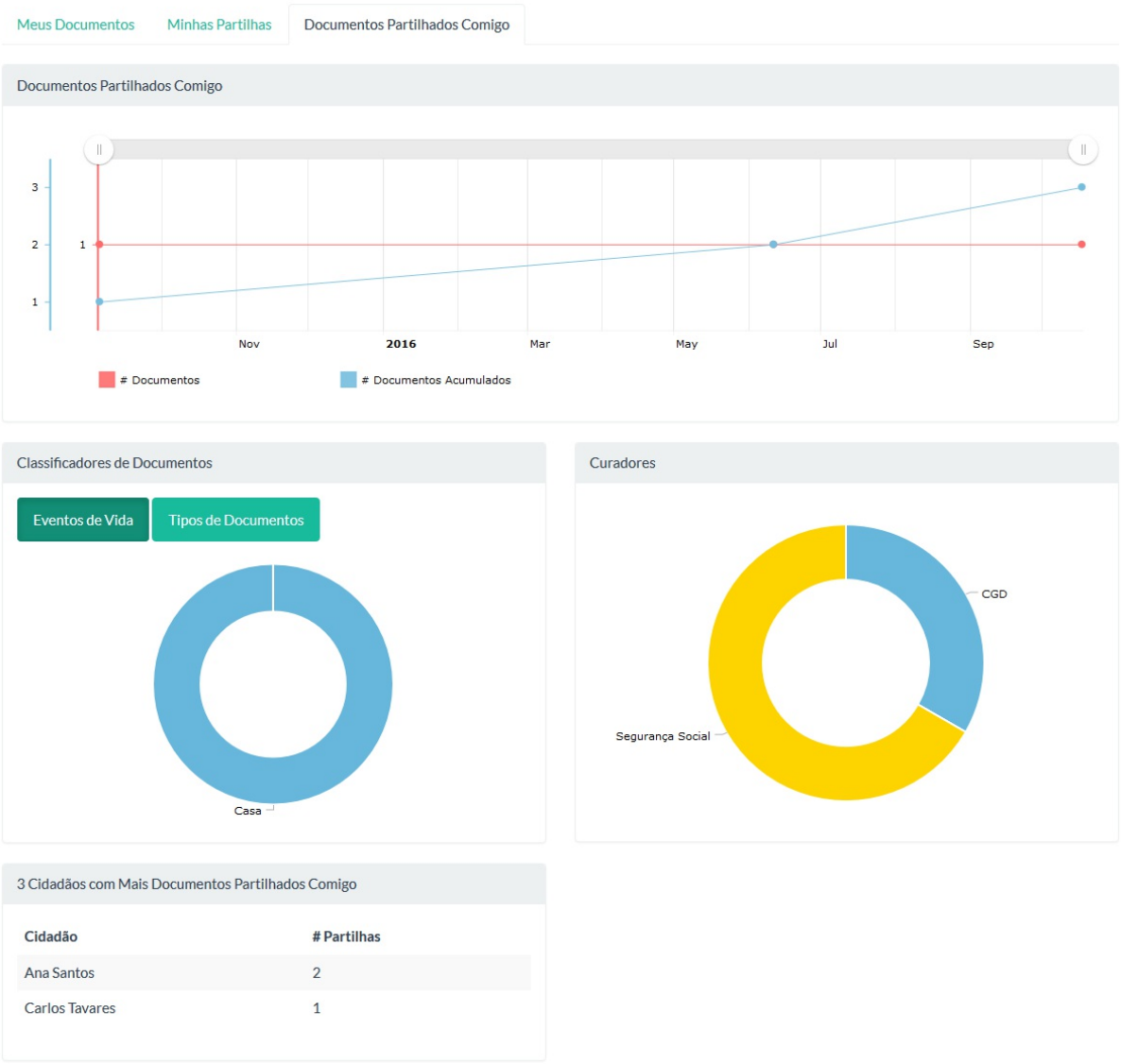


Figure 4.5: Citizen: Documents Shared With Me Page

4.3.2 Administrator Interface

Another role in the system is about the administration where someone from dDocs is responsible to manage the system and which implies that there must be a dashboard for this kind of user.

Initially, there are several overviews at the top that indicate the total number of citizens in dDocs, the total number of curators, the total number of inactive curators, total number of documents, total number of templates and the total number of inactive templates and the corresponding percentage of each that were created during the last week.

Below, a set of informations about the citizens is represented, which include a line chart for indicating the number of citizens that were created on the system at some date and in parallel there is a line that sums the number of citizens since the beginning until at some date that the administrator would like to know. Here, he can select what he wants to see, or the number of citizens through time, or the cumulative number of citizens, or both or none.

In this dashboard, the top four citizens that have more documents in the system are shown, where if the administrator hovers on the green area of each citizen, the percentage of documents that each have in the system appears.

As the citizens are responsible for creating documents, in this area, there is also another line chart but now the information is about the number of documents in the system at some time as well as the cumulative number of documents through time. The administrator can perform the same actions as the other line charts mentioned above.

Then, there is a table which contains the total number of documents distinguished between actions (created, edited, closed and shared) in the platform.

