A case using the DB4G framework and public cloud for an information system

Jorge Guilherme Rodrigues de Almeida
Instituto Superior Técnico
jorgegralmeida@tecnico.ulisboa.pt

ABSTRACT

Information Systems are nowadays a key element for any organization who would like to be competitive and efficient. Being able to share information in an agile and safety way is a must and it’s a requirement that these systems must answer to. A system available in one single device, locally accessible and without the options to access data in real time might be seen as lacking. Cloud has been expanding in the latest years and provides viable solutions for such problems. Thus, we will be making a case study using a framework called DB4G, which uses the Cloud’s potential to store and deliver key-users their information. Making use of this tools, we will model and implement an information system solution to a nonprofit organization, making use of their current business processes.

Keywords: Cloud, Google, API, DB4G, Spreadsheet

1. INTRODUCTION

In the last decade, the term Cloud has been gaining popularity. Cloud computing is defined as a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction [1].

The term cloud computing can be referred to applications delivered as services over the internet as well as the hardware and software in data centers that provide those services. The data center hardware and software is named "cloud" while the service sold is designated as "utility computing". Cloud computing is a join of these two concepts and can be considered public, private, or hybrid [2].

In public cloud, services are available to the general public and provide great levels of efficiency and scalability as the resources are shared. In other hand, control over data, network and security is absent. Private cloud are services and infrastructure maintained on a private network, only available internally to an organization. This solution increases control over data but might increasing costs as it requires the organization to purchase and maintain hardware and software.

Cloud computing has made possible fully serverless models of computing where logic can be amplified on-demand, taking away the user’s responsibility of managing infrastructures and reducing costs of maintenance and electricity consumption[3] [4].

There are several known cloud services as Microsoft Azure[5], Amazon Web Services[6], IBM[7], Salesforce[8], SAP[9] or Google Cloud[10] [3] and they all have been growing adoption both by general public as enterprises[6]. The percentages of individuals and enterprises that are running applications in Cloud can be seen in Figure 1.

![Public Cloud Adoption: 2017 vs 2018](https://www.ibm.com/cloud)

Figure 1 Public Cloud Adoption: 2017 vs 2018 [6].

1.1. PROBLEM ANALYSIS

"Database for Google framework" or, “DB4G” for short is a PHP framework that uses public cloud as a form of database, more precisely Google Cloud. It was developed by a fellow IST student (Diogo Serafim) and provides a barebone and a set of functions that we can use to setup our system and interact with Google Drive. Our goal is to provide a case study scenario by developing a system using this framework. We want to assess if this tool is a viable option for organizations that meet certain requirements such as requiring a relative low amount of stored data, simple or medium complexity business processes.

Knowing this, it was desirable to work with an organization where we could understand and model their processes with detail. The organization chosen was Associação Portuguesa de Gestão de Projetos (APOGEP). APOGEP is a nonprofit Portuguese association which promotes and certifies professionals in Project Management. They are linked to International Project Management

---

[1] https://trends.google.com/trends/explore?date=all&q=cloud
[8] https://cloud.google.com
Association (IPMA) and, having the national Certification Body (CB), follow their International Certification Rules (ICR). To perform these certification processes and corresponding support operations, APOGEP relies exclusively on Microsoft Excel with several Workbooks and Sheets.

This generates a high risk of duplicated information, as well as a higher risk of mistype or miscopy due to the amount of information transiting between different workbooks or worksheets while it might increase the overall time required to run the whole certification process.

We aim to deliver a system that allows to reduce the time needed to run the process tasks, automating repetitive tasks, creating a reliable database and having a well-defined business process implemented with a known architecture.

2. RELATED WORK

2.1. GOOGLE CLOUD

Google Cloud is the cloud service offered by Google, having a free usage program for less demanding users and price plans for the ones who need to access data more often or store bigger amounts. Google Cloud provides APIs in various areas that allows interacting with data on Google Cloud or Google Drive over the internet with dynamic hardware allocation based on user needs.

G Suite, previously known as Google Apps for Work and Google Apps for Your Domain, is a set of Google Cloud productivity and collaboration applications aimed for business use. Gmail, Google Calendar, Google drive, Google Docs (Word Documents, Spreadsheets, Forms and Presentations) are examples of those applications, all of them accessible through the Google Cloud Platform. G Suite, like Google Cloud itself, is free but feature different price plans for users or companies who intend to obtain added functionalities.

