

Transport application prototype using a progressive framework together with various tools and protocols.

Prototipo de aplicación de transportes utilizando un framework progresivo junto con varias herramientas y protocolos.

SERGIO GONZALEZ FRANCISCO

MÁSTER EN INGENIERÍA INFORMÁTICA, FACULTAD DE INFORMÁTICA,
UNIVERSIDAD COMPLUTENSE DE MADRID



Trabajo de Fin de Máster

Fecha: 06/09/2019

Convocatoria de Septiembre - Calificación: 7.1

Director/Colaborador:
Pedro Pablo Gómez Martín
Wojciech Czart

Resumen

Hoy en día una de las cosas más buscadas por las empresas informáticas es la de disminuir los costes y tiempos de sus proyectos. Para ello, y gracias al avance en la tecnología y en el sector informático, las empresas han recurrido a la comodidad y buen resultado que ofrecen frameworks progresivos, como puede ser Vue, para desarrollar sus aplicaciones. De esta manera, la necesidad de contratar desarrolladores especializados en las distintas plataformas móviles (Android, iOS...), desaparece, y por tanto la implementación de una aplicación queda delegada a un solo código (híbrido) que puede ser desplegado por cualquier plataforma, disminuyendo de esta manera tanto los costes como los tiempos.

Por otro lado, la creatividad e innovación, son otro de los aspectos más demandados. El uso de tecnologías innovadoras y que aporten cosas distintas a los proyectos, hacen que te situes un escalón por encima de la competencia.

El objetivo principal de este proyecto es el de entrar en el mundo de estos frameworks, ver que ventajas y desventajas tienen respecto a los desarrollos tradicionales, que resultado proporcionan etc. De la mano con esto, se utilizarán diferentes herramientas como Firebase (autenticación de usuarios y base de datos en tiempo real), Google Analytics (estadísticas y analíticas), Google Maps (mapas) o Google Spreadsheets (generación de hojas de cálculo en la nube), y diferentes protocolos como OAuth2 (conexión segura, autorización a terceros) o Measurement Protocol (envío de solicitudes HTTP sin procesar a Google Analytics), queriendo de esta manera, aportar cosas innovadoras al proyecto.

Utilizando varios de los microservicios proporcionados por la API que ofrece la EMT de Madrid (empresa de transportes), se ha construido un prototipo de una aplicación de transportes, con el objetivo de plasmar todo el estudio de lo mencionado anteriormente, de una manera mucho más práctica y visual, y demostrar que con todas estas herramientas se puede obtener un resultado satisfactorio, según las opiniones de los usuarios, aproximándose a lo que sería una aplicación real.

Palabras clave

1. Single Page Application
2. Aplicaciones híbridas
3. Tiempo real
4. Servicios en la nube
5. Estadísticas
6. Aplicación de transportes

Abstract

Nowadays one of the things most requested by computer companies is to reduce the costs and times of their projects. For this, and thanks to the advancement in technology and in the computer sector, companies have resorted to the comfort and good results that progressive frameworks offer, such as Vue, to develop their applications. In this way, the need to hire developers specialized in the different mobile platforms (Android, iOS ...), disappears, and therefore, the implementation of an application is delegated to a single code (hybrid) that can be deployed by any platform, thus reducing both costs and times.

On the other hand, creativity and innovation are another of the most demanded aspects. The use of innovative technologies and that they contribute different things to the projects, make you stand a step above the competence.

The main objective of this project is to enter the world of these frameworks, see what advantages and disadvantages they have respect to traditional developments, what results they provide, etc. Hand in hand with this, different tools such as Firebase (real-time user and database authentication), Google Analytics (statistics and analytics), Google Maps (maps) or Google Sheets (cloud spreadsheet generation) are used, and different protocols such as OAuth2 (secure connection, authorization to third parties) or Measurement Protocol (sending unprocessed HTTP requests to GA), wanting to contribute innovative things to the project.

Using several of the micro services provided by the API offered by the EMT of Madrid (transport company), a prototype of a transport application has been built, with the aim of capturing the entire study of the aforementioned, in a much more practical and visual way, demonstrating that with all these tools you can obtain a satisfactory result, according to the opinions of the users, approaching what a real application would be.

Keywords

1. Single Page Application
2. Hybrid applications
3. Real time
4. Cloud services
5. Statistics
6. Transports application

Index of contents

| | |
|--|-----------|
| Chapter 1 - Introduction | 11 |
| 1.1. Introduction | 11 |
| 1.2. Motivation and objectives of the work | 12 |
| 1.3. Document structure | 14 |
| Chapter 2 - State of the art | 16 |
| 2.1. Mobile development options | 16 |
| 2.1.1. Native App Development | 16 |
| 2.1.2. Hybrid App Development | 17 |
| 2.2. Web Development Options | 18 |
| 2.2.1. Multi Page Application (MPA) | 18 |
| 2.2.2. Transition from MPA to SPA | 19 |
| 2.2.3. Single Page Application (SPA) | 20 |
| 2.2.3.1 Web components | 22 |
| 2.3. Frameworks | 23 |
| 2.3.1. Vue | 23 |
| 2.3.2. React | 23 |
| 2.3.3. AngularJS | 23 |
| 2.3.4. Angular | 24 |
| 2.3.5. Ember | 24 |
| 2.3.6. Knockout | 24 |
| 2.3.7. Polymer | 24 |
| 2.3.8. Riot | 24 |
| 2.4. ¿Why Vue and not the others? | 25 |
| 2.4.1. Vue vs React | 25 |
| 2.4.2. Vue vs AngularJS | 26 |
| 2.4.3. Vue vs Angular | 27 |
| 2.4.4. Vue vs Ember | 27 |
| 2.4.5. Vue vs Knockout | 27 |
| 2.4.6. Vue vs Polymer | 28 |
| 2.4.7. Vue vs Riot | 28 |
| 2.5. Backend | 28 |
| 2.5.1. Google Spreadsheet | 29 |
| 2.5.2. Firebase | 31 |
| 2.6. Analytics | 31 |

| | |
|--|-----------|
| 2.6.1. ¿What is a tracker and a web analytic tool? | 31 |
| 2.6.2. Google Analytics | 32 |
| 2.6.3. Clicky | 33 |
| 2.6.4. Mint | 34 |
| 2.6.5. Matomo | 34 |
| 2.6.6. Clicktale | 34 |
| 2.7. Maps | 35 |
| 2.7.1. What is a location map? | 35 |
| 2.7.2. Google Maps | 35 |
| 2.7.3. Navmii | 35 |
| 2.7.4. Here we go | 36 |
| 2.7.5. Bing maps | 36 |
| 2.8. Protocols | 36 |
| 2.8.1. OAuth2 | 36 |
| 2.8.2. Measurement Protocol | 38 |
| 2.9. Smart Cities | 39 |
| 2.9.1. EMT Madrid API | 39 |
| Chapter 3 - System development | 41 |
| 3.1. Application description | 41 |
| 3.2. Methodology and project management | 55 |
| 3.2.1. Methodology | 55 |
| 3.2.2. Project management | 57 |
| 3.3. Use cases | 57 |
| 3.3.1. Introduction | 57 |
| 3.3.2. Actors | 58 |
| 3.3.3. Diagram | 59 |
| 3.3.4. Specification | 60 |
| 3.4. Data and libraries | 64 |
| 3.4.1. Data | 68 |
| 3.4.2. Libraries | 70 |
| Chapter 4 - Tests and results | 74 |
| 4.1. Initial evaluation | 74 |
| 4.2. Practical evaluation with real users | 78 |
| 4.3. Results of the practical evaluation | 79 |
| Chapter 5 - Conclusions and future work | 86 |
| 5.1. Conclusions | 86 |

| | |
|--|------------|
| 5.2. Future Work | 87 |
| Chapter 6 - Introducción | 89 |
| Chapter 7 - Conclusiones y trabajo futuro | 91 |
| 7.1. Conclusiones | 91 |
| 7.2. Trabajo futuro | 92 |
| Bibliography | 94 |
| Appendix A - Set up Vue | 96 |
| Ap.A.1. Building an app locally | 97 |
| Appendix B - Set up Firebase | 101 |
| Ap.B.1. Firebase as user's authentication | 103 |
| Ap.B.2. Firebase as real-time database | 106 |
| Appendix C - Set up Google Spreadsheet + OAuth2 | 108 |
| Ap.C.1. Google Spreadsheet without authentication | 108 |
| Ap.C.2. Google Spreadsheet with OAuth2 | 110 |
| Ap.C.2.1. Create the project and the credentials | 112 |
| Ap.C.2.2. Get authorization tokens | 114 |
| Ap.C.2.3. Enable the APIs | 119 |
| Appendix D - Set up Google Analytics | 120 |
| Ap.D.1. Tracking with Library | 120 |
| Ap.D.2. Tracking with Measurement Protocol | 124 |

Acknowledgments

I would like to thank my family for all the support (both emotional and economic) received during the realization of this master during the last two years.

Also, I would like to thank Wojciech for all the help he has given me being in Poland and all the new tools he has taught me, I am sure they will be very useful for me in the future.

Chapter 1 - Introduction

1.1. Introduction

At the moment, on planet Earth, there are approximately 7650 million people spread across all countries and framed at different ages.

In the past year, there have been more than 3000 million users of smart mobile phones (smartphones), but thanks to various studies, this figure will increase over the next few years, reaching an amount close to 3800 million users, a real barbarity .

It is very rare to see a person on the street who does not have a telephone with internet connection from which consult an innumerable amount of services, so developing software that does not fit for mobile devices, currently, is an important loss of customers.

In addition, the apps are ahead of the websites, in almost all relevant aspects: easier and faster browsing, more immediate access to the content, it adapts better to the needs of the client and remembers its browsing preferences, so develop only a website and not developing an app is a negative impact for the company.

Technology has improved a lot during the last years (it has always done it, and every time it is faster), and therefore, the field of programming has also done so.

Currently there are many ways to program the same thing, using different programming languages, different tools etc. But what is clear, we must carry out a study before starting to develop, to see which tool fits best with the expected result.

A very important aspect is the innovation and creativity, so using innovative technologies to bet on results, other than the results already exist, making a difference with the competition, is a crucial point in the development of a new project.

This is what I want to achieve in this project. Study a series of innovative technologies, which have not been used by so many people, to try to achieve a successful result, embodied in a small software.

The most important technologies that have been used in this project are: Vue (framework), Google tools (Google Analytics, Firebase, Google Maps, Google Spreadsheet) and protocols (Measurement Protocol, OAuth2).

As can be verified in the bibliography, there are many references to these technologies separately, but when all of them come together, there are problems to find references, mostly because Vue is a really new framework.

It has been chosen the EMT Madrid API to work with its data in order to show how these technologies work with real data and to try to show how can they be used approaching a real example of application. The information that this API offers is related to bus lines , the use of public bicycles and parkings, in Madrid.

The application that has been implemented focuses on the information offered on buses and uses it to create some functionalities such as consulting the closest stops to your location, checking how many people have consulted a certain stop or getting which buses are going to arrive at a stop.

The proposed solution should be understood as a starting point to advance in this direction and as proof of concept of how these technologies can work together.

1.2. Motivation and objectives of the work

At present, the need to have a mobile phone is huge, since the world in general is being digitized faster and faster, and what was once on the streets is now on Internet.

Nowadays, most people carry out all the procedures of their daily lives through the internet, from registering with social security to buying food or clothes. That is why, the growth of the development of mobile phones has increased in recent years, since you can take your mobile phone anywhere, without it being a great burden of weight, and from which you can do everything.

Due to this, the idea of knowing how to develop applications for mobile platforms, for a developer, is a highly valued skill by computer companies.

In addition to this, and due to the great advancement of technology, what better than venturing into the world of hybrid applications and being able to develop for several platforms at the same time? This means a great reduction in both costs and time.

On the other hand, and as I have explained in the introduction, being creative and providing new ideas (technologies, tools, etc.) that help when you are implementing a software or make it different, is highly valued.

This is why the idea of introducing relatively new tools, in the development of this software, such as Firebase or Google Spreadsheet.

Finally, the marketing sector it has always been very attractive and it includes many things like advertisements, how the statistics are analyzed, where the business is getting from etc.

Marketing has always been a very important factor in the companies of any sector, a good advertising campaign can carry out a product, so using something related to marketing, is a new challenge to face for everyone, that's where Google Analytics came into play.

As can be deduced by the previous introduction section and the motivation of the project, there are a number of main objectives, and other objectives that are secondary, which have been divided into two sections.

Main objectives

- Use of new technologies with the aim of developing a unique software that can be used by different mobile and web platforms.
- Simplify processes that are usually in all applications, such as login / logout or cloud storage.
- Enter the world of marketing and advertisements, informatically speaking, in order to gain knowledge and develop future marketing campaigns.
- Carry out the development of a software, as a real example of application, including the objectives mentioned above.

- Improve and obtain the knowledge of developing a real project starting from scratch, going through all the stages of a project, from the idea / planning, to the tests / deployment.

Secondary objectives

Apart from the main objectives there are others that can be done to fully satisfy the project:

- Check the viability of the different options that are available within the choice of tools or services to be used.
- Try to remain in the free layer of the servers, to demonstrate that the objectives can be achieved without a large outlay of money.
- Explain how to set up the different tools used and different concepts related to them so any user can use them.

1.3. Document structure

This document is organized in 7 chapters, each dedicated to a theme related to this project.

This section belong to the first chapter of introduction, which also presents the motivations and objectives of the project.