The next section in about the information related to curators, which have two line charts, one that represents the number of curators created at some date in the system and the other that represents the number of templates that were created (since only the curators can manage the templates) in the system across time, where in each chart there is the cumulative number of the respectively represented object. Here, the administrator can manage what information he wants to see. The line chart, that represents templates over time, changes when the administrator selects a curator in the drop down in the templates per curator. The default data in this line chart is about all the curators. There are, also, two donut charts, one for letting him know which are the top 3 curators in the system, that is, which curators have more templates in dDocs and the other donut chart is about the curators that have templates at this current year, where the administrator can disable or enable which curators he would like to be in the donut chart.

Also, in the templates per curator, the administrator can select the curator, active in the system, that he would like to obtain an information, then, the total number of templates created by the selected curator appears. If he would like to obtain more detailed information about this number of templates, he can click on the More Info/"Ver Mais" button that displays the number of templates with the corresponding

life event or document type.

At last, there is an overview information about the last added curator and the last inactive curator.

The figure 4.6 shows an overview over this dashboard.

4.3.3 Curator Interface

Here, someone responsible for managing his organization will have, on top of this dashboard, an overview over the number of created templates by the curator, the number of documents created based on his templates, the number of documents that has been shared with other citizens and the number of documents that has been shared with other curators.

Then, the curator will have access to a line chart that relates the number of citizens that has been served by curator's templates and the dates where it happens. The line chart provides two informations: the number of active citizens created at some date and the cumulative number of citizens, in the same situation, until some date. If the curator won't take a look at some of these two informative option, he can simply disable the desired one. The chart provides the zooming functionality.

Another line chart is presented below to show the number of templates created through time as well as the cumulative number of templates at a date. Here, he can disable and enable what he wants to see and can zoom it the way he wants it.

Next, there is information about the access on his templates. In one hand, there is a donut chart that lets the curator know which of his templates have more access, that is, which templates were transformed into documents by the citizens. In the other hand, there is another donut chart that represents the templates that have less access. In both, the number of accesses in each template is specified. Also, the accesses on the templates can be visualized in a line chart. On here, the curator may choose what available template he wants to see the number of accesses over time. The default option is all templates.

In the bottom is located a list that has his five recently created templates. In each content of the list is the name of the template and the respective date of creation. In this list, the most important is to understand which of them is the last created template, for this reason, it is highlighted. The figure 4.7 overviews a curator's dashboard.

Discussion

All these dashboard are responsive, that is, they are prepared to be visualized in different screen sizes of user's devices and the language used was Portuguese for the fact that dDocs is a system where the audience is Portuguese.