2.2. GOOGLE CLOUD API

An API is a "set of commands, functions, protocols, and objects that programmers can use to create software or interact with an external system"[10]. This makes it easier for a developer to use a certain technology as it allows to call standard commands, does not require writing code from scratch, decreases complexity to understand the system and, consequently, improves development times[10].

Google APIs use a JSON REST interface that can be called directly or through Google client libraries, being accessible from server applications in a variety of programming languages (Java, JavaScript, .NET, Objective-C, PHP and Python, among others), from mobile apps or from Google Cloud Platform Console web[11].

2.3. VALUE ANALYSIS OF GOOGLE CLOUD

Google Docs is a free web-based software office suite that allow users to create, share and edit documents through the Google Cloud. Google Docs applications are compatible with Microsoft Office formats, so it gives users the ability to upload Microsoft Word or other text documents, spreadsheets or presentations into the system and convert them to Google’s editing files. From there, multiple users can create or edit the files both online and offline, asynchrony or simultaneous, in a real-time collaboration.

There is also a version control that allow users to keep track of changes, who made those and revert them to a previous state if necessary. Google saves up to thirty days of modifications in documents. Google Docs is fully integrated with Google Drive, and all files created with one of these applications are saved by default into Google Drive[12,13].

Although Google Cloud offers some good perks, there are some negatives as limit on documents size[14] (50MB for Google Docs, 100MB for Google Slides and 5TB on other files) and limit on the number of cells available in Google Sheets (two million). One negative aspect that we found the most impactful is the time needed to communicate with our documents through the API and perform modifications or reads. When compared to a local MySQL, NoSQL, or other type of local database, these access times may be significantly longer due to the need of exchanging data online.

2.4. SECURITY IN GOOGLE CLOUD

Google APIs use OAuth2.0 as authentication and authorization security measure. “OAuth 2.0 authorization framework enables a third-party application to obtain limited access to an HTTP service” [7]. OAuth introduces an authorization layer that separates the client from the resource owner which means it is no longer needed to share credentials for third-party applications to access information. A token is only issued to a third-party client based on approval of the resource owner and can only be used to access resources based on the authorization scope [7]. Google also offers “2-Step Verification”, also known as two factor authentication, where after entering the correct login details, a second code/challenge is required. It can be sent by email, telephone message or presented via a U2F Security Key[15,16].

2.5. MINI FRAMEWORK

MINI[17] is “simple and easy to understand skeleton PHP application”. MINI is a minimalist framework, “easy to install” and “runs nearly everywhere”. Also removes the need of “reading in massive documentations of highly complex professional frameworks”. MINI GitHub page states PHP 5.3.0 or upper, MySQL and apache “mod_rewrite” module activated as requirements for the framework.

2.6. PHP SPREADSHEET CLIENT LIBRARY

In order to interact with Google Spreadsheets, the framework uses Asim Liaquat PHP google spreadsheet client[18] that “provides a simple interface to the Google Spreadsheet API v3”. It was used on this framework due to the needs of higher development speeds since it lets developers skip the setup of google drive API interactions but also with the aim of having a more stable implementation. This PHP library allows to retrieve a spreadsheet or list of spreadsheets, manage worksheets, access or modify cell reads. When compared to a local MySQL, NoSQL, or other type of local database, these access times may be significantly longer due to the need of exchanging data online.

---

9 https://gsuite.google.com/index.html
10 https://techterms.com/definition/api
11 https://cloud.google.com/apis/docs/overview
12 https://www.google.com/intl/pt_pt/docs/about/
14 https://support.google.com/drive/answer/37603?hl=en
15 https://support.google.com/accounts/answer/6103523?hl=en&ref_topic=6103521
17 https://github.com/panique/mini
18 https://github.com/asimlqt/php-google-spreadsheet-client
3. PROBLEM FUNCTIONAL ANALYSIS AND SOLUTION DESIGN

3.1. BUSINESS PROCESS STRUCTURES AND MODELING

In our study case, the organization aims to certify individuals in project, programme and/or portfolio management, having a level system to differentiate competences levels inside each of the three domains of expertise. The organization provides certification following defined processes in Version 4.0 of IPMA International Certification Regulations (IPMA ICR)[8] and evaluates competences based on Version 4.0 of IPMA Individual Competence Baseline (IPMA ICB)[9].