Chapter 2 collects all the study work done before starting the implementation. The chapter is dedicated to face the different options that exist and to explain the tools are used.

Chapter 3 of the development of the system includes one of the most important parts, since there is a global description of the system and the way in which the implementation has been carried out and the corresponding result.

In Chapter 4, you can see a description of the tests performed on real users after the implementation is finished. The results obtained and a discussion on these results are detailed.

Chapter 5 deals with future improvements that can be made in the project, new ideas, suggestions from users etc. Also, it sets out the conclusions reached after this project was carried out.

Chapter 6 includes the introduction translated to Spanish.

Chapter 7 includes the conclusions and the future work translated to Spanish.

Finally, there is a small bibliography where you can check different sources from which part of the information, has been taken, and also a series of appendices, related to configurations of the tools used.

Chapter 2 - State of the art

The first question that comes to our mind is, What is the perfect programming language or environment to develop our app? To know what could be the best option, first we are going to analyze and explain some essential concepts about it.

2.1. Mobile development options

The strong growth of the smartphone market in recent years has led to the share of the mobile phone traffic to 52.2% in the year 2018. The availability of rentable smartphones has led to an increase in our dependency on them thus making it difficult for the developers to choose between Native app development and Hybrid app development ^[1].

2.1.1. Native App Development

A native mobile app is an application created to work in a single particular platform or device, like iOS (Swift or Objective-C), Android (Java, Kotlin) or Windows Phone (C#).

In native mobile app development, we have to implement coding for different platforms using different programming languages specific to each operating system. In result, we have different versions of a single application made for each platform. Being specific to that operating system, it runs smoothly over it.

These apps will achieve higher marks in speed and performance tests considering that they have been optimized for an specific platform. The feel and experience are much better with better scrolling, profound effects and animations ^[2].

One important advantage of these apps, is the data protection and security, because full efficiency of hardware resources are required and this is only possible on them. Also, native apps offer fast access to inbuilt device utilities like GPS, camera etc...

It is pretty clear, what are the disadvantages of these apps: development time and development cost. Implementing the design for every device dimensions like numerous varieties

of Android, iPhones etc... takes longer to complete the app. In building a native app, you will need as many development teams as the platforms you want the app to be created on, that implies a lot of cost.

2.1.2. Hybrid App Development

A hybrid app is created as a single app to use on multiple platforms like Android, iPhone or Windows Phone. They look like a native application but are actually run by company website.

They are created with HTML, CSS and JavaScript, and they are basically a web-based program put in a native app shell and connected to the device hardware ^[2].

As we already know, hybrid apps are web apps incorporated in a native shell, so its content can be updated as many times as it is needed or wanted, so hybrid apps have a low maintenance. Also, you don't need to hire different programmers depending on the platform, so the cost of development decreased.

Moreover, if you want that your app go live quickly, maybe because there is a high competition in the market, and you want to introduce your app to the target audience as quickly as possible, then an hybrid app has to be your choice.

Between the disadvantages of these apps, we can find that it could be a loss of performance, because hybrid apps add an extra layer between the source code and the target mobile platform, but it varies from application to application. This extra layer makes debugging a bigger affair because developers have to rely on the framework itself to play with the targeted operating system without introducing new bugs.

As compared to native app development, it is difficult to maintain a proper user experience between platforms. If you focus more on one platform, the user experience would worsen for the other platforms

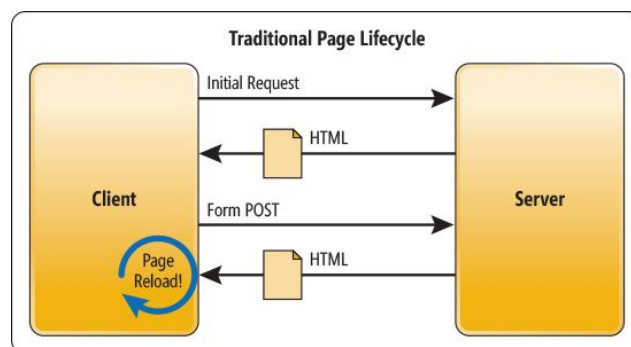
2.2. Web Development Options

Since I have decided that my app will be hybrid, the next topic is deal with the type of the web application.

The inception of web introduced hypertext documents, but those were replaced by the sophisticated web apps. The current web application development scene can be divided into two approaches. They are: Multi Page Application and Single Page Application.

2.2.1. Multi Page Application (MPA)

Multi page applications are the traditional web applications that reload the entire page and displays the new one when a user interacts with the web app. Each time when a data is exchanged, a new page is requested from the server to display in the web browser. This process takes time to generate the pages on the server, sending it to a client and displaying in the browser, which probably it will affect the user experience [3].

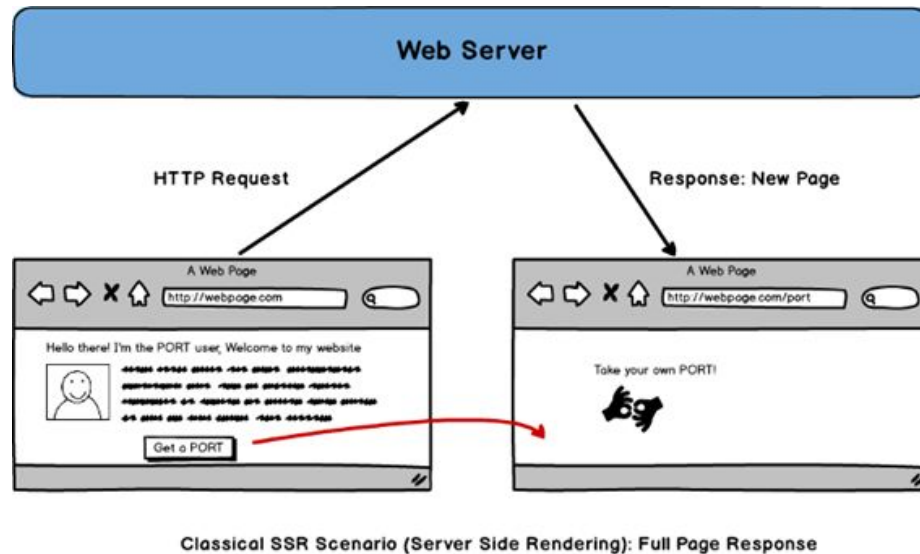


MPA normal flow

These websites perform well on the search engine and provide a visual map of the web app to the user, however, in the other hand, the development is complex and the backend and frontend are coupled.

2.2.2. Transition from MPA to SPA

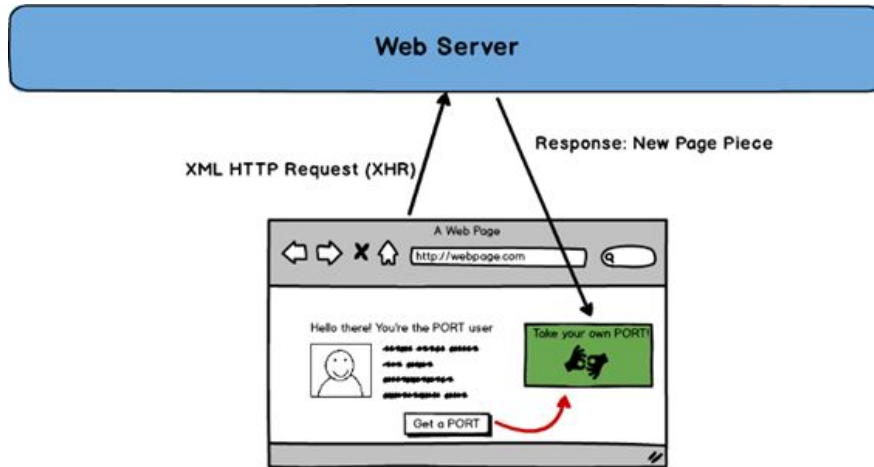
As we can see in the graphic below, a client-server relationship tended to be like the following conceptual scheme: rendering on the server side, that is, the ability to build the labeling of pages and views with data that will be booted to the browser and this just will paint them [19].



First step Transition MPA - SPA

Later, we learnt to alleviate that burden through AJAX, making partial AJAX requests through Javascript that were used to update specific sections, repaint blocks with updated data, modify views etc.

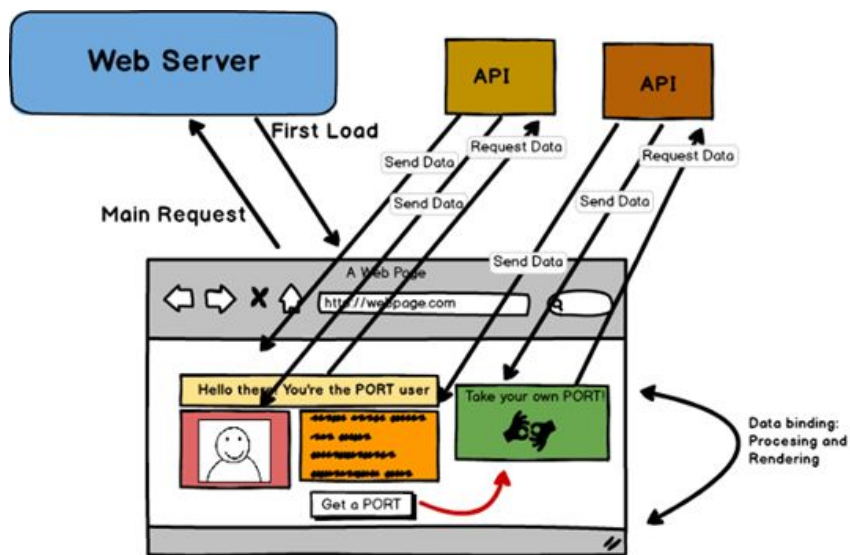
Nowadays, there is a different type of MPA, that it approaches more to the SPAs. These websites tend to depend on the AJAX's requests that try to make the content more dynamic and that speeds up the load and the user experience creating a hybrid model, but this makes the development process more difficult and complex.



Typical Partial Rendering (AJAX): XMLHttpRequest - Response

2.2.3. Single Page Application (SPA)

Finally, we arrive at the SPA paradigm: MVC decoupling where the views are partial blocks that obtain or enter data into the model and are updated individually.



Single Page Application - SPA Rendering Model (Idealized)

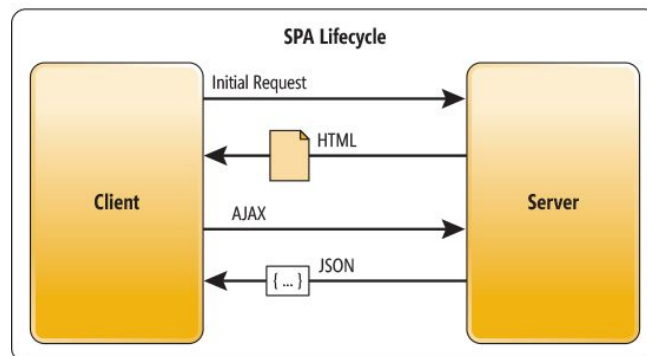
Second step Transition MPA - SPA

Implementing a SPA means that you have to find a way to bring the classic UI programming to the web. When an MPA is made, the HTML is sent directly, with the appearance of the page. A SPA has different visual elements that you want to manipulate based on data that you get from the server, you have to make elements appear and disappear, and that is not easy in a “classic” programming model of the DOM behind the HTML, here is where the Web Components play.

It is necessary to have like a MVC (Model-View-Controller) type, where the view is the HTML that you don't want to manipulate manually. In addition, the part of the model has to be synchronized with the data in the backend.

SPAs present the content in an easy, elegant and effective way as it loads all the content on just one single page rather than navigating the user to different pages ^[3].

SPAs execute the logic in the web browser itself rather than on the server, and after the initial page load, only data is sent back and forth instead of the entire HTML that reduces the bandwidth.



SPA normal flow

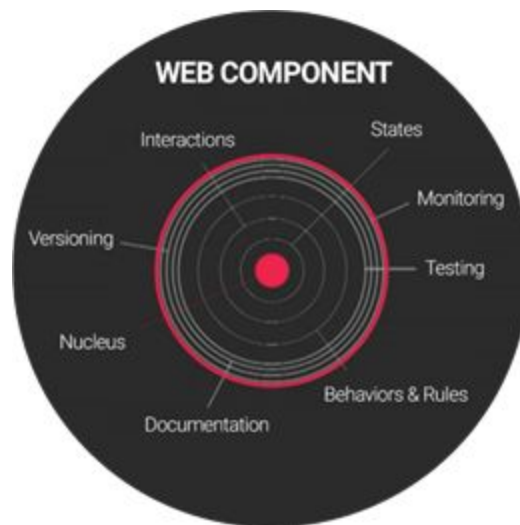
SPAs are smooth and fast, they are easy to develop, deploy and debug, and can be transited to mobile apps by reusing the same backend code. However, SPAs perform poor in the search engine but they can be optimized for it, they just provide single sharing link and are less secure compared to traditional multi-page apps.

Web apps these days, whether it be SPAs or MPAs are highly sophisticated, feature-rich and they provide amazing user experience. Though SPAs approach can be used to build any kind of web app, there are some things that we have to consider before we choose one.

Implementing a SPA and integrating it into a hybrid application generates a much more natural user experience, closer to a typical ‘desktop’ application, than if it is implementing with an MPA model. In fact, a hybrid SPA application allows ‘offline use’ (if the render is not done in the server), something you could not do with an MPA.