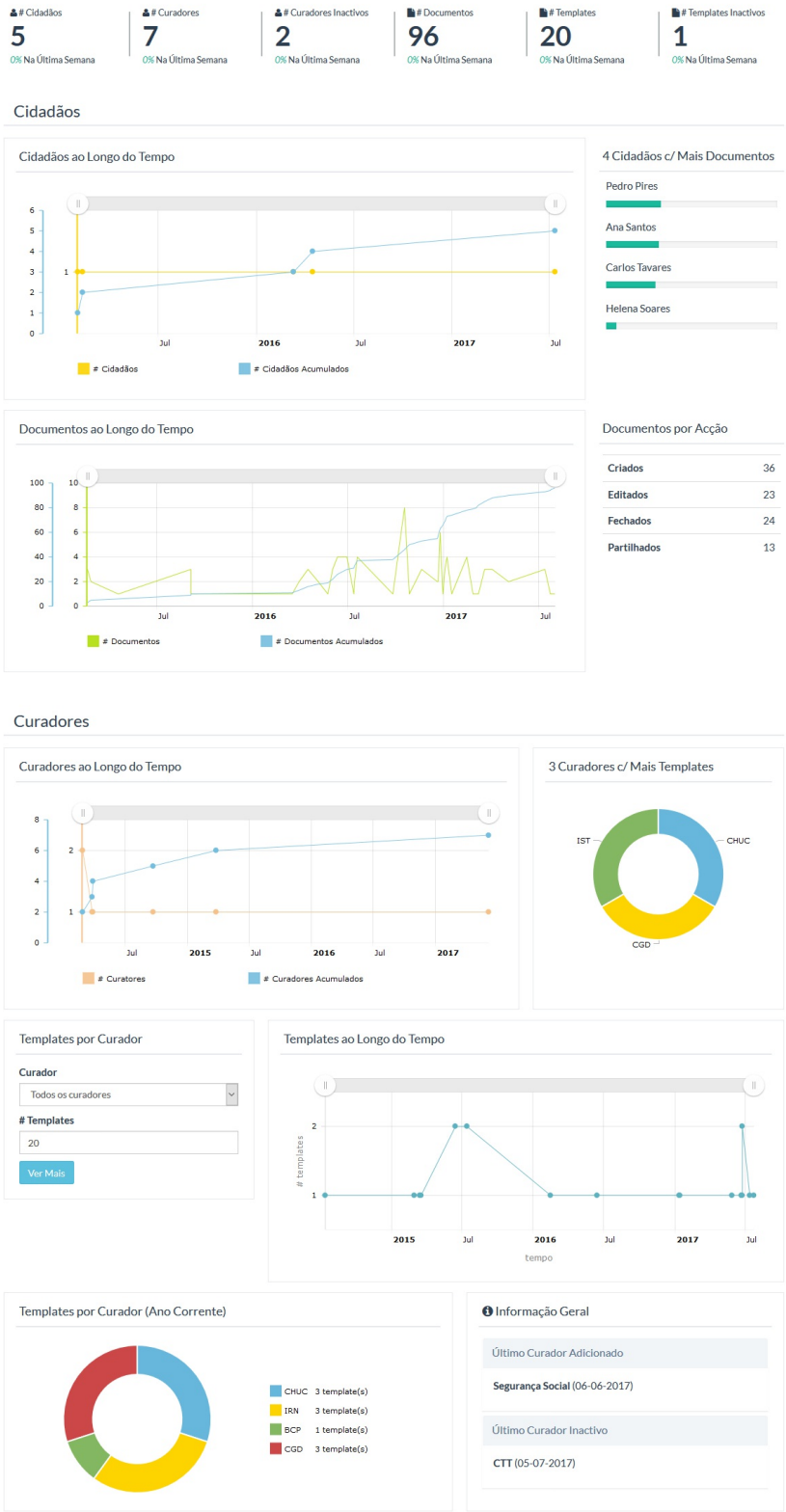


Figure 4.6: Administrator Dashboard

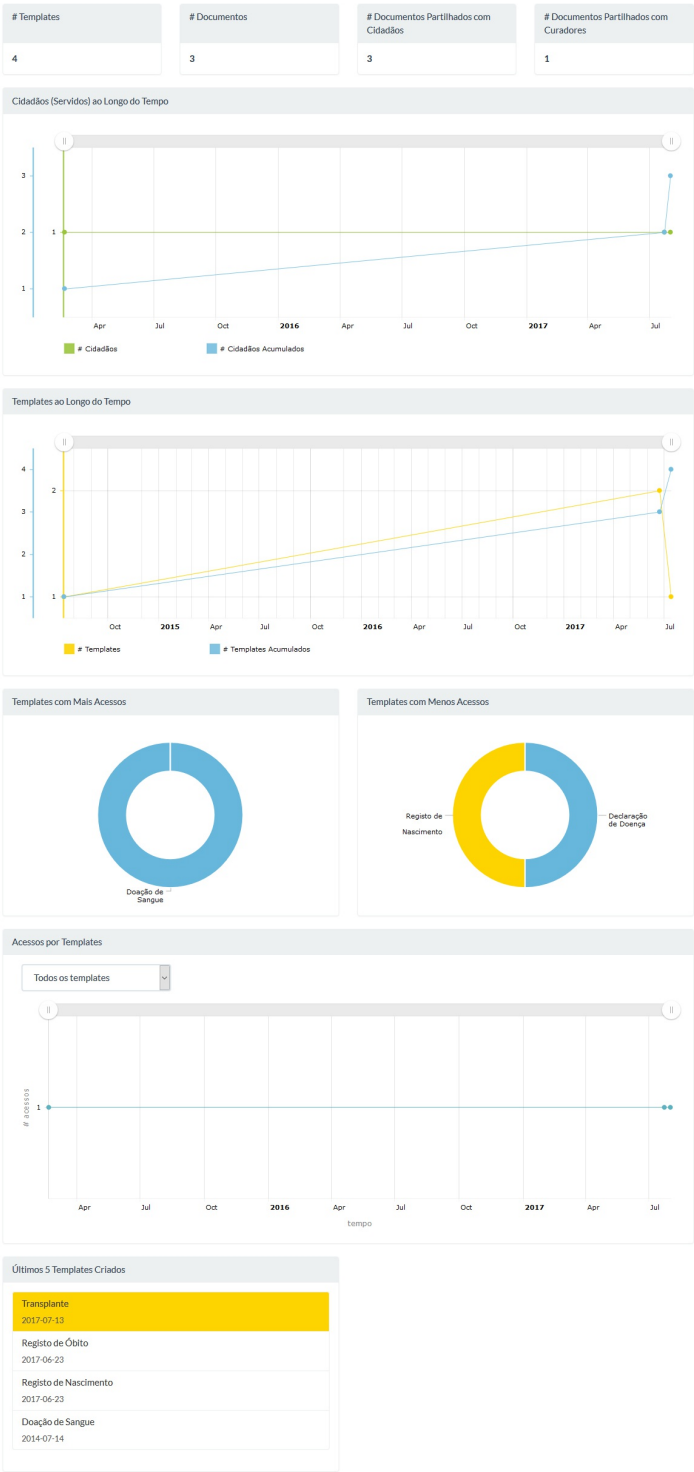


Figure 4.7: Curator Dashboard

Meanwhile, line charts are good for trends. Here, each line chart shows the evolution of information over time.

Donut charts show parts of a whole. The use of donut charts is to present the distribution of life events/document types/curators based on the number of documents/templates/accesses. For instance, in this visual there is a fast knowledge which has more documents, consequentially, also, knowledge of which has less documents.