The mission and purpose of each IPMA CB is to “implement and maintain the IPMA Four Level Certification (IPMA 4-L-C). There are three domains in IPMA 4-L-C System being them project, programme and portfolio. In each of those three, project domain has four levels (A, B, C and D), programme and portfolio have two levels (A and B). This creates eight different unique options shown in Table 1.

Table 1 IPMA 4-Level-Certification Systems [8]

<table>
<thead>
<tr>
<th>Level</th>
<th>Domain</th>
<th>Project</th>
<th>Programme</th>
<th>Portfolio</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Project</td>
<td>Programme</td>
<td>Portfolio</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>Certified Project Director</td>
<td>Certified Programme Director</td>
<td>Certified Portfolio Director</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>Certified Senior Project Manager</td>
<td>Certified Senior Programme Manager</td>
<td>Certified Senior Portfolio Manager</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>Certified Project Manager</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>Certified Project Management Associate</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For each level of certification, there are multiple possible paths that each CB may select. Some CBs may have additional requirements or certification, nevertheless those steps need to be approved by IPMA Certification and Validation Management Board (CVMB). Option paths for Level B is illustrated in Figure 2.

3.2. REPRESENTATION OF CERTIFICATION PROCESSES USING BPM NOTATION

Processes are a “set of interrelated activities that transform inputs into outputs and that create value for a customer”[10]. The study and comprehension of the previous information about certification, as well as repeated conversation with the association stakeholders and both ICR and Manual do Candidato (Candidate’s Manual) gave us enough knowledge to represent these processes. This section will detail all the certification processes that were identified.

This representation allows, regardless of the organizational structure, to simplify and understand the actions performed by the users and, are a way to analyze bottlenecks or issues and identify possible optimizations”. To represent these processes we used Business Process Model and Notation (BPMN) 19.

Figure 3 shows the main certification process, which includes all levels. 5 shows level D certification process. Level D is the most common process in the organization. An exam is the only evaluation demanded for the candidate to pass in order to get a certificate. If the candidate fails the first try, repetition can be requested within two months or the certification will be marked as archived. Failing a second time will also lead to fail and archive later. If the candidate

19 http://www.bpmn.org/
successfully passes the exam (sub-process illustrated in Figure 6), this process will end.

![Figure 5 Level D Certification Process](image)

![Figure 6 Exam Process](image)

Level C, demonstrated in Figure 7, is the process that contains the most type of evaluations. The same repeat logic applies, where the candidate can repeat the exam once or it is timed out if doesn’t request in two months. If the candidate passes, will then need to deliver a report (sub-process shown in Figure 8) and, if it passes, will have a simulation and interview which can be repeated if it fails. Finally, in case of positive decision, certificate sub-process begins, and a certificate is emitted.

![Figure 7 Level C Certification Process](image)

Report sub-process (Figure 8) is called in level C, B and A certifications. After assessors have been assigned to a candidate, we expect to receive a report delivery, which is then distributed by the secretariat to all assessors. Candidate’s summary report and report is assessed, and grade is communicated to CB Director which homologates it. If the report is accepted, it is communicated the positive grade, along with next evaluation dates. If the report is not accepted, the candidate has one month to submit a reviewed report that tackles assessor’s rejection reasons.

Certificate sub-process (Figure 9) is the last of every successful certification process. After candidate passes all evaluations and is decision is favorable this sub-process will start. The certificate is then prepared and, in parallel, APOGEP’s website is updated with certificate holder information if visibility was marked as authorized in application documents. When the certificate is ready, candidate is informed and certificate will be delivered to or picked up by candidate.

![Figure 8 Report Process](image)

![Figure 9 Certificate Process](image)

### 3.3. USE CASE MODEL

In our project, we identified twelve use cases. Figure 10 illustrates a Use Case model for simpler comprehension. On the right of the Figure, we have three actors, all of them belong to the association and perform specific tasks while left side represents people external to the organization, in this case, the ones who intend to get certified.

### 3.4. DOMAIN MODEL

Domain model is an object that includes both behavior and data. As business processes can be very complex, where rules and logic can describe multiple behaviors, a domain model interconnects objects that represent something meaningful [11]. With the usage of the domain model we want to represent the relations between the different concepts that were identified, and that will be implemented in the system. Our domain model shows the architecture that will be implemented using spreadsheets.