2.2.3.1 Web components

Web components are a suite of different technologies allowing you to create small, reusable, modular containers that fit into web apps like puzzle pieces. This means that we can think of anything we have ever created, turn it into an atom and re-use it globally.



Web Component structure

2.3. Frameworks

In this section, we analyze what framework we are going to choose and why. For it, we are going to analyze each of the possibilities ^[5]:

2.3.1. Vue

Vue is a progressive framework to build user interfaces. That means that the main functionalities (rendering and component system) are in a small library, however it is possible to add all the functionalities that other frameworks have like client-side routing or status management, adding additional libraries.

Unlike other monolithic frameworks, Vue is designed from the ground up to be incrementally adoptable. The core library is focused on the view layer only, and is easy to pick up and integrate with other libraries or existing projects ^[6].

2.3.2. React

React is a JavaScript framework maintained by Facebook and a community of developers and companies for building user interfaces. It is optimal for fetching quickly changing data that needs to be recorded. React applications usually require the use of additional libraries to do some tasks like routing or interaction with an API.

2.3.3. AngularJS

AngularJS is an open-source JavaScript framework, maintained by Google, that is used to create and maintain SPAs. Its goal is to increase browser-based applications with Model View Controller (MVC) capability, in an effort to make development and testing easier.

2.3.4. Angular

After releasing AngularJS, the Angular team released Angular which is a complete rewrite of its original AngularJS. Angular is built around the concept of the component and it was rewritten from scratch using Typescript. It offers better performance to web developers.

2.3.5. Ember

Ember is an open-source Javascript web framework, based on the model-view-viewmodel (MVVM). Adding common idioms and best practices into the framework, it allows to create scalable SPAs to developers.

2.3.6. Knockout

Knockout is an autonomous JavaScript implementation of the model-view-viewmodel pattern with templates. The data and the view components are clearly separated and there is a defined layer of code that manage the relationships between the view components.

2.3.7. Polymer

Polymer is an open-source JavaScript library developed by Google developers and users on Github for building web applications using Web Components (set of features that provide a standard component model for the Web).

2.3.8. Riot

Riot is an open-source JavaScript library whose objective is trying to decrease the volume of the frontend space because it is so crowded suppressing,for instance, extra HTML elements.

2.4. ¿Why Vue and not the others?

Why not Angular, React, Ember, or any other good frameworks out there? Whilst React and Angular are backed by big companies, Vue is mostly backed by a very active open source community.

In this section, we are going to compare all the frameworks above mentioned, 1 by 1 with Vue ^[20].

2.4.1. Vue vs React

Probably this comparison would be the hardest, because both have many similarities like they provide reactive and composable view components, they utilize a virtual DOM or they maintain focus in the core library (routing done by a library, for example).

- In React, when the state of a component changes, it triggers the re-render of the entire component sub-tree (starting at that component as root), meanwhile in Vue, a component's dependencies are automatically tracked during its render, so the system knows precisely which components need to re-render when state changes ⁸.
- In React, everything is just JavaScript. Not only are HTML structures expressed via JSX (a declarative XML syntax that works within JavaScript), the recent trends also tend to put CSS management inside JavaScript as well. This approach has some benefits, but also various tradeoffs. In the other hand, Vue embraces classic web technologies and builds on top of them. Also, in React all components express their UI within render functions using JSX, meantime in Vue, it has render functions and even support JSX, because sometimes you do need that power. Furthermore, scoping CSS in React is often done via CSS-in-JS solutions (a new component-oriented styling is introduced), however in Vue, the default and familiar method of styling (style tags) is available, and also, supports many of the popular CSS-in-JS libraries.
- For large applications, both offer robust routing solutions, but it is said that Vue offers a superior development experience. Another important difference is that Vue's routing and state management libraries are kept up-to-date with the core library while React chooses to leave these concerns to the community.
- Vue offers a CLI project generator that makes it easy to start a new project. React is also making progress in this area with create-react-app but it currently has a few limitations. Before you can really get started with React, you need to know many concepts about JSX, build systems etc... while with Vue to get started, all you have to do is drop a single script tag into the page.

2.4.2. Vue vs AngularJS

Some of Vue's syntax look very similar to AngularJS (e.g. v-if and ng-if). This is because there were many things that AngularJS got right and it was an inspiration for Vue in its development ^[7].

- Vue is much simpler than AngularJS, both in terms of design and API.
- AngularJS says how your applications should be structured, while Vue is a more flexible solution and it can be adaptable to a wide variety of projects. Also, thanks to VueCLI you can focus on writing your app instead of spending hours with configurations with AngularJS.
- Vue enforces a one-way data flow between components while AngularJS uses two-way binding between scopes. This makes the flow of data easier to reason.
- Vue separates directives (encapsule DOM manipulations) and components (units that have their own view and data logic). However, in AngularJS, directives do everything and components are just a kind of directive.
- AngularJS becomes slow when there are lot of watchers because everytime anything in the scope changes, all these watchers have to be revised. In the other hand, Vue has better performance and is much easier to optimize because it doesn't use dirty checking (it uses a transparent dependency-tracking observation system).¹

2.4.3. Vue vs Angular

Angular is the next version of AngularJS, but it seems like a completely different framework, that's why the comparison is totally different.

- Angular essentially requires using TypeScript. In many small use cases, introducing a type system may result in more overhead than productivity gain. In those cases it is better to off going with Vue. Vue also can offers official typing and official decorator for those who wish to use TypeScript.
- Both frameworks are really fast, with very similar metrics on benchmarks.

- A full-featured Vue project (30KB gzipped) is significantly lighter than a one generated by angular-cli (65KB gzipped)
- Vue is much less opinionated than Angular, offering official support for a variety of build systems, with no restrictions on how you structure your application.
- Angular's learning curve is much harder because the framework is huge and you need to get familiarized yourself with a lot more concepts. However, to get started with Vue, you just need to have basic skills with HTML and JavaScript.

2.4.4. Vue vs Ember

- Ember's learning curve is high and flexibility suffers while Vue's learning curve is quite easy.
- In Ember, you need to wrap everything in Ember objects and manually declare dependencies for computed properties, meanwhile in Vue, it offers automatic computed properties.
- Vue outperforms Ember by a fair margin. It automatically batches updates, while in Ember you need to manually manage run loops in critical situations.

2.4.5. Vue vs Knockout

Knockout was a pioneer in the MVVM and dependency tracking spaces and its browser support is very impressive.

- Knockout development has slowed and there are many things, like the component system, that they have become outdated.
- Vue is considered much less complex and better structured than Knockout.

2.4.6. Vue vs Polymer

Polymer is another Google-sponsored project and it was a source of inspiration for Vue as well.

- The biggest difference is that Polymer is built upon the latest Web Components features and requires non-trivial polyfills to work in browsers that don't support those features natively. In the other hand, Vue works without any dependencies or polyfills.
- Polymer has its data-binding system very limited in order to compensate it for the performance comparing it with Vue.

2.4.7. Vue vs Riot

Riot provides a similar component-based development model and shares a lot in design with Vue.

- Vue has better performance than Riot because it traverses a DOM tree rather than using a virtual DOM.
- Riot tooling support is much worse than the one in Vue. Riot relies on community support for build system integration.

2.5. Backend

The backend is the web development's part that makes sure that all the logic of a web page works, the set of actions that happen in a web but that we do not see.

It is true that the backend can not live without the frontend and vice versa. However, the back part is especially necessary and essential to develop custom projects.

There are some milestone in the history that I think they have transformed our way of seeing and working with the backend:

- Appearance of mobile applications: the big use of smartphones have brought with it that data could be accessed in an uncoupled way and in conditions of connection not always favorable.
- The cloud: it was already there, but its proliferation has made it extremely cheap to store data on the Internet.

- Functional programming languages: if the cloud saves money, functional programming saves lines of code. It is very present in backend technologies and allows us to focus on the important thing.

It is clear, that many apps need a database to work properly and to save tons of data. At present, there is a very wide range of databases systems: RDBMS, NoSQL, SQL etc. that can work as databases in-memory and as databases as a service.

Selecting the database is very important, especially when we think about the service that will be offered to users and the performance we need. In addition, we should not think only when we install it but also in the future, as the volume grows the problem will grow and this can be translated into slow queries or processes with a very high consumption of resources.

Being different from the others it is a good opportunity nowadays. That's why I wanted to explore new options related to databases. Exploring this database world, I discovered two quite new options and really different from the others, that we can use as database: Google Spreadsheet and Firebase.

2.5.1. Google Spreadsheet

Google Spreadsheet it is a web service of spreadsheets, made in AJAX technologies. Google spreadsheets are wickedly customizable, hosted in the cloud for easy access by anyone and can be designed to solve almost any problem ^[11]. Nowadays it is used by people for many things like:

- Scheduling
- Gantt Charts
- Approvals
- Data validation
- Planner
- Many more...

Also, many people use it as a database. This is not well-known by people, I mean, when someone wants to choose one database for his/her project, they do not usually think about Google Spreadsheet.

After use it as a database, I have got some conclusion:

- It is hard to set it up. You need to set up many things and go through many permission screens to get the API's acceptance.
- It is not so much secure. If you really want to have your spreadsheet full secure, you have to spend so much time to do it, if not, your spreadsheet will be public and everyone can see it.
- The libraries to use it are not very plain and the documentation is quite short, it is more like you have to be a self-taught person. Also, you need to know the format of your spreadsheet before you edit it.
- In the other hand, I find it quite easy for people that do not know anything about queries (using SQL or noSQL). You don't need that knowledge to have your database on Google Spreadsheet.

Having said this, I do not find Google Spreadsheet very suitable to fulfill the database function. I find Google Spreadsheet more suitable to show results or statistics that you send there, and then you can edit them, as you want, in its console/web.

2.5.2. Firebase

Firebase is a mobile and web application development platform that was acquired by Google in 2014. It has 18 products, which are used by 1.5 million apps.

Firebase is a set of tools aimed at the creation of high quality applications, the growth of users and to earn more money^[10]. In other words, is a suite of different services that will make it easier for us to develop our application.

Some of the services that Firebase offers are the following described:

- Authentication: it is a service that simplifies us the beginning of session (log in) and the management of it in our application. It supports social login providers of Facebook, Github, Google..
- Realtime Database: it provides a realtime database and backend as a service. The data is stored in JSON format and rules can be added to allow requests with token or only from a URL, for example.
- Hosting: is a static and dynamic web hosting service that supports hosting static files such as CSS, HTML, JavaScript and other files. It should also be noted that deploying your application will be part of the Firebase CDN and will replicate to servers throughout the world, reducing the time of transfer from these to your visitor's computer.
- Storage: this service is very useful for applications that require saving user files. The developer can use it to store images, audio, video or other user-generated content.
- Many more like test lab for Android and iOS, performance, crashlytics, notifications, cloud functions etc...^[21]

2.6. Analytics

2.6.1. ¿What is a tracker and a web analytic tool?

Trackers are objects that can collect and store data and then send that data to a platform like Google Analytics.

Web analytics is the measurement, collection, analysis and reporting of web data. It provides information about the number of page views and information about the number of visitors to a website ^[12].

The result of your online businesses success is always due to combination of many ideas and efforts, and the successful ‘coming together’ of multiple marketing strategies and different traffic sources. It is really important for a business to keep track its site and its traffic.

Without a good analytics system, your web business is flying blind.

2.6.2. Google Analytics

Google Analytics is a web analytics service offered by Google that tracks and reports web traffic. Google Analytics uses a small piece of JavaScript tracking code to collect data about your website visitors and their interactions on your website ^[21].

The tracking code sends information (hit) to Google Analytics about the users and their interactions on the website. Interactions are all the types of actions users perform on your website.

Google Analytics groups user’s activity into a period of time called a “session”.

- A session begins when a user navigates to a page that includes the GA tracking code.
- A session ends after 30 minutes of inactivity.
- If the user returns to a page after a session ends, a new session will begin ^[13].

A hit is an URL string with parameters of useful information and there are 3 common types of hits:

- Page view hit: it is captured when a user loads a page.
- Event hit: it is sent when a user interacts with a particular element of one page.
- Transaction hit: it is captured when a user buy a product, for example.

Google Analytics is called a ‘freemium’ service, meaning that businesses can use the service without paying a monthly charge, but if they need advanced features, there is a fee.

There are so many things you can find out with Google Analytics, for instance:

- Explore your most important analytics data: Google Analytics offers us customizable dashboards to have all the data that is the most important for us, on the main screen.
- Determine where your best visitors are located: knowing where your best visitors are located, you can consider using advertising via social networks or other services.
- Visualize what people click on the most: if you have a particular link you want visitors to see, you should be sure to place it in the areas of your website that receive the most clicks.
- Find out which campaigns work better: checking what campaigns have had the most traffic, it is useful to make new campaigns in the future.
- Identify your best and worst content: it is possible to check what pages are the top exist pages and, in what pages, users spend the longest time.
- Determine where people leave the shopping cart: by using this setup, you will be able to view reports showing you when people abandon their shopping cart during their purchasing process
- Discover if it is necessary a mobile app for your project: there is an option that allows you to check the percentage of the visits that correspond to each device (mobile, computer, tablet...)

2.6.3. Clicky

Clicky is a useful service that offer a variety of plans for tracking real-time metrics on your website. It is often the first tool that users migrate over to from Google Analytics.

The key difference is in the delay of Google Analytics. Clicky is in real-time, which can be essential if you get spike in visitors from some outside source.