5

Evaluation

Contents

This chapter delineates two evaluations based on performance test and usability test. The section 5.1 shows the time that takes to load all data related to different dashboards while, section 5.2 presents the dashboards evaluation made by a few participants.

Furthermore, in which of these two kinds of tests is present all the aspects of the process that are involved.

5.1 Performance Evaluation

The performance of the different dashboards in the system is measured. To accomplish this measurement, it was resorted a script in C# that populates each MongoDB collection, mentioned in section 4.2, with a controlled random data (to make sense when analyzed it on dashboards), for a better performance was implement for each MongoDB collection indexes. This test has four scenarios:

1. 1000 BSON documents.
2. 10 000 BSON documents.
3. 100 000 BSON documents.
4. 1 000 000 BSON documents.

The results are separated by dashboards, namely, Citizen which contains three pages: My Documents/"Meus Documentos", MyShares/"Minhas Partilhas" and Documents Shared With Me/"Documentos Partilhados Comigo", Administrator and Curator.

Firstly, were tested a set of collections responsible for showing the information on My Documents/"Meus Documentos" for each scenario. The results are represented in the figure 5.1.

Analyzing the figure 5.1, the time that takes to load all the dashboard increases with the number of BSON documents in each collection. Even with 1 000 000 BSON documents, the time to load is good which proves that MongoDB is capable to handling very fast with a considerable number of data.

During the visualization of the loading time, in the My Documents/"Meus Documentos", the table takes more time to show numbers after 100 000 BSON documents. The reason why can be in some calculations made in JavaScript.

In the next dashboard My Shares/"Minhas Partilhas", the related collections were filled based on each scenario. The performance to load all the content in the dashboard is good. With 1 000 000 BSON documents took more time than expected, which could be explained by the fact that JavaScript math operations happen simultaneously. Figure 5.2 shows the results.

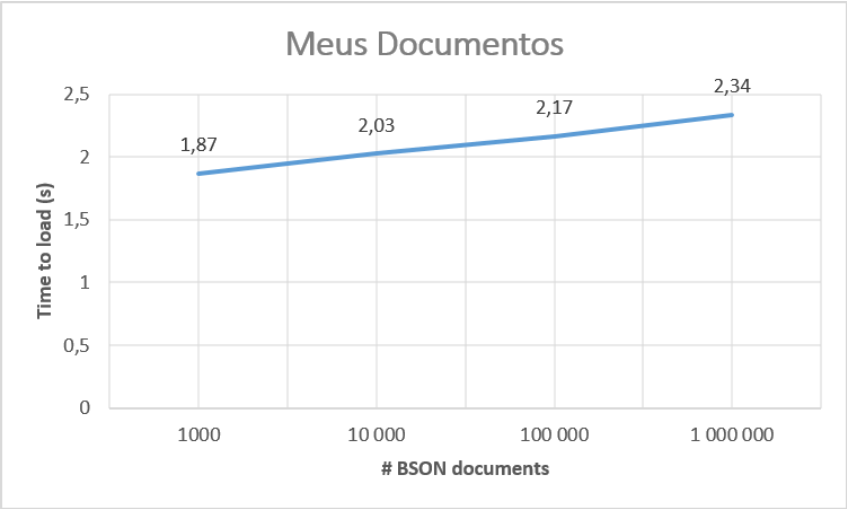


Figure 5.1: Performance Evaluation/Citizen: My Documents Page Results

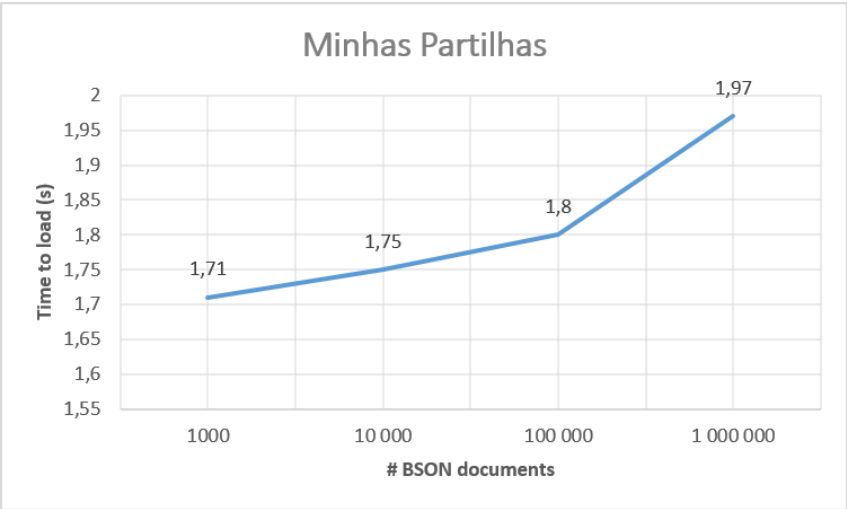


Figure 5.2: Performance Evaluation/Citizen: My Shares Page Results

The third tab in the citizen environment, Documents Shared With Me/"Documentos Partilhados Comigo", all the related collections were prepared for the four scenarios and the results are presented in the figure 5.3. The load time is similar as the other pages in citizen's dashboard and the results are equal as the others.