Although data types are represented in domain model illustrated in Figure 11, google spreadsheets cannot have data types since every cell is treated as a string by DB4G API.

Each person will have to provide their information, including personal contacts and business contact, if they are connected to an organization. The person or the organization connected to it might be a partner of the association.

Each person might apply to several certifications. Each certification corresponds to level A, B, C, D or recertification. Each one has specific evaluations defined in APOGEP’s certification manual. For example, a level B candidate will have to pass a report, an interview and an oral exam, being able to repeat each one if he fails the first try.

Assessors are persons linked to the association that support certification process. They are connected to all evaluations (report, oral exam, interview and workshop) as
they require a lead assessor, co-assessor and, when necessary, an observer assessor.

4. SOLUTION DEVELOPMENT

After searching and learning about related work, deepen our knowledge on the topic of cloud computing, API’s, Google Cloud and DB4G framework, performing requirement analysis, developing a use case and domain model, as well as identifying the data flows and business processes inside the organization, we then proceeded to implement our solution.

4.1. DEVELOPMENT METHOD AND PLANNING

The development method used in this project was evolutionary prototyping.

A prototype is a system, intentionally incomplete, that captures the essential features of a later system [12]. The process itself consists in several iterative cycles, where after the initial prototype is built based on user needs and requirements, the user feedback is requested to further improve the prototype and requirements [13, 14].

Evolutionary prototyping is a sequence of design, implementation and evaluation cycles without any attempt to capture (extensive) requirements in advance, focusing on gaining feedback from early prototypes [15, 16]. It provides flexibility to the software development process, making it gradually progress while adapting itself to shifting organizational contexts and new requirements [13]. It focuses on customer interaction to guarantee that iterations are consistent with expectations [17]. Since implementation of a new system might change the perspective in which the organization operates, new requirements can also emerge in this situation [15].

During the development, and following evolutionary prototyping method, we collected feedback and suggestions from different end-users of the association, both management and collaborators that could use the system on a daily basis.

Figure 13 shows the Work Breakdown Structure (WBS) of our thesis.
4.2. TECHNOLOGY CONSTRAINS

DB4G framework is the technology constrain while developing this system. It allows us to develop our application without the need of implementing direct calls to Google Sheets API. This framework’s use was proposed by thesis supervisors and will be part of a case study using APOGEP’s certification processes. Framework’s architecture is demonstrated in Figure 12.

Everything performed by the user will interact with the Application Server through an HTTP/Ajax communication. Each component depends from the communication with the Google Cloud Platform database it can be seen in the flow of dependencies in Figure 12.

The API that DB4G framework relies the most on this project is Google Sheets API, but also uses OAuth API to authenticate users and provide access to certain website pages. The Spreadsheet API version that the framework uses is v3 and, let users read, write in Google Sheets.

4.2.1. DB4G IDENTIFIED LIMITATIONS

We are using google to grant ownership of a project, so if we transfer those rights to another user, we can no longer transfer the owner permissions again to any other user nor delete the folder from Google Drive. This google limitation is documented20.

All attributes are currently strings (“normal text”) because, contrary to databases, we cannot define a type for each attribute column. Spreadsheet’s API does not allow input verifications nor exceptions throwing if type of data is not correct, so this will have to be controlled and verified internally by the system logic. Nevertheless, we are still exploring a workaround for this problem while he develops the framework.

Last currently identified limitation is that the API does not return an exception if the email inserted does not exist and, as a result, it still attempts to share the folder/spreadsheet with the incorrect email. The only way to know there was an error is that a delivery failure email is sent to the owner inbox.

We identified several internet topics that point to a slow performance of Google Spreadsheet API21,22,23,24,25,26.

The framework has limits in terms of the number of read and write requests it can make (see Table 2). Those limits are imposed by Google to avoid an excessive amount of request and in an attempt to protect their servers. It is possible to control the quota consumption in Google Cloud Platform Console.