Another interesting feature is that Clicky automatically tracks certain RSS events or campaigns. Also, it can track some Twitter's data and add an individual user tracking (specifying the user name or the email address).

One of the disadvantages of this tool, it is that the free service is quite small, just 3.000 daily page views.

2.6.4. Mint

Mint is a self/hosted web site analytics program. Its interface is very simple and you can get the visits, referrers, popular pages and searches from your app.

It comes with a really fresh layout, with clean and easy to understand charts that display all the information you need to know.

Mint charges an upfront fee of \$30 per site, but with no recurring costs afterwards.

2.6.5. Matomo

Matomo is a free and open source web analytics application (previously known as Piwik). It provides a similar dataset as Google Analytics.

It is a PHP/MySQL software program that you download and install on your own server. You can access to your analytics reports in real time.

The problem of Matomo is that you have to have your own server if you do not want to pay anything, so you would have to spend quite a lot of time setting up the tool.

2.6.6. Clicktale

Clicktale is a cloud-based analytic system and service that allows you to visualize your customer's experience on your website from their perspective. With this tool, it is possible to see what your viewers did while they were on your site. You can literally view exactly what interactions they have had with your site.

For example, it is possible to check if people are putting their mouse on images that belong to the website, but they are not interacting with them.

There are other examples like Church Analytics, KISSmetrics, Open Web Analytics, CrazyEgg, CloudFlare etc... that are similar but they focus their effort in different things.

2.7. Maps

2.7.1. What is a location map?

A location map is typically a simple map to show the location of a particular geographic area. Used on their own, location maps do not differ significantly from traditional maps, differing primarily in the fact that solitary locator maps focus the attention on a single location within the map frame, where traditional maps generally seek to portray a multitude of features across the entire frame.

2.7.2. Google Maps

Google Maps has become an almost obligatory tool for those who search from an address, to more specific details of routes, traffic, public transport, business and the appearance of the streets.

It provides the layout of roads, the locations of cities, state boundaries, geographical features, restaurant reviews, satellite images (Street View perspective) and multiple

transportation modes ^[14]. Also you can share Maps with family and friends and embed it on web pages and emails.

One of the disadvantages of Google Maps, is that Google Maps does not have up-to-the-minute information on unusual conditions, such as roads damaged by the weather or streets blocked by recent construction work.

2.7.3. Navmii

One of the best alternatives that exist is undoubtedly Navmii, especially because of the large community that makes you feed information at a good place.

Navmii is a map and navigation services with many features and it provides services in more than 80 countries. It offers all the basic functions such as navigation, bookmarks, nearest locations, satellite views etc.

It offers a good service to use without Internet, like offline navigation, speed limits, traffic alerts etc.

2.7.4. Here we go

This service has one of the best navigation systems, which can also be used without consuming data. It offers routes and public transport. One of the best functions is the possibility of consulting information while drawing a route, having access to police reports, seed sensors and other data updated in real time.

2.7.5. Bing maps

This service belongs to Microsoft and it is an ideal option for those who are looking for something else in terms of visual appearance. The best features we found are 3D views, real-time traffic, site information and street-level views.

There are other options like OpenStreetMaps, CityMapper, Maps.me, Waze, OsmAnd etc.

2.8. Protocols

In this section, we are going to explain two of the protocols we are going to use: OAuth2 (Google Spreadsheets) and Measurement Protocol (Google Analytics).

2.8.1. OAuth2

OAuth is an open standard for access delegation, commonly used as a way for Internet users to grant websites or applications access to their information on other websites but without giving them the passwords.

In this case, it is needed OAuth2 because provides specific authorization flows for web applications, desktop applications, mobile phones and smart devices.

This protocol is going to be used to established the connexion to Google Spreadsheet in a safer way.

There are 5 important concepts inside this protocol:

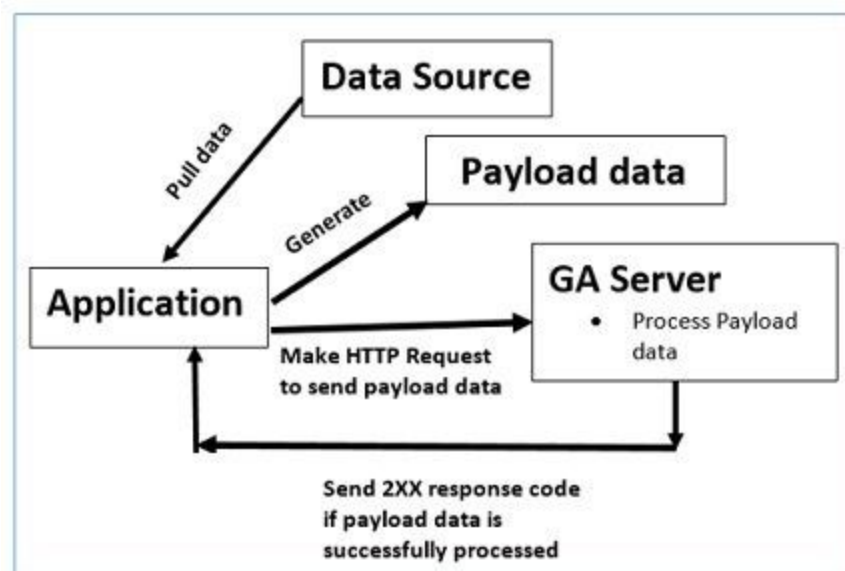
- Client Id . The client must have a unique id and is called Client Id
- Client Secret . It is used to prevent situations in which a fraudulent request is sent.
- Authorization Code .This is a short lived token that is used to obtain access token and refresh token.
- Access Token . This is the token that is used while obtaining the data from a restricted resource. This has medium lifetime, it may expire in an hour's time.
- Refresh Token . This is the long-lived token that is also obtained in exchange for a valid Authorization Code. This is used to get a new Access Token when the current one expires.
- Authorization Code. This is used to get a new Access Token when the current one expires.

It is useful to choose OAuth2 due to ^[15]:

- Two-legged authentication. Sometimes, though, a server just needs to control access to its own content. Two-legged authentication allows the client to authenticate the user directly with the server.
- Simplicity. OAuth 2 simplifies all the communications between the client, server, and content provider. The refresh tokens only exist to provide security when content is being accessed unencrypted.

2.8.2. Measurement Protocol

The Google Analytics Measurement Protocol is a standard set of rules for collecting and sending hits from any Internet-connected device to Analytics. It's particularly useful when you want to send data to Analytics from a point of sale system or anything that is not a website or mobile app, because while the Analytics JavaScript and mobile SDKs automatically build hits to send data to Analytics from websites and mobile apps, you must manually build data collection hits for other kinds of devices. The Measurement Protocol defines how to construct the hits and how to send them to Analytics ^[22].



Measurement Protocol normal flow

The Google Analytics Measurement Protocol allows developers to make HTTP requests to send raw user interaction data directly to Google Analytics servers. This, allows developers to measure how users interact with their business from almost any environment. Developers can then use the Measurement Protocol to:

- Measure user activity in new environments.
- Tie online to offline behavior.
- Send data from both the client and server.

2.9. Smart Cities

A smart city is a designation given to a city that incorporates information and communication technologies to enhance the performance and the quality of urban services such as transportation, energy, utilities etc. in order to reduce resource consumption and overall costs.

The overarching aim of a smart city is to enhance the quality of living for its citizens through smart technology^[16].

These cities are using an architecture that consists of micro services, APIs and visual interfaces to make use of this connectivity to collect, sort and visualize data. By analyzing the patterns and trend in this data, it is possible to obtain information on a wide range of problems that affect the city and use that vision to make it safer and more productive.

Madrid has an API, among others, that boost this concept, and this one is the EMT Madrid API.

2.9.1. EMT Madrid API

The EMT Madrid API is included in MobilityLabs. MobilityLabs is an open platform that allows the exchange of ideas, initiatives and data in an open manner aimed at the enrichment of the information. Within this platform, it is possible to obtain data, resources and repositories collected by EMT Madrid^[23].

With the Mobility Labs community, EMT Madrid strengthens the development of new technologies to improve urban mobility and promote the use of public transport through applications and services.

This API has many webmethods available to get information related to the buses (lines, stops and timetables), bicycles (stations) and parkings (real time free spaces and list of parkings), among others. It is necessary to be registered on their system to use those methods

If you are a developer, there is a guide and documentation on GitLab in order to provide help and utilities.

After explaining the different technologies, the proposed solution will be presented together with information about the methodology, project management, use cases and data and libraries that have been used.

Chapter 3 - System development

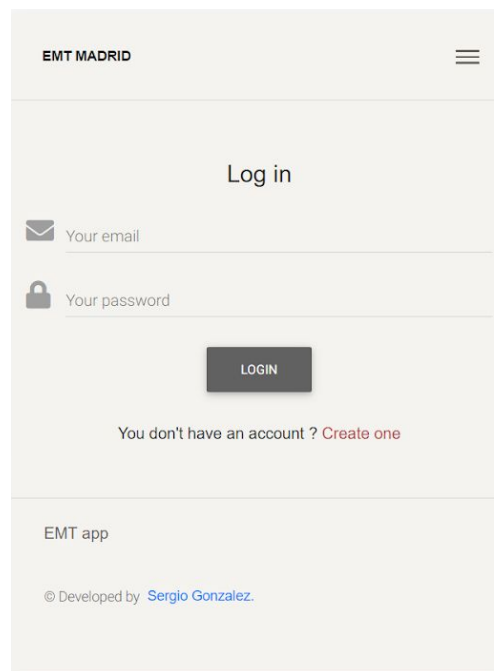
3.1. Application description

In this section, we are going to explain the application that has been implemented. The application is organized in 8 main pages and each one corresponds to a 'web page' that they become a screen in the hybrid mobile application and we also briefly mention how the navigability is.

In the following [link](#), there is a demo with all the functionalities explained below.

Login

This is the first page that is displayed once you have launched the application.



There is a form in which you have to type your email and password in order to access to the system. This authentication is done by Firebase, using its authentication method.

Once you have registered on the system (sign up), you can log in. To check if your credentials are right, the following function, that is provided by Firebase, is used:

```
login: function () {
  firebase.auth().signInWithEmailAndPassword(this.email, this.password).then(
    (user) => {
      this.$router.replace('dashboard')
    },
    (err) => {
      // alert('Oops. ' + err.message)
      alert.log(err.message)
    }
  )
}
```

Buscar por dirección de correo electrónico, número de teléfono o UID de usuario Añadir usuario ↻ ⋮

| Identificador | Proveedores | Fecha de creación | Inicio de sesión | UID de usuario ↑ |
|----------------------|-------------|-------------------|------------------|------------------------------|
| dfdsfds@sdfs.com | ✉ | 4 jun. 2019 | 4 jun. 2019 | 5QkP8hmuiZbA7crrv2IT97mjpit2 |
| hola@gmail.com | ✉ | 15 may. 2019 | 15 may. 2019 | 6PMImAPgX7V83H8s1XIXeCoutlg1 |
| mrsergiogf@gmail.com | ✉ | 27 ago. 2019 | 27 ago. 2019 | BB16EwQGGWStX03D2zp4FhLxG... |
| b@gmail.com | ✉ | 3 ago. 2019 | 3 ago. 2019 | La1NZdS7GeTZo3o4We0ulM5rjt2 |
| sergiogf@gmail.com | ✉ | 27 ago. 2019 | 27 ago. 2019 | 0InHvOK22zPE6SG1dClrmXVwm... |
| sergon02@ucm.es | ✉ | 4 jun. 2019 | 5 jun. 2019 | SQCBCPIFXUcHIHs14ojV3yPWRRi2 |
| a@a.com | ✉ | 20 jul. 2019 | 20 jul. 2019 | YBXf110EeStBwvpzZWlilhngjk2 |
| c@gmail.com | ✉ | 3 ago. 2019 | 27 ago. 2019 | a3UsGhYtBHYQIUbmBp0nmMBIB... |
| sergon@gmail.com | ✉ | 12 may. 2019 | 12 may. 2019 | eaoQNFdehgl7dc3GTkSKsJBfjh1 |
| a@gmail.com | ✉ | 12 jul. 2019 | 12 jul. 2019 | JFCF1xpTqZdC1SpvbnlIKNF8h0q2 |
| hola@ucm.es | ✉ | 3 jul. 2019 | 5 jul. 2019 | sZwFUoTwGqYroMTF80Q7ySEeR2 |

If the credentials are right, the app will be redirected to the ‘dashboard’ page, and if not, it will show you a message, explaining you why it is not correct.

There is an hyperlink ‘Create one’, in case you don’t have an account yet, that redirects you to the sign up process.

Sign up

This page is quite similar to the login page, but in this case it is used to sign up on the system.