The figure 5.4 shows the results related to the performance tests for the administrator's dashboard. Here, one more time, the scenarios were done in the collections that belong to this dashboard. Until the third scenario, everything loads normally, afterwards, the line charts suffer a little clutter, since it has two y axis, the right one values were on the other axis line. Also, it took more time to load the content, when occurring the fourth test scenario.

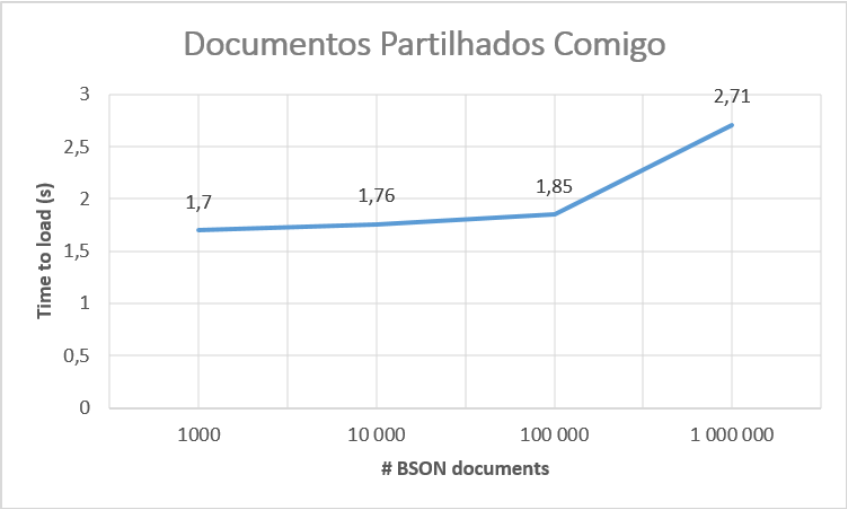


Figure 5.3: Performance Evaluation/Citizen: Documents Shared With Me Page Results

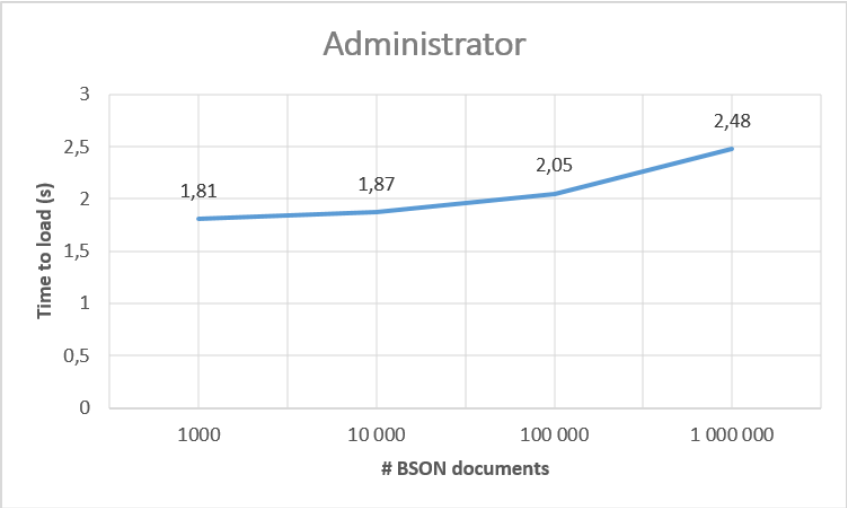


Figure 5.4: Performance Evaluation/Administrator Results

The last dashboard that was tested for performance was about the curators. Testing in one of them. The load time increases as the number of records in each related collection increases. Overall, it has a good performance. The same as the other dashboards happens, that is, after 100 000 BSON documents inserted in each collection, the line charts suffer a clutter in the values of the y axis, also, took more time to load the data, the reason can be the calculations made in the client application to show some of the data. The load time results for each scenario are in figure 5.5.

In general, the performance of dashboards, when increasing the number of records in the MongoDB is linear.

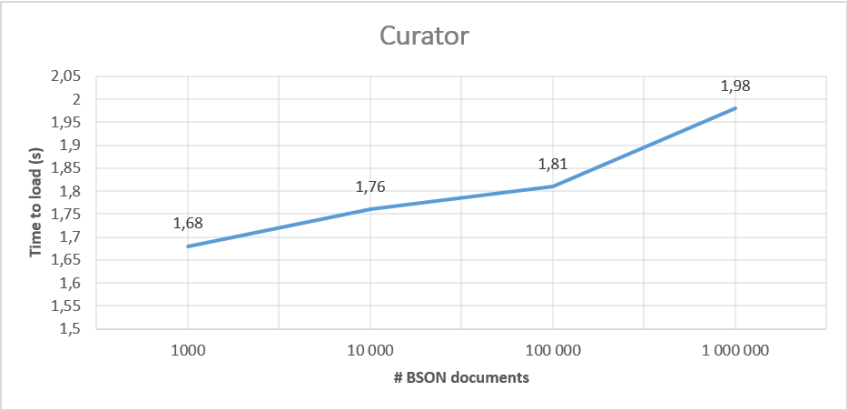


Figure 5.5: Performance Evaluation/Curator Results

5.2 Usability Evaluation

To gathered information related with the user experience to this work, it was made a session where it was explained all the process until the performance test. Here, all the involved users took a look in all dashboards, where they were able to formulate questions and explore all the dashboards.