Table 2 Google Sheets API – read quota limits

<table>
<thead>
<tr>
<th>Quota</th>
<th>Read Limit</th>
<th>Write Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requests per 100 seconds</td>
<td>500</td>
<td>500</td>
</tr>
<tr>
<td>Requests per 100 seconds per user</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Requests per day</td>
<td>Unlimited</td>
<td>Unlimited</td>
</tr>
</tbody>
</table>

4.2.2. CRITICAL ANALYSIS OF DB4G FRAMEWORK

After exploring the framework and related tools, we can expect some positive and negative aspects when using DB4G framework to develop our solution. We expected positive characteristics such as (1) possibility to import (or mimic) current spreadsheets structure and modify them to meet the new needs; (2) Google Drive offers online forms that can be used directly to integrate and collect information into spreadsheets; (3) API allows greater flexibility and facilitates future changes in the system; (4) Google Cloud is available online on different devices with 99.978% availability and no scheduled downtime; (5) version control allows rollbacks up to thirty days in case of unwanted changes or other complications; (6) possibility to use or create custom macros inside spreadsheets to transmit or verify information; and (7) can be automated with spreadsheet-driven and time-driven functions triggers.

On the other hand, we can assume negative aspects such as (1) the system will still be spreadsheet based; (2) all confidential information is online inside the Google Account which, without additional security like two factor authentication can lead to security problems; (3) the process of sharing spreadsheets rest on sharing links or linking accounts and, the creation of permission groups needs to be replicated in every sheet or folder; and (4) difficult for the system to perform data mining using spreadsheets.

4.3. DEVELOPMENT

We started by creating a fork of DB4G framework from GitHub so we could get all framework version updates during our development phase. We instantiated the structure that contains vendor file’s structure, including Google API services and Asim Liaquat PHP google spreadsheet client.

Following the administrator manual provided by the framework, we created the folder structure in the cloud (Figure 14).

Next, we proceeded to create the respective domain model using spreadsheets. Each class corresponds to one spreadsheet in our system, except the class certification process where we included the attributes in class “Level

---

20 https://support.google.com/drive/answer/2494892?hl=pt-PT
21 https://productforums.google.com/forum/#!topic/docs/ZRsUjAKOpTA
22 https://stackoverflow.com/questions/18332239/google-spreadsheet-batch-update-very-slow
23 https://github.com/burnash/gspread/issues/190
24 https://stackoverflow.com/questions/27166493/google-apps-script-iterating-over-spreadsheets-very-slow
25 https://github.com/asimlqt/php-google-spreadsheet-client/issues/114
26 https://groups.google.com/forum/#!topic/google-visualization-api/Hu8lmt21jlg
Domain” to ease the calls on the API. A schedule spreadsheet was also added as support for the evaluations spreadsheet, where we combined all types of evaluation classes seen on the domain model.

We should refer that, when creating those spreadsheets, neither DB4G framework nor Google Cloud has a method of assigning data types to our header or data. When critically needed, this verification is done within our controllers’ code.

Figure 14 DB4G Main Structure

Figure 15 shows the spreadsheets that were mapped into our Google Cloud using the framework and based on our Domain model.

We can also see one folder name “UserApplicationInfo”, which was created to hold our applicant’s submitted files. This was intended to show that is possible to develop a system where we could also upload and download files using Google Cloud. Each applicant will have one folder where he has all his data. When creating a new application, if the applicant does not have a folder on record we create one new folder for the person and one folder inside that one for the application. A person can have zero or one main folders and each main folder might have one or more applications.

Figure 16 represents our code structure. Inside the controller folder we have created our four main controllers:

- Application Management
- People Management
- CB Director Management
- Billing Management

The other controllers came by default with the framework and allow us to create and manage our Google Cloud structure directly from the web interface. Our system code is split into three folders, the application that contains DB4G and our customs controllers and configurations, the public, which provides an interface to the exterior, that contains the html and calls to our application and vendor folder and, lastly, vendor folder, that contains functions that are called on our controllers. Our custom controllers were created in order to accommodate and match the different Use Cases and all the logic in those controllers are based on the processes we have identified earlier.

In “Application Management” controller we provide all functions that are related to the application itself. Examples of those are register a new application, manage applicant documents, manage eligibility, assign a candidate for assessment, issue certification, among others. We also added what we call “First Contact” management, which is was not inside the initial scope but was requested in early stages of development.

“People Management” controller includes all functions related with registering and management both people and associates. Some examples of those functions are register new person, manage personal information, register new partner, edit partner information, new individual associate and new collective associate and manage associates.

“CB Director Management” controllers, on the other hand, includes everything that is within CB Director scope
such as register and manage assessment schedules, managing assessors, manage complaints and appeals.

It was also requested early if it was possible to include the billing status of each applicant. We included it in the first iteration and created a new spreadsheet to register those details. It is relevant to say that this does not have integration with the organization current invoice software.