EMT MADRID

Sign up

Your name

Your email

Confirm your email

Your password

REGISTER

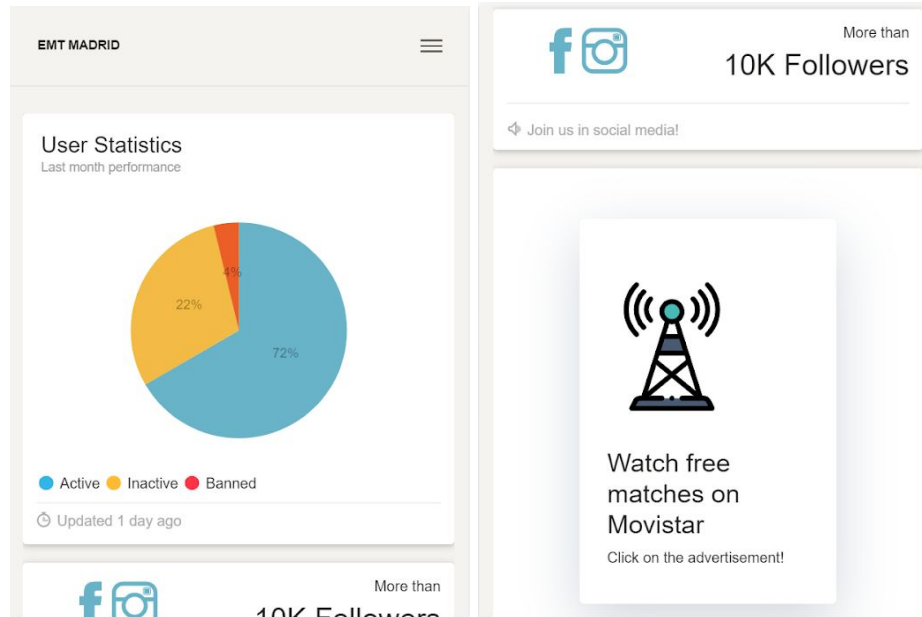
Do you already have one? [Log in](#)

You just have to type your name, email and password. As before, Firebase will check if that account is already on the system, if it exists, it will show you a message saying that, or in case it was created properly, it will redirect you to the login page.

```
signUp: function () {
  firebase.auth().createUserWithEmailAndPassword(this.email, this.password).then(
    (user) => {
      alert("Account created!")
      this.$router.replace('login')
    },
    (err) => {
      alert('Oops. ' + err.message)
    }
  )
}
```

Dashboard

Once you have logged in correctly on the system, the 'dashboard' page will be your main page, from which you can access to all the functionalities.



On the dashboard page, you can find statistics about the users that are already registered on the system, the social networks of the EMT Madrid and one section that will be used to companies that want to put their advertisements on it.

About the advertisements topic, Google Analytics has been chosen to control them. The intention is to not use an intermediary, like Google AdWords, for simplicity. The advertisements are “banners” that are bought offline and I want to monitor them. On the dashboard are advertisements, but also there are more on other pages, so thanks to Google Analytics, we control when a user click on each advertisement.

In order to configure Google Analytics (Appendix D explains how to set up Google Analytics), we have to include it in our project:

```
import VueAnalytics from 'vue-analytics'
Vue.use(VueAnalytics, {
  id: 'UA-140020631-2'
})
```

Also, we have to include an event in that section to know when the user clicks there. Once the user clicks, an event is caught, and we send a request to Google Analytics using the Measurement Protocol.

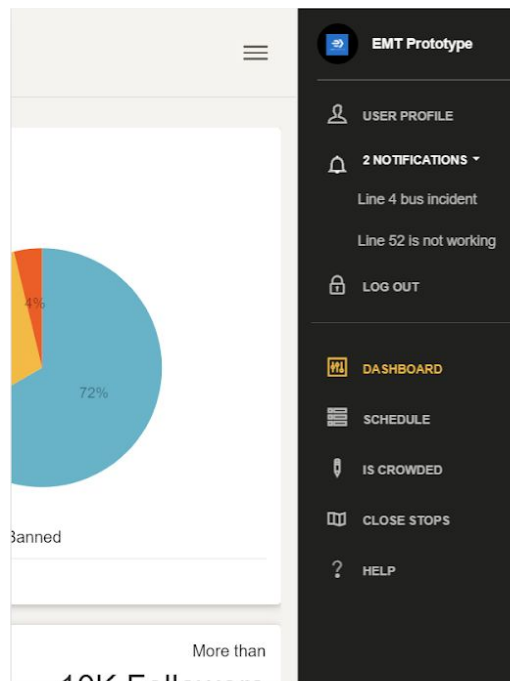
```
this.$http.post('http://www.google-analytics.com/collect?v=1&t=pageview&tid=UA-140020631-2&cid=555&dp=%2FAdvDashboard')
```


We have named these advertisements (AdvDashboard), so if we go to the Google Analytics console, we can check that someone has clicked that advertisement.



On this way, we can check how many people click that advertisement, from where, what device is using etc. and we can estimate, for example, how much money one company have to pay if it wants to put its advertisement there, or if advertisements in that section of the application are working or striking the attention of the user.

Finally, we have a menu that we can display if we click the icon that is at the top on the right.



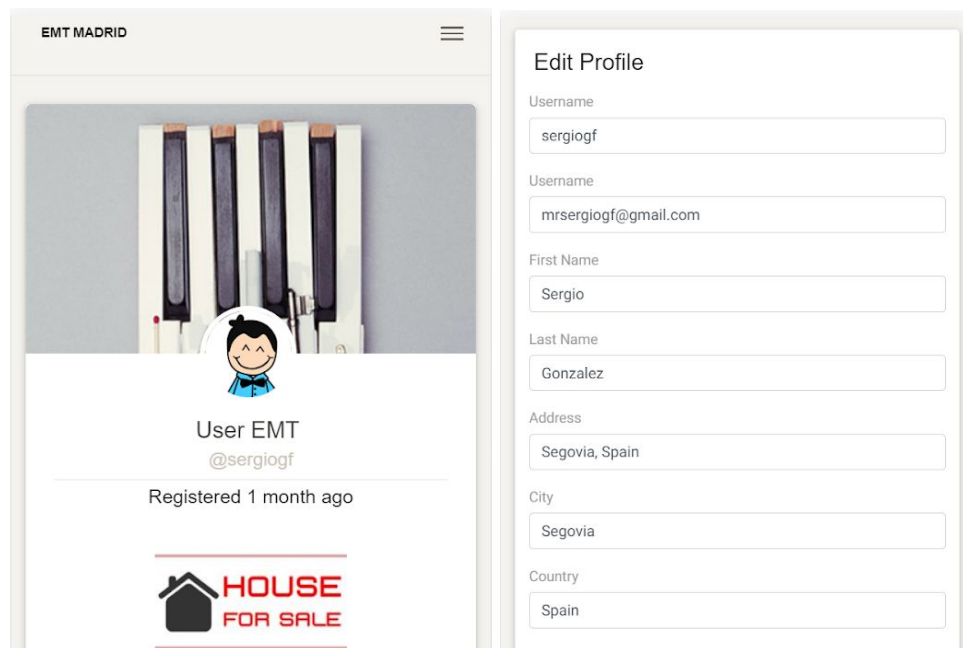
From this menu, you can access to all the pages that are available on the application. In the first section of the menu, there is an element that redirects you to the user profile page, there is another element 'Notifications' that displays you a dropdown with the incidents that exists right now in the buses stops in Madrid, and an icon (Logout) that disconnects your session and redirects you to the login page .

In the second section of the menu, there are five icons in which they are includes the three main functionalities ('Schedule', 'Is crowded' and 'Close stops'), and also the actual Dashboard page, and a page for help.

This menu is common to all the pages, except to the Login and Sign up page, because obviously, you have not accessed the system yet. The menu item you are in right now is highlighted in yellow.

User profile

This page is used to show information related to the user and also the user can update that information.



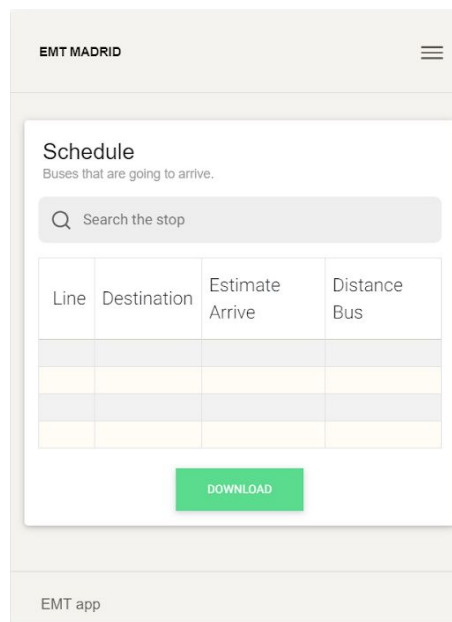
The screenshot displays the user profile page for 'User EMT' (@sergiogf). The profile card on the left includes a profile picture, the name 'User EMT', the handle '@sergiogf', and the text 'Registered 1 month ago'. Below the profile card is a 'HOUSE FOR SALE' advertisement. The right column is titled 'Edit Profile' and contains several input fields for user information: Username (sergiogf), Email (mrsergiogf@gmail.com), First Name (Sergio), Last Name (Gonzalez), Address (Segovia, Spain), City (Segovia), and Country (Spain).

The 'house for sale' image is included in other section destined to advertisements. As it is explained in the Dashboard section, this section will be monitored with the same purpose.



Schedule

This is one of the main functionalities. With it, you can get the buses that are going to arrive to a specific bus stop and also, you can download that schedule.

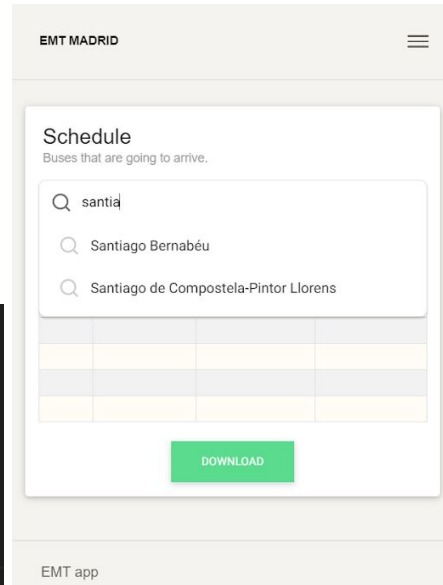


An autocomplete input is provided in order to control the names that the users type. So, first of all of the stops are retrieved to fill the autocomplete.

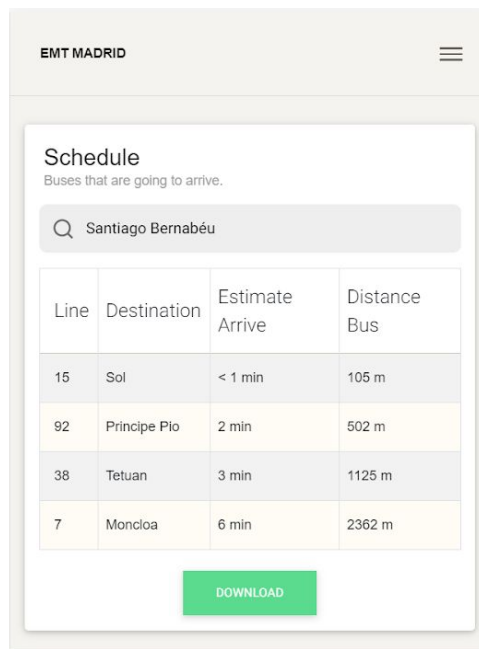
```

initDB(){
  var vm = this
  var db = firebase.database()
  db.ref('data/').on('value', function (snapshot) {
    var objeto = snapshot.val();
    var name;
    for (var i = 0; i < objeto.length; i++) {
      name = objeto[i].name
      vm.namesAux.push({'name':name, 'idStop':i})
    }
  })
},

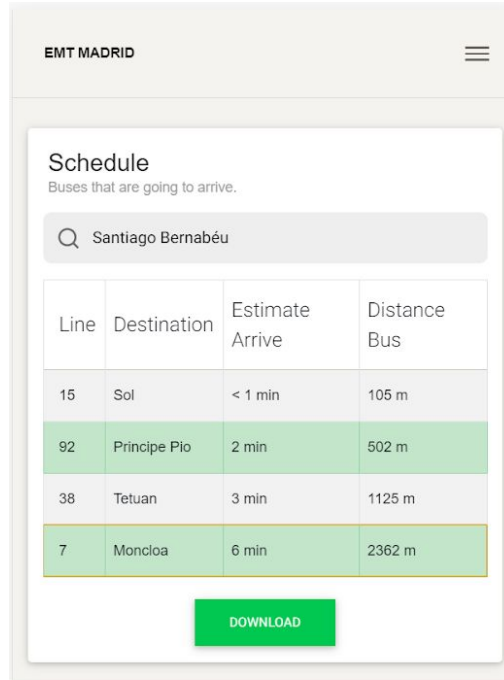
```



Once you have selected the bus stop, a request is made to the EMT api to get the buses that are going to arrive to that stop and the table is completed with information related to those buses.



There is a Download button that let you download that schedule, but first you need to select what lines (rows) are you interested in.



If you select one or more lines, the download button is enabled. The download it will not be a 'typical' download. The data it is send to a sheet on Google Spreadsheets, and you can download or keep it there.