This session counted with the participation of four people, the lower number of users is explained by the reason that the understanding of this work is not easy and implied to combine a large number of inexperienced participants to be reunited in one session before the test to elucidate them about the scope where it is in. Also, Jeff Rubin and Dana Chisnell described in [15] that all participants should have nearly the identical experience. So, the participants are people involved in dDocs project.

In the end, to easily analyze the opinions, it was sent to each of these users a questionnaire to, briefly, answer seven questions:

Q1. How do you evaluate the dashboards?

Q2. How easy is it to understand the information presented in the dashboards?

Q3. How consistent are the dashboards?

Q4. How clean are the dashboards?

Q5. Do you think these dashboards are useful to dDocs platform?

Q6. What dashboard page or pages you like the most?

Q7. Do you have any suggestions?

The last question is optional, while the others are required to answer.

Table 5.1 reveals the average score for the answers regarding the first four questions above, broken down by question. In overall, these questions had a positive score.

About the question 1, the participants think the dashboards are good. In the second quetsion, dDocs/Dashboards is classified by three participants as extremely easy to use while one participant classified as easy. Meanwhile, in the third question, 75% of the participants think that the dashboards are consistent and 25% think they are very consistent. In the question 4, the answers were divided into clean and very clean.

Table 5.1: Questionnaire Average Score (1-5 Scale) By The First Four Questions

	Q1	Q2	Q3	Q4
Average	4.5	4.75	4.25	4.5

In the question 5: Do you think these dashboards are useful to dDocs platform?, the participants had to select one of the two options: yes or no and the results are that 100% of the users consider that the dashboards are useful to dDocs. The figure 5.6 presents the answers.

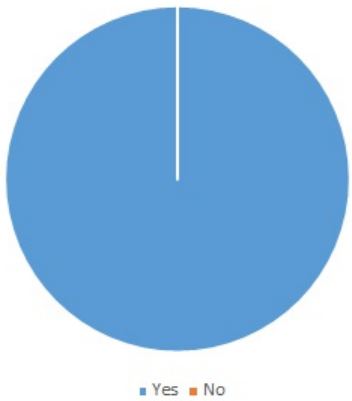


Figure 5.6: Usability Evaluation/Questionnaire: Q5 Results

In the question 6: What dashboard page or pages you like the most?, the participants were enabled to select all dashboards, none or a specific dashboard page. The responses are in figure 5.7.

As presented in figure 5.7, the favorite dashboard is related to the curator (100% of the participants agree), the next preference is about the administrator's dashboard and 50% of them liked the citizen's dashboards. Actually, one of the participants selected his preference in all the dashboards.

The last question, which is related to give suggestions was answered by 50% of the users. Here the suggestions are: have incremental log imports to reduce the analyzed period and to change and, in the administrator's dashboard, from absolute numbers to percentage when distinguishing the number of documents per action.

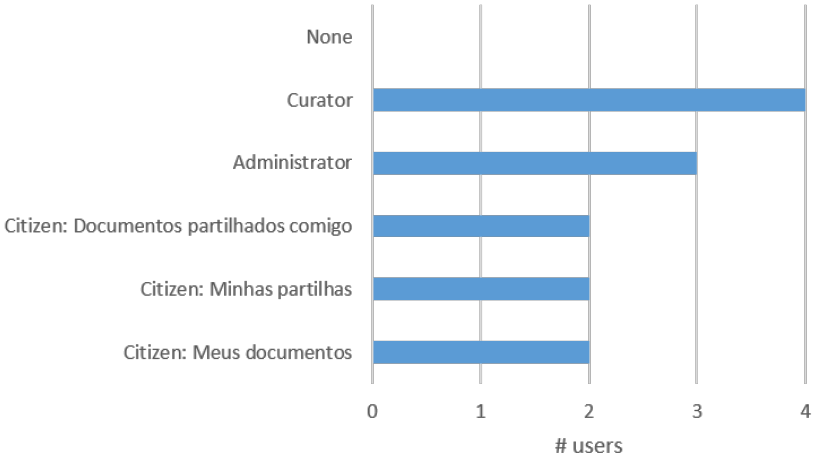


Figure 5.7: Usability Evaluation/Questionnaire: Q6 Results

To sum up, although the number of participants is small, these answers elucidate that this work received a good feedback. The first four results of the questionnaire are presented in Appendix A.

6

Conclusion

Contents

This chapter presents the conclusion about this work as well as some recommended directions for the future of the work.

6.1 Conclusion

Over time, the increasing use of platforms and the fast development of them, raise new concerns. One of them is related with the data, since years ago, there was an exponential increase of it that brought a lack of space and to solve it a well-known term, Big Data, appeared to allow large and complex datasets.