Our GitHub repository held the fork code and, furthermore, we also used it as an issue register. We created twenty-four issues with the label “framework”, which we attributed whenever the scope was outside our own controllers. There were also issues reported personally that were not registered in GitHub.

The development using the framework was relatively easy. We had all most of the functions ready and fully operational to use. There were some exceptions where we had to include our own code or extend the framework code ourselves. For example, since we had no way to mimic a join SQL statement in the framework, we had to create a function in the framework that could take two arrays and join them given a common key. Other function added to the framework spreadsheet library was “getRowDataCount” that returns the number of that have been written. This is critical when we need to get the primary key (ID) since we don’t have a global iterator variable that stores the current ID for each spreadsheet.

Another addition was “getAllNonMatchContents”, that performs a search by a key (column) in a worksheet and returns all cells that don’t match a certain string. The DB4G framework did provide a good advantage in terms of data migration, in a way that is relatively easy to grab the current organization data from Excel and map it to our Google spreadsheets. There is a need to create an ID for each person and application in the legacy excel and split that data into our different new worksheets. That data will instantly be available.

5. VALIDATION

5.1. TESTING

5.1.1. USABILITY

To test our system for usability we arranged a meeting with APOGEP secretariat. We followed our defined use cases, trying to simulate the everyday tasks secretariat would be performing in our system. These tasks were performed in the development’s computer to ensure maximum performance and compatibility.

We met in APOGEP’s office and made the testing in a meeting room. This meeting room did not have a cable connection available, so we had to use wi-fi to connect to the internet. For reference, all our system utilization was made via cable connection or with a strong wi-fi connection.

Unfortunately, this was not the case in APOGEP’s meeting room.

Since the beginning of the tests we could perceive a delay that was a bit more than usual. By default, the timeout on the API calls is thirty seconds and, in fact, we got timeouts from simple operations that we have not been facing before in a developments environment. We increased the timeout time on the spot to ensure that we could perform the tasks, even with a high waiting time.

Although the system itself behaved as expected, excepting some bugs that we either fix on the spot or were not critical enough to delay the tasks, the test user said that the current performance might rule out the usage of such system in a work environment.

Table 3 Relation between Controllers and Use Cases

<table>
<thead>
<tr>
<th>Application Management</th>
<th>UC01</th>
<th>UC02</th>
<th>UC03</th>
<th>UC04</th>
<th>UC05</th>
<th>UC06</th>
<th>UC07</th>
<th>UC08</th>
<th>UC09</th>
<th>UC10</th>
<th>UC11</th>
<th>UC12</th>
</tr>
</thead>
<tbody>
<tr>
<td>People Management</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CB Director Management</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Billing Management</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The test user also suggested having a google form that would be an input to manual operations as user registering.

5.1.2. PERFORMANCE

In order to identify the performance bottleneck of the system, we analyzed the time that it took the browser to get the first byte, denominated by “Time to first byte” (TTFB), how much time it took each of the framework calls, how much time it took to render the page. In this section we will perform a statistical and critical analysis on those values.

Time to first byte (TTFB) can be seen as a measurement to indicate the responsiveness of a webserver or network resource27. This timing includes one round trip of latency and the time the server took to prepare the response28.

The following table shows the raw data collected in the several tests we made. Those tests were made using a cable connection instead of wi-fi to ensure we had the least interference as possible in the results. To measure and ensure stability of our internet connection between our computer and our API host, which is google, we pinged google.pt for twenty packets.

We had eighteen packets with a 16ms response and one with 15ms and another with 17ms. We can affirm that our connection was stable to perform these tests.

---

Furthermore, tests were prepared and executed sequentially in the same time frame to safeguard the least disturbance in data values as possible.

Table 4 shows some of the statistical analysis we made with the results. The TTFB and Full Time (includes DOM downloading and DOM processing) values are in milliseconds (ms) and were taken from Chrome DevTools\(^29\).