To use Google Spreadsheet (you can check the Appendix C, to set it up), we have used the following library (google-spreadsheet) that uses the protocol OAuth2 to make the requests to Google Spreadsheet and it offers some functions to access and modify the sheets:

- Get information of the worksheets
- Work with rows (values)
- Work with cells (formulas)
- Create and delete sheets

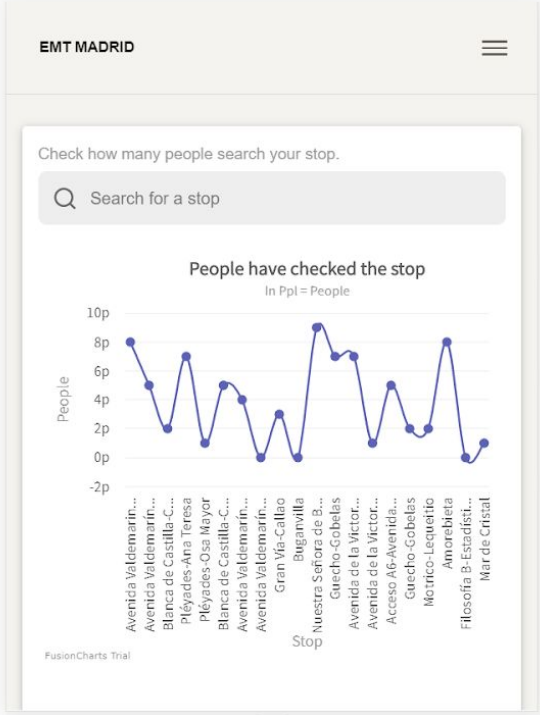
```
Vue.use(Autocomplete)
import GoogleSpreadsheet from 'google-spreadsheet'
Vue.use(GoogleSpreadsheet)
```

Once you have clicked the download button, the information of the rows that you have selected is displayed on the sheet:

| Line | Destination | Estimate arrive | Distance Bus |
|------|--------------|-----------------|--------------|
| 92 | Principe Pio | 2 min | 502 m |
| 7 | Moncloa | 6 min | 2362 m |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

Is crowded

With this functionality it is possible to know how many people have checked one specific stop. A graphic chart is displayed with some random stops and the number of people that have checked those stops.



As in the schedule functionality, there is an autocomplete input and the stops are retrieved to fill it. There is a small difference here. When the data is loaded on the database, none row have a count value, so it is necessary to add it in order to count how many people check each stop.

```

db.ref('data/').on('value', function (snapshot) {
  var objeto = snapshot.val();
  var name;
  var count;
  for (var i = 0; i < objeto.length; i++) {
    name = objeto[i].name
    if(objeto[i].count == null){
      count = 0
    } else {
      count = objeto[i].count
    }
    if(i < 20){ //Summary, 20 firsts.
      aux.push({'label':name, 'value': count})
    }
    vm.auxComplete.push({'label':name, 'value': count})
    namesAux.push({'name':name, 'idStop':i})
  }
})

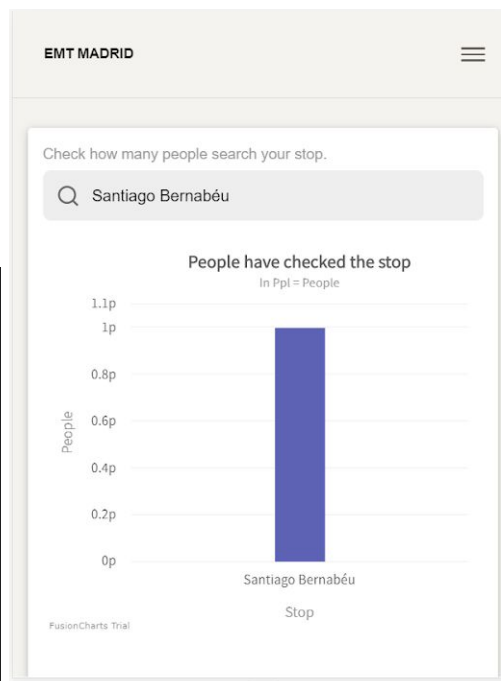
```

Once you have selected the stop in the autocomplete form, a new graphic is displayed to show you that stop with the number of people that have checked it. Also, the database is updated, increasing the count field.

```

handleSubmit(result) {
  var numSearchs
  var vm = this
  var stops = vm.auxComplete
  var i=0
  var ok=false
  var idStop
  while(i < stops.length && !ok){
    if(stops[i].label == result){
      stops[i].value +=1
      var countStop = []
      countStop.push({'label':result, 'value':stops[i].value})
      vm.type = 'column2d'
      vm.dataSource.data = countStop
      ok = true
      idStop = vm.names[i].idStop
    }
    i++
  }
  var query= db.ref('data/' + idStop)
  query.update({'count': stops[i-1].value})
}

```



Close stops

This is the last of the main functionalities. In this one, you can check the closest buses stops considering your current location.

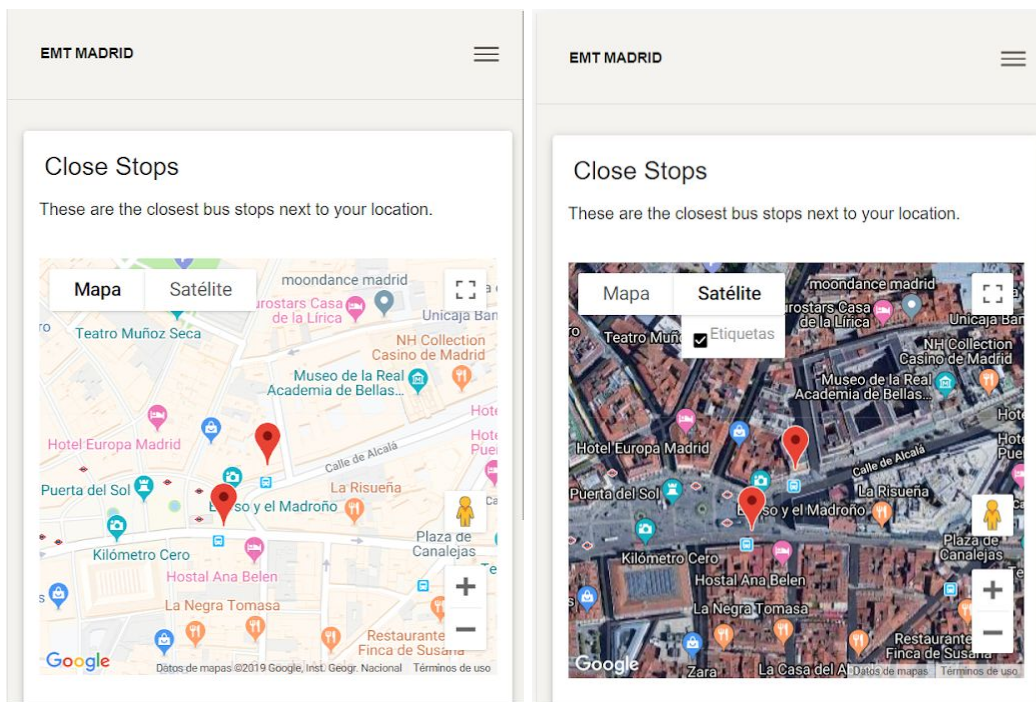
To display this information, Google maps has been used. There is a library for Vue 'vue2-google-maps'. You need an api key to use Google Maps.

```
import * as VueGoogleMaps from "vue2-google-maps";
Vue.use(VueGoogleMaps, {
  load: {
    key: "AIzaSyDtDhERr61E5ZeIsDag0R0n2fde4-QPCxU",
    libraries: "places" // necessary for places input
  }
})
```

A request to the EMT api is made, considering the location of the user, in order to get the closest buses stops:

```
mounted () {
  this.$http.get("https://openapi.emtmadrid.es/v1/mobilitylabs/user/login", { responseType: 'json', headers: { 'email': '...', 'password': '...' } })
  this.$http.get("https://openapi.emtmadrid.es/v1/transport/busemtmad/stops/aroundxy/-3.7048859/40.4178158/200/", { responseType: 'json' })
  var data = response.body.data[0]
  for(var i=0; i < data.stops.length; i++){
    var lat = data.stops[i].geometry.coordinates[0]
    var long = data.stops[i].geometry.coordinates[1]
    var name = data.stops[i].name
    var description = data.stops[i].postalAddress
    this.markers.push({'name': name, 'description': description})
    this.addMarker(lat, long, name, description)
  }
  this.spinnerDis = false
  this.mapDis = true
}
```

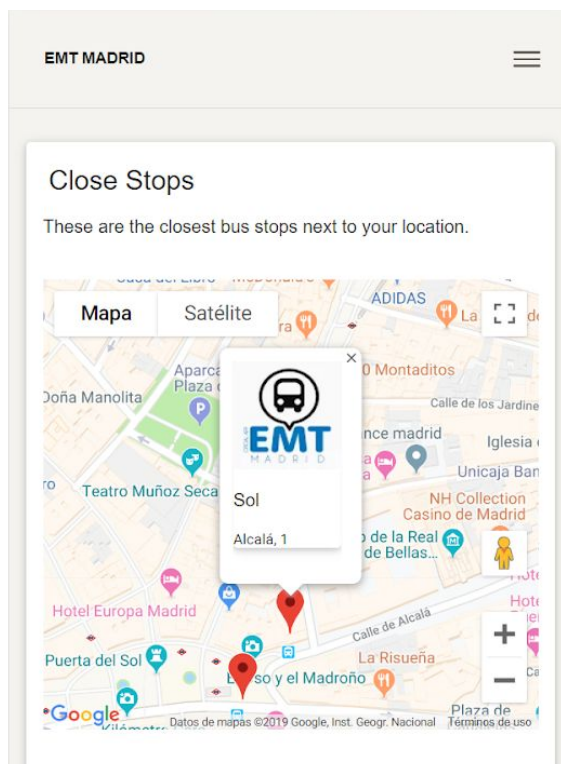
Once we have got the data, we set all the information that is going to be displayed at the markers and the map is loaded:



The information that is set at the markers, it is shown when the user clicks one of them:

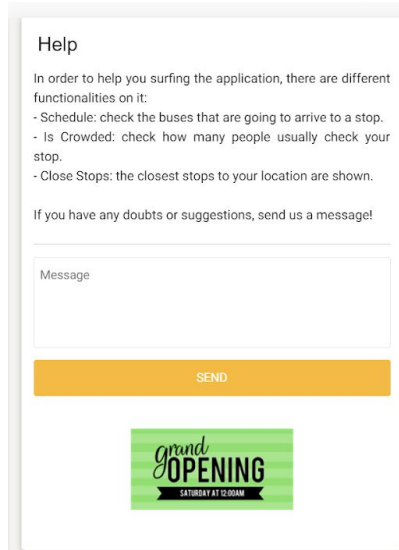

```
getInfowindowContent: function (marker) {
  return (`<div class="card">
    <div>
      <figure class="image is-4by3">
        
      </figure>
    </div>
    <div class="card-content">
      <div class="media">
        <div class="media-content">
          <p class="content">${marker.name}</p>
        </div>
      </div>
      <div class="content">
        ${marker.description}
      </div>
    </div>
  </div>`);
}
```

In the information, it just appears the name of the stop and the street in which is located that stop in Madrid.

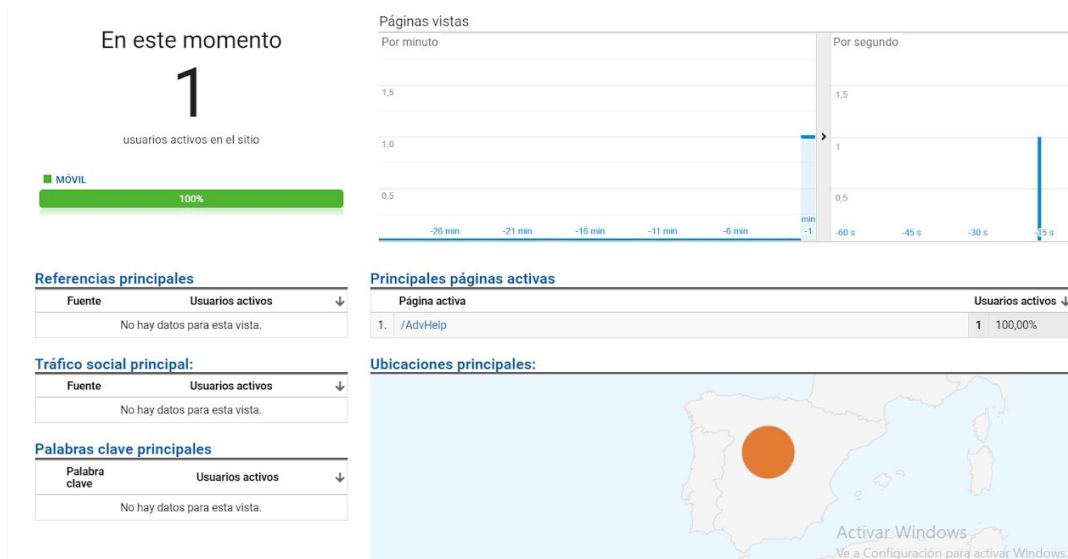


Help

This is a simple page in which it is explained what the main functionalities do, and also there is a form in case the users want to say something about the application: doubts, suggestions etc.



The green image is included in other section destined to advertisements. As it is explained in the Dashboard section, this section will be monitored with the same purpose.



3.2. Methodology and project management

3.2.1. Methodology

The project has been planned by dividing the total time available in several stages. These stages will be: analysis and research, design and implementation, test, deployment and documentation.

In the early stages, all possible alternatives that could be useful when implementing the application are analyzed and documented. In that way, we can determine the viability of each of them in the future.

After that, the chosen alternative is analyzed and a prototype is developed, checking that the viability is good and developing the final software.

Finally, in the last stages, tests are carried out on the software and the whole process is documented.

The different stages are described below:

- **Analysis of requirements and research.** During this stage, first of all there is a brainstorming (method for generating ideas). After deciding one idea, it is needed to explain what the software should do to satisfy the needs of the users who will use it. The analysis and specification of requirements is one of the most important parts, the software produced will have the expected quality, largely depending on this stage. These requirements are determined taking into account the needs to satisfy the final user.
- **Design.** In this stage, one decide how the software will work without going into details and specifying aspects of the technologies. Normally, it is usually done based on diagrams that allow describing the interactions between actors and entities. Also, the components are designing in order to carry out the functionalities of the software.
- **Implementation.** The software's code is implemented, considering all the specifications of the previous stages and giving them the result that is waited. As the functionalities are being implemented, they will be testing at the same time.