MongoDB is one of many existing NoSQL database systems. They are capable of supporting these large and complex datasets mentioned above. So, for that characteristic and, also, defined a requirement, during the development, MongoDB is used.

Also, e-Government and Record Management are fundamental areas that were merged to serve as a base to dDocs platform. Being a system that provides communication with different users roles, the resulting data could be transformed into something valuable for the system.

Throughout history, data visualization had a huge impact and, nowadays, are commonly used in Business Intelligence and Business Analytics areas. Having the necessity to expose understandable information through visualizations, many tools have been created to support the implementation of the desirable data into many types of charts and maps.

Between the data visualization techniques, the most popular one is dashboard. One well-known definition is that dashboards are ”a visual display of the most important information needed to achieve one or more objectives, consolidated and arranged on a single screen so the information can be monitored at a glance”. Before starting to build dashboards, it is important to understand who is the audience and what they need, as well as what it is essential to define the most significant metrics to each role. Furthermore, there are some considerations to take into account related to the design of the dashboards, choose the right visualizations, choose the proper use of colors and an effective dashboard must be a responsive and contain an easy-to-read information.

In this dissertation, based on the resulting data from dDocs platform, is presented a set of dashboards each one for a user’s role, which are: citizens, dDocs administrator and curators, taking into account the previous research and requirements. The process of development has passed by inserting the initial data in a MongoDB database, then, with Pentaho Data Integration the data was transformed to be stored in another MongoDB database. Having the conditions to proceed, the development of the dashboards with different technologies, being relevant the use of some JavaScript libraries, such as, underscore.js and amCharts.

This work has been evaluated in order to check its usefulness and quality and this evaluation took

two strands, performance evaluation and usability evaluation. To get the results of performance test was made a script that filled each MongoDB collection with different scenarios followed by measuring the time to load the data in each dashboard and observe the dashboard's modification, in case they happen. The results of the usability test were captured through a user session were the participants explored the different dashboards and answered a questionnaire to assess the opinions of the participants over the look and consistency of the dashboards. In both evaluations, the results collected were positive and showed evidence that demonstrate the usefulness and feasibility of the data analytics on dDocs and support the thesis behind this dissertation. Moreover, the goals in the section 1.3 were accomplished.

The following section, 6.2, describes the possible directions to take in the future work.

6.2 Future Work

This section describes a few directions to the future research about this work. During this work some ideas were appearing but reasons as time constraints or complexity do not turn to be implemented, it is important to notice that the main goals were accomplished in this work. Therefore, the future research directions could be:

Incremental log imports: this was one of the suggestions made in the usability test. This means, instead of processing data since the first event of the log, when new events are inserted, this data should be in another log to be analyzed, resulting in a performance improvement in the ETL tool.

Export data: create a button at the top of each dashboard page to export data as .csv, for instance, for advanced users to manipulate the data for other purposes.

Add a network graph: In the administrator's dashboard could be implemented a network that shows the relation between the curators or even in the curator's dashboard to show the relation of his templates to the rest of the curators that may receive them as a document.

Add a map: In the administrator's dashboard for specifying the number of curators in each district of the country.

Report: add a new button in the dashboards to generate a pdf to save and have the report of user's dashboard.

Bibliography

[1] R. Santos, “dDocs - A Citizen-Centric Electronic Document and Record Management System,” *MSc Thesis, Instituto Superior Técnico*, 2016.

[2] F. Yao, “A comparative analysis of administration-oriented and customer-oriented electronic government,” in *Management and Service Science (MASS), 2011 International Conference on*. IEEE, 2011, pp. 1–3.

[3] International Organization for Standardization, “Information and documentation - Records management - Part 1: Concepts and principles,” *ISO 15489-1:2016*.

[4] MDSS, “dDocs: Citizen centric document technology,” white paper, v1.1c, June 2017.

[5] A. Ribeiro, A. Silva, and A. R. da Silva, “Data modeling and data analytics: a survey from a big data perspective,” *Journal of Software Engineering and Applications*, vol. 8, no. 12, p. 617, 2015.

[6] H. Ye, X. Cheng, M. Yuan, L. Xu, J. Gao, and C. Cheng, “A survey of security and privacy in big data,” in *Communications and Information Technologies (ISCIT), 2016 16th International Symposium on*. IEEE, 2016, pp. 268–272.

[7] Q. Xu, L. Yang, and C. Yu, “Research on influence of big data to chinese e-govemment,” in *Reliability Systems Engineering (ICRSE), 2015 First International Conference on*. IEEE, 2015, pp. 1–8.

[8] M. Pannu, B. Gill, W. Tebb, and K. Yang, “The impact of big data on government processes,” in *Information Technology, Electronics and Mobile Communication Conference (IEMCON), 2016 IEEE 7th Annual*. IEEE, 2016, pp. 1–5.

[9] T. A. Majchrzak, T. Jansen, and H. Kuchen, “Efficiency evaluation of open source etl tools,” in *Proceedings of the 2011 ACM Symposium on Applied Computing*. ACM, 2011, pp. 287–294.