We have analyzed the time of the following:

- “google.com” page;
- Main page of the framework, which has no framework calls;
- Manage first contact, which is one simple call to retrieve the values from one single spreadsheet;
- Register first contact, which retrieve and posts data to a single spreadsheet;
- Manage Associates, which has a much more complex call as it retrieves data from four different spreadsheets;

<table>
<thead>
<tr>
<th></th>
<th>API POST First Contact</th>
<th>API GET Associates</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TTFB</td>
<td>Full Time</td>
</tr>
<tr>
<td>Average</td>
<td>4178,85</td>
<td>4386,8</td>
</tr>
<tr>
<td>St. Dev</td>
<td>763,82</td>
<td>762,52</td>
</tr>
<tr>
<td>Median</td>
<td>3827,5</td>
<td>4045</td>
</tr>
<tr>
<td>Maximum</td>
<td>6143</td>
<td>6334</td>
</tr>
<tr>
<td>Minimum</td>
<td>3363</td>
<td>3581</td>
</tr>
<tr>
<td>Norm. Dist. 90%</td>
<td>5157,73</td>
<td>5364,01</td>
</tr>
</tbody>
</table>

5.2. CRITICAL ANALYSIS OF TESTING RESULTS

This performance problem was identified since the beginning and we could say results were as expected. Although the framework did improve since the start of this project with some iterations over the main code, Google API might prove itself not to be very effective for this kind of implementations compared to traditional databases options.

It took our most simple page with API calls around 2,5 seconds (2469,35 ms) to load, which means that, the more complex or, the more different spreadsheets and worksheets are needed for a specific use case, this time will multiply, inducing frustration on the end-user. We can this happening in “Manage Associates” (described as “API GET Associates” in the above table) where the average time was close to 17 seconds to retrieve information on 6 different spreadsheets and, statistically we would get a 90% confidence of getting the page ready for the user before the 20,5 seconds mark.

For example, by making a scenario of Use Case 01 (Create Application), assuming the applicant is already registered in the system and excluding data input time, it would take the system almost 19 seconds (18757,2 ms) to process the data.

These values seem indeed a little too much for a system to be used in an organization compared with the easiness and speed of an Excel Spreadsheet and, the value gained by using Google Cloud and having an automated system might not be enough to justify the utilization of such system. These values might also have the tendency to increase as the size of our spreadsheets increase.

There are some mitigations that could nullify some of the effects that those calls have on our performance. First option would be merging all spreadsheets into one and have separated worksheets. Another option would be storing the spreadsheet in our system for read calls and only make API calls when updating values. This second option could create inconsistency between stored values locally and Google Cloud.

---

\(^29\) https://developer.chrome.com/devtools
6. CONCLUSIONS

6.1. CONTRIBUTIONS

The framework provided nearly all functions we needed to implement our system. We added new functions in the spreadsheet model controller as they have a potential to be used outside our project scope. These functions included getting values that would not match a specific string. We also added a function that would retrieve the current number of rows inserted in the framework as there were situations were we needed to get the ID in advance to register data. We also worked together with Diogo Serafim, the developer of the DB4G framework, to fix any problems identified during framework development iterations.

After development we can say that we have fulfilled the defined requirements we have set for this work, as our system answer to all of Use Cases. Furthermore, we focused on creating the system according to current organization processes.

During this work we have stumbled upon a few difficulties. Those varied from having an instable framework in the beginning of this work since it was still under development, poor performance, due to slow API responses. These problems did not impact development itself at a great scale but did impact testing due to degraded performance on the organization wi-fi network.

Nonetheless, we can say that the framework itself provides a good solution for a system using Cloud, but there is a need to empathize implementing a system where spreadsheets perform a similar role to SQL tables is hard and underperforming. We suggest caution implementing in organizations where processes or domain model’ complexity is medium to high, as the system will perform faster as the less API calls are made.

6.2. FUTURE WORK

The framework uses Google Spreadsheet API v3, and a new API (v4) is now available. Recent comments in IT websites state that this new API does not provide an increased performance globally. Yet, some of the possible work to be done in this framework would be updating to the new API. It makes calls easier and the PHP Client Spreadsheet Library the DB4G framework uses would not be needed.

Knowing recent development on European rules about data protection (GDPR), it would be interesting to provide encrypting and decrypting functions in the application, so we could make sure that our data stored in the spreadsheets is safe and encrypted.

Knowing that secretariat has one of the most manual time-consuming tasks in this process, insertion of applicant data, an integration of our system with Google Forms is one improvement that could be seen as beneficial to the end-user. Latest Google Form even allows document update from the applicant computers, which would lead to a much easier and automated system.

Other functionality that could improve system’s usability would be automatic emailing of information in cases as: “Application Approved”, “Payment receive confirmation”, “Certificate Emitted”, among others.

REFERENCES