- **Application tests and deployment.** This stage consists of checking and verifying that the software correctly performs the tasks indicated in the specification of the requirements. Once this has been done and the tests are successful, the application is deployed to the final users.
- **Documentation.** This stage generates all the documentation related to the software (UML diagrams, tests, user manuals, code etc.), so anyone could understand the project and edit this documentation in the event that modifications or extensions are made.

It is clear the documentation has been done during all these stages, but in the last stage, I was more focused on it, following the execution guidelines.

Due to a changes in the project and because the estimate was not completely accurate, the period initially set in the estimate was not fulfilled, so the tasks were delayed, being reflected in the following Gantt chart [18]:



The last of the project is 7 months, so every month is represented by 2 periods.

*The documentation has been doing during the whole process.

Gantt Diagram

3.2.2. Project management

In order to set a control over the project and manage it more easily, the following tools have been used:

- Google Drive. It has been used to store images, files, documents etc. in the cloud.
- Google Docs. This tool has been used to write the entire documentation.
- GitHub. It has been used as a code version control.

In this way, I have had access to all the things related to my project (code, documentation, files etc.) from everywhere. Also, I find it very useful, in case your laptop gets broken or something happens to it, because you have access to it from every device, as long as you have your account's credentials.

3.3. Use cases

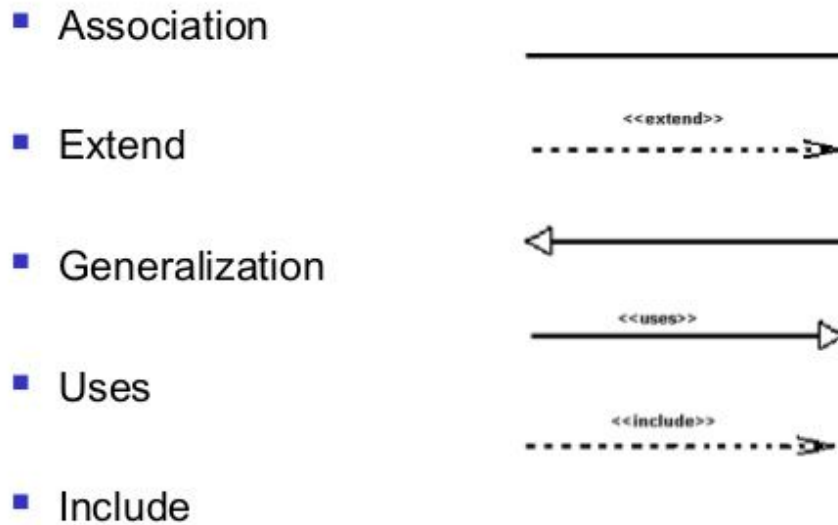
3.3.1. Introduction

The use case diagrams represent the behavior that the system must present using simple language and not too technical. Each use case focuses on describing how to achieve a single goal or task. As a result of this, a use case can be understood as an interaction between actor and system in response to an event.

The use cases are related to each other through various types of relationships:

- Communicates: association between an actor and a use case to describe the involvement in said scenario. It is usually represented with a continuous line.
- Include: inclusion of the behavior of a scenario of a use case in another use case. It is usually represented with a dashed line and with the label "include".
- Extends: denotes that a use case is a specialization of another use case. It is usually represented by a dashed line and labeled "extends".

In addition, an actor can inherit use cases that communicate with another actor by using the generalization notation that is represented by a continuous arrow that goes up from the “child” to the “father” actor ^[17].



3.3.2. Actors

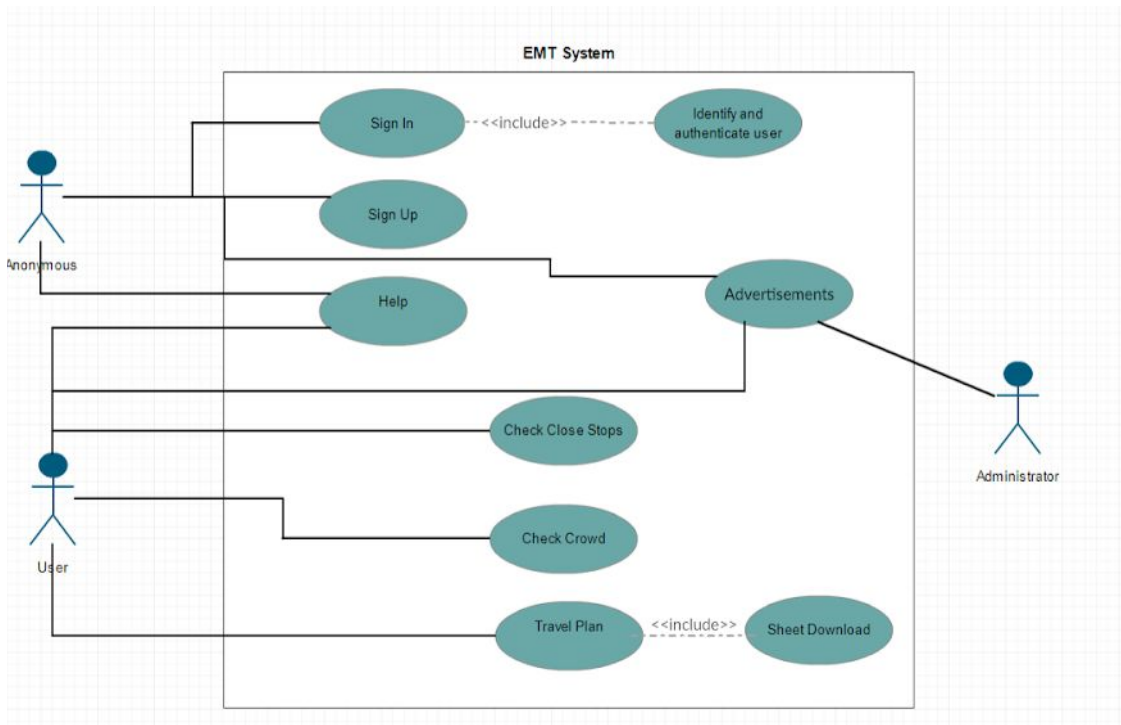
The actors represent the role played by a user or by any other external system that interacts with the application. There are three different actors:

| | |
|-------------|--|
| ACT-01 | Anonymous |
| Description | This actor represents the person who has not registered in the system yet. |

| | |
|-------------|--|
| ACT-02 | User |
| Description | This actor represents the person who is going to access the application to use it. |

| | |
|-------------|---|
| ACT-03 | Administrator |
| Description | <p>This actor represents the developer/main role and is the person who can check the application's performance and the statistics on Google Analytics console.</p> <p>It is considered an external actor and used to specify that there is an external role that checks that.</p> |

3.3.3. Diagram



Use of cases diagram

3.3.4. Specification

| | |
|-----------------|--|
| CU-01 | Sign In |
| Precondition | The user has to have an email. |
| Description | The system should permit the user register its e-mail and one password, in order to sign up on the application. |
| Normal Sequence | <ol style="list-style-type: none">1. The system asks the user the email and password that it wants to use to sign in.2. The user fill both fields (email and password)3. The system shows a message saying that the account has been created correctly.4. The user is redirected to the sign up page. |
| Postcondition | There is a new row on the Firebase authentication section with the email and password of the new user. Also, the user can authenticate on the application with those fields. |
| Exception | The email's format it is wrong or the email is already on the database, so the system will show an error message explaining what happened. |
| Importance | High. |

| | |
|-----------------|---|
| CU-02 | Sign Up |
| Precondition | The user has to have signed in on the application. |
| Description | The system should allow access the user to access the application with its email and password. |
| Normal Sequence | <ol style="list-style-type: none">1. The system asks the email and password to the user in order to give him access to the application.2. The user fills both fields (email and password)3. The system allows the user the access.4. The user is on the home page. |

| | |
|---------------|---|
| Postcondition | The user can access to the different features the application offers. |
| Exception | The email and/or the password are not correct, so the system will show an error message explaining what happened. |
| Importance | High. |

| | |
|-----------------|--|
| CU-03 | Check Close Stops |
| Precondition | The user has logged in with a valid account and it has activated the GPS. |
| Description | The system check the user's location and, in function of, it shows the closest stops. |
| Normal Sequence | <ol style="list-style-type: none"> 1. The user selects the 'Close Stops' option in the menu. 2. The system checks the user's location. 3. The system shows the closest stops. |
| Postcondition | The system has to show the user the closes stops. |
| Exception | The system can't get the user's location. |
| Importance | High. |

| | |
|-----------------|--|
| CU-04 | Check Crowd |
| Precondition | The user has logged in with a valid account and it has to know the name of the stop. |
| Description | The system checks the stop that the user has written and it shows a chart with the number of people that has searched that stop and at what time. |
| Normal Sequence | <ol style="list-style-type: none"> 1. The user selects the 'Crowd' option in the menu. 2. The system checks the database with that stop. 3. The system shows the chart. |

| | |
|---------------|--|
| Postcondition | The system has to show the chart with the different hours and number of people that has searched it. |
| Exception | The stop that the user types does not exist. |
| Importance | High. |

| | |
|-----------------|---|
| CU-05 | Schedule |
| Precondition | The user has logged in with a valid account and it has to know the name of the stop he wants to check. |
| Description | The system check the stop that the user has written and it shows all the buses that will arrive. |
| Normal Sequence | <ol style="list-style-type: none"> 1. The user selects the 'Schedule' option in the menu. 2. The system calls the EMT API in order to check the stop. 3. The system shows the buses. |
| Postcondition | The system shows all the buses that are going to arrive. |
| Exception | There are no buses that will arrive to that stop. |
| Importance | High. |

| | |
|-----------------|---|
| CU-05 | Sheet Download |
| Precondition | The user has logged in with a valid account and selected at least a row from that option. |
| Description | The system provides a download button to download that information on a Google sheet. |
| Normal Sequence | <ol style="list-style-type: none"> 1. The user clicks on the download button. 2. The system download the document with the selected rows. |
| Postcondition | The system open the sheet with the information selected. |

| | |
|------------|---|
| Exception | The download has not been possible because the Google Spreadsheet API is not working. |
| Importance | High. |

| | |
|-----------------|--|
| CU-07 | Advertisements |
| Precondition | The user has logged in with a valid account. |
| Description | The system shows advertisements in different places and the user can click them. |
| Normal Sequence | <ol style="list-style-type: none"> 1. The user signs in on the application. 2. There are different advertisements according to the page. 3. The user can click one of the advertisements and the user will be redirected to the link that is in that advertisement. |
| Postcondition | One browser is opened with the correspondent advertisement and the click is registered on Google Analytics. |
| Exception | The webpage that is linked is not working. |
| Importance | High. |

| | |
|-----------------|---|
| CU-08 | User profile |
| Precondition | The user has logged in with a valid account. |
| Description | There are some information related to the user that can be modified by itself. |
| Normal Sequence | <ol style="list-style-type: none"> 1. The user open the menu and select the option 'User profile'. 2. The user types the new information and press the button 'Update'. |
| Postcondition | The information is updated. |
| Exception | None. |
| Importance | Medium. |

| | |
|-----------------|--|
| CU-09 | Help |
| Precondition | None. |
| Description | This option provides help and instructions to do all the actions the application offers. |
| Normal Sequence | 1. The user open the menu and select the option 'Help' |
| Postcondition | The system has to show the instructions and help to the user. |
| Exception | None. |
| Importance | Low. |

3.4. Data and libraries

Vue has been chosen as a framework to develop this prototype due to all the advantages that it offers and that can be checked in the chapter 2 of this master thesis.

In order to facilitate the work to a person that really wants to try this software, there is a manual for setting up Vue and create a new simple Vue project, in the Appendix A of this master thesis.

The code can be downloaded on [GitHub](#) and used by anyone due to it has a MIT license.

First of all, this is the main structure of the code:

| | | | |
|---------------|------------------|---------------------|--------|
| node_modules | 06/08/2019 20:56 | Carpeta de archivos | |
| src | 03/08/2019 16:21 | Carpeta de archivos | |
| .babelrc | 20/05/2019 13:13 | Archivo BABELRC | 1 KB |
| .editorconfig | 20/05/2019 13:13 | Archivo EDITORCO... | 1 KB |
| .eslintrc | 20/05/2019 13:13 | Archivo ESLINTRC | 1 KB |
| package | 06/08/2019 20:56 | Archivo JSON | 2 KB |
| package-lock | 06/08/2019 20:56 | Archivo JSON | 503 KB |
| vue.config | 20/05/2019 13:13 | Archivo JS | 1 KB |
| yarn.lock | 20/05/2019 13:13 | Archivo LOCK | 251 KB |

The folder `node_modules` contains all the modules and libraries that are necessary for the project. All the files that are in the main folder, like `package.json` / `vue.config` are just files to configure the project. In the other hand, we have the folder `src`, that is the biggest one, and we can find in it many folders and elements:

| | | | |
|------------------------|------------------|---------------------|------|
| assets | 02/08/2019 18:33 | Carpeta de archivos | |
| components | 02/08/2019 19:30 | Carpeta de archivos | |
| layout | 20/05/2019 13:13 | Carpeta de archivos | |
| pages | 05/08/2019 20:57 | Carpeta de archivos | |
| plugins | 20/05/2019 13:13 | Carpeta de archivos | |
| router | 20/05/2019 13:13 | Carpeta de archivos | |
| App | 04/08/2019 18:30 | Archivo VUE | 1 KB |
| google-generated-creds | 17/04/2019 16:53 | Archivo JSON | 3 KB |
| main | 06/08/2019 18:51 | Archivo JS | 2 KB |

The ‘assets’ folder is used to store all the images, style sheets, special styles for a specific component, fonts etc.