[10] O. Belo, H. Correia, P. Rodrigues, and R. Barros, “A personalization system for data visualization platforms,” in *Innovative Computing Technology (INTECH), 2016 Sixth International Conference on*. IEEE, 2016, pp. 162–167.

[11] M. O. Ward, G. Grinstein, and D. Keim, *Interactive data visualization: foundations, techniques, and applications*. CRC Press, 2010.

[12] Tamara Munzner, *Visualization Design and Analysis: Abstractions, Principles, and Methods*. AK Peters, 2014.

[13] S. Few, *Information Dashboard Design: The Effective Visual Communication of Data*, 1st ed. O'Reilly Media, 2006.

[14] A. Vaisman and E. Zimányi, *Data Warehouse Systems: Design and Implementation*, 2014th ed. Springer, 2014.

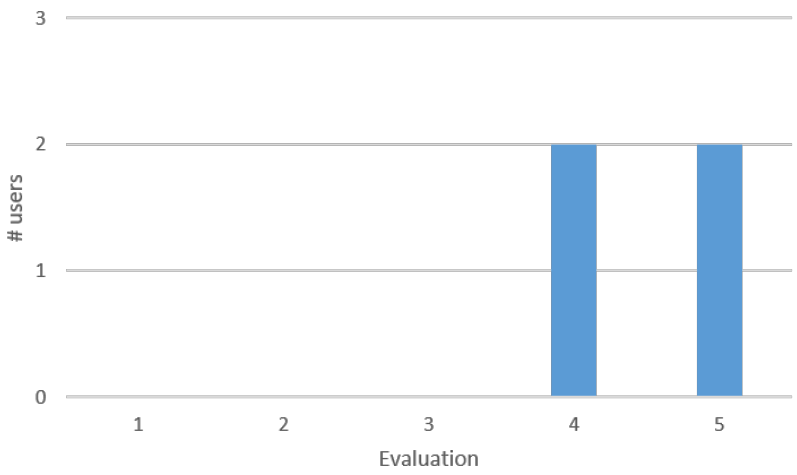
[15] J. Rubin and D. Chisnell, *Handbook of usability testing: how to plan, design, and conduct effective tests*. John Wiley & Sons, 2008.

A

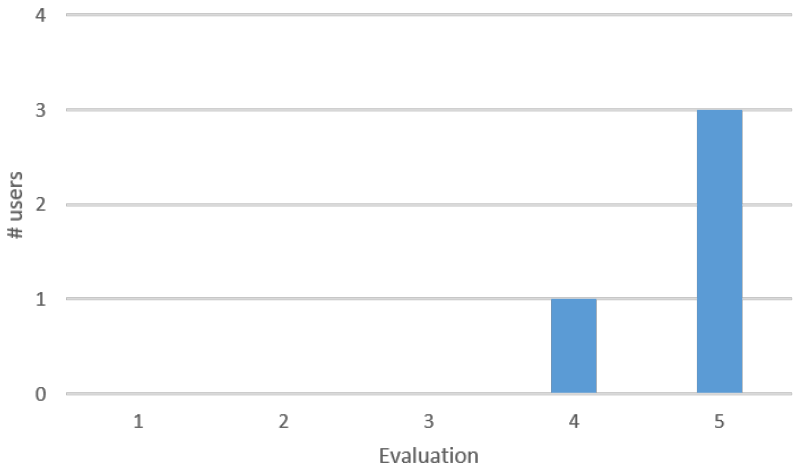
User Session Questionnaire Results

Number of participants: 4.

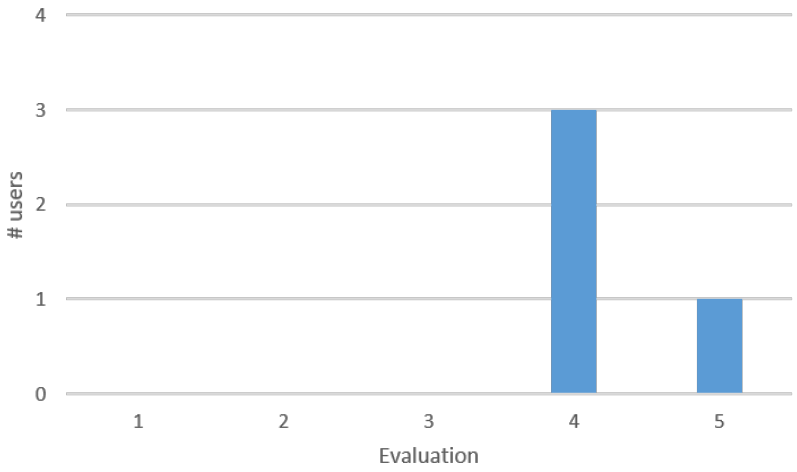
How do you evaluate the dashboards? (1-Awful, 5-Excellent)



How easy is it to understand the information presented in the dashboards? (1-Extremely difficult, 5-Extremely easy)



How consistent are the dashboards? (1-Very inconsistent, 5-Very consistent)



How clean are the dashboards? (1-Very unclean, 5-Very clean)