The ‘components’ folder contains all the ‘small’ components that are developed in a way that can be reused by different pages, like cards, buttons, arrows etc.

The ‘layout’ folder was created in order to store all the menus, navbars, top and bottom contents that will be included.

The ‘pages’ folder contains all the main pages that will be displayed on the application and that will use all the folders have been explained before (styles, components, menus etc.).

The ‘plugins’ folder has some plugins that are necessary to display the pages.

The ‘router’ folder contains the routes of the different pages and those files specify ‘the life cycle’ of the application.

Finally, the main.js is the main class of the project. It is where the application will get initialized, and after it, it will appear the main page (App.vue). The 'google-generated-creds' is just a file that contains the credentials necessary to use the Google Platform.

Every .vue file is structured in 3 different parts, but it is recommended to use just 2 of those 3 parts:

```
1 <template>
2 <!-- Material form login -->
3 <form>
4   <p class="h4 text-center mb-4">Log in</p>
5   <div class="grey-text">
6     <mdb-input label="Your email" icon="envelope" type="email" v-model="email" required/>
7     <mdb-input label="Your password" icon="lock" type="password" v-model="password" required/>
8   </div>
9   <div class="text-center">
10    <mdb-btn @click="login" color="grey">Login</mdb-btn>
11  </div><br>
12  <p class="text-center mb-4">You don't have an account ? <router-link to="/signup">Create one</router-link></p>
13 </form>
14 <!-- Material form login -->
15 </template>
16
17 <script>
18 import firebase from 'firebase'
19 import { mdbInput, mdbBtn } from 'mdvvue';
20 export default {
21   name: 'login',
22   components: {
23     mdbInput,
24     mdbBtn
25   },
26   data () {
27     return {
28       email: '',
29       password: '',
30     }
31   },
32   methods: {
33     login: function () {
34       firebase.auth().signInWithEmailAndPassword(this.email, this.password).then(
35         (user) => {
36           this.$router.replace('dashboard')
37         },
38         (err) => {
39           // alert('Oops, ' + err.message)
40           console.log(err.message)
41         }
42       )
43     }
44   }
45 }
46 </script>
47 <style>
48
49 </style>
50
```

<template> : it is used to implement the front-end of the page/component, what it will be visualized.

<script>: it is used to implement the back-end of the page/component, to implement all the performance of the page/component: queries BD, calls to APIS etc.

<style>: this is the part that is recommended not to use, and have all the styles store in style sheets and include them.

In our main class (App.vue), we just define the <router-view> to navigate through all the pages, and the sidebar (menu) to change between pages:

```
<template>
  <div :class="{ 'nav-open': $sidebar.showSidebar}">
    <router-view></router-view>
  </div>
</template>
```

We can navigate through all the pages, thanks to the router, so in the router it is necessary to define all the posibles paths. As we can see, the 'main' path or first page that will be display, it will be the 'login' page:

```
const routes = [
  {
    path: "/",
    component: DashboardLayout,
    redirect: "/login",
    children: [
      {
        path: "dashboard",
        name: "dashboard",
        component: Dashboard
      },
      {
        path: "login",
        name: "login",
        component: Login
      },
      {
        path: "signup",
        name: "signup",
        component: SignUp
      },
      {
        path: "stats",
        name: "stats",
        component: UserProfile
      },
      {
        path: "crowded",
        name: "Crowded",
        component: Crowded
      },
      {
        path: "closeStops",
        name: "Close Stops",
        component: CloseStops
      },
      {
        path: "schedule",
        name: "Schedule",
        component: Schedule
      },
      {
        path: "help",
        name: "Help",
        component: Help
      }
    ]
  }
]
```

These are the 'common' and essential components that are needed to launch the application and to navigate through it.

First of all, we are going to explain how we get the data which we are going to work with, and then, what libraries we have used for the front-end.

3.4.1. Data

With the aim of bringing the prototype closer to a real application, we decided to use the EMTMadrid-MobilityLabs API . This is an API that provides information about the public transport in Madrid (buses, bicycles, parkings etc.). We just focused on the bus section, because it is the ones that has many micro-services, the others are very simple.

So, the idea was to get all the data related to the buses stops and work with it, implementing different functionalities.

In order to make the requests to the API, it is necessary to be registered on it (you have to do it on their website) and after it, you need to get an Access Token to make the requests, so the first step before to make a request, is login and get the access token:

```
this.$http.get('https://openapi.emtmadrid.es/v1/mobilitylabs/user/login',  
{responseType: 'json', headers: {'email': 'xxxx@xxxx', 'password': 'xxxxx',  
'X-ApiKey': 'xxxxx', 'X-ClientId': 'xxxx'}})
```

With the response of this request, you will get the access token, and you can make other requests to a specific services.

In the whole application, we are going to use this API 3 times:

Load the initial data. At first, we need to store the data of the stops in our database (Firebase).

To use Firebase as a real-time database, we have to include it in our project (you can check the Appendix B, to set up Firebase):

```
var firebaseConfig = {  
  apiKey: 'XXXXXXXXXXXXXXXXXXXXXXXXXXXX',  
  authDomain: 'XXXXXXXXXX.firebaseio.com',  
  databaseURL: 'https://XXXXXXXX.firebaseio.com',  
  projectId: 'XXXXXXXX',  
  storageBucket: 'XXXXXXXXXX.appspot.com',  
  messagingSenderId: '35XXXXX',  
  appId: '1:354XXXX:web:29ee7XXXX'  
}
```

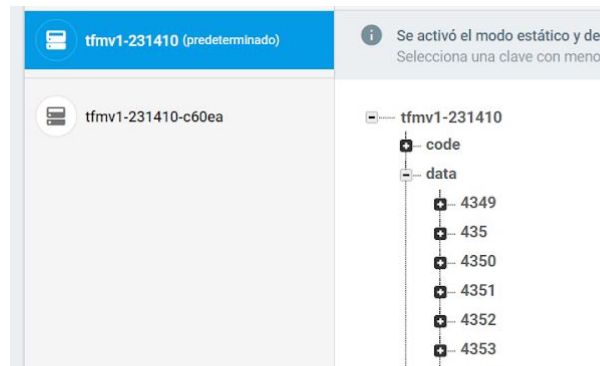
The .json of the stops it is quite big, so it was decided to load that .json directly to Firebase through the console it provides. But anyway, a function was implemented to do it once the application it is initialized, in the main.js:


```

var db = firebase.database()
db.ref('stops').on('value', function (snapshot) {
var objeto = snapshot.val()
if(objeto == null){ // Get database
this.$http.get('https://openapi.emtmadrid.es/v1/mobilitylabs/user/login', {responseType: 'json', headers: {'d
// success callback
var accessToken = response.body.data[0].accessToken
this.$http.post('https://openapi.emtmadrid.es/v1/transport/busemtmad/stops/list',{headers: {'accessToken': ac
var searches = []
for (var i = 0; i < response.body.data.length; i++) {
searches[i] = response.body.data[i].name
}
db.ref('stops').push(this.searchs)
}), response => {
alert('error2')
}
}), response => {
// error callback
alert('error')
}
}

```

So, this is the aspect that our Firebase database should present, after loading the data. The numbers are the identifiers of each stop.



Schedule data. Using the service of ‘Time arrival bus’, we will get the buses that will arrive soon to a specific bus stop.

```

this.$http.post('https://openapi.emtmadrid.es/v1/transport/busemtmad/stops/'+
idStop+'/arrives/all/',{headers: {'accessToken': response.body.data[0].accessToken}
, body: {"mode": "raw", "raw": bodyJSON}})

```

Close stops. Using the service of ‘Stops around geographical point’, we will get the buses stops around one geographical point in a specific radius.

```

this.$http.get('https://openapi.emtmadrid.es/v1/transport/busemtmad/stops/around
xy/XX/YY/MM/', {responseType: 'json', headers: {'accessToken': response.body.data[0].a
ccessToken}})

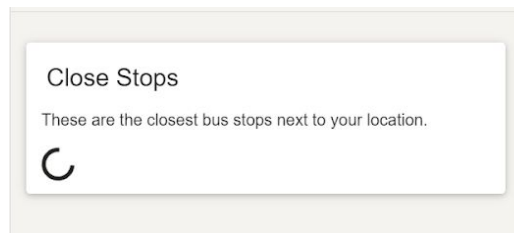
```

3.4.2. Libraries

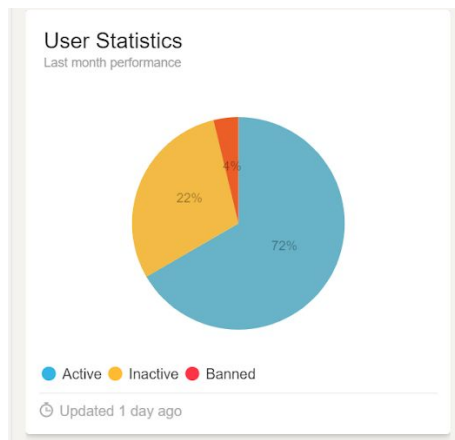
For the front-end part, several libraries have been used, among which stand out the following ones:

Bootstrap Vue. With Bootstrap you can build responsive, mobile-first projects and it gives you a wide range of possibilities in terms of components⁹. Some of these components are:

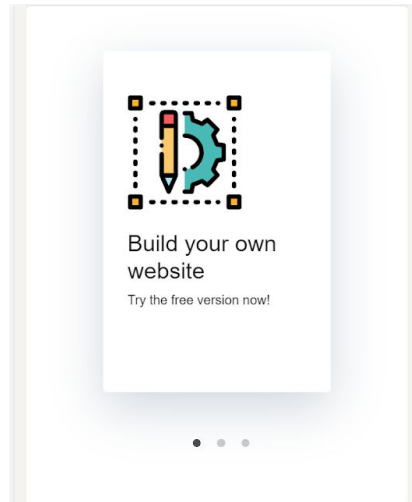
- Spinners: used when the data is loading.



- Cards: to present the data in a cuter way.



- Carousels: present different images at the same time.



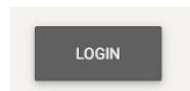
- And many more like buttons, forms, modals etc.

Vuetify. It is a Material Design component framework that provides many tools necessary to create beautiful content rich applications [9]. Accompanying Bootstrap, it has been used for many things:

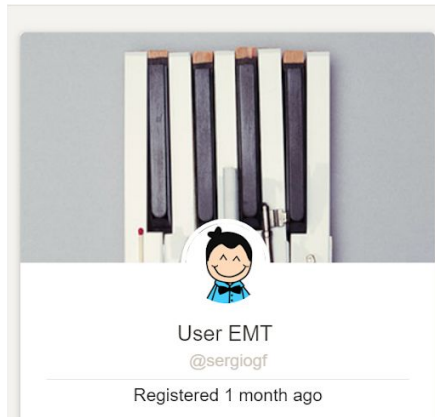
- Tables: to present the data received from the api.

| Line | Destination | Estimate Arrive | Distance Bus |
|------|--------------|-----------------|--------------|
| 15 | Sol | < 1 min | 105 m |
| 92 | Príncipe Pio | 2 min | 502 m |
| 38 | Tetuan | 3 min | 1125 m |
| 7 | Moncloa | 6 min | 2362 m |

- Buttons: in order to submit some events.

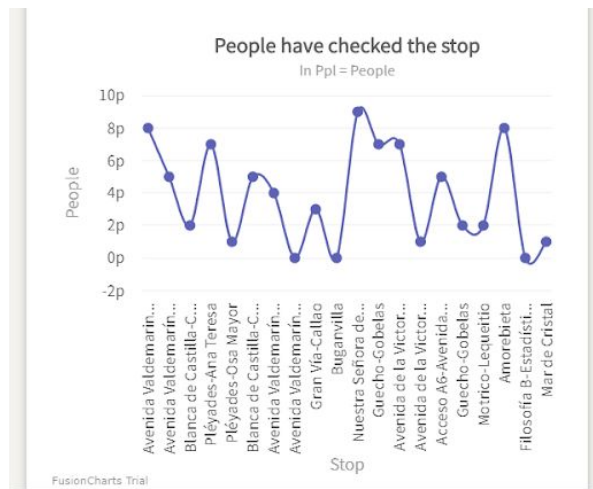


- Avatars: decorating the user profile section.

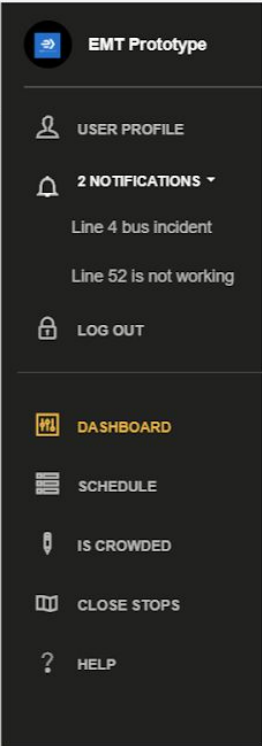


- And many more like dividers, colors, typography etc.

FusionCharts. This library is used specially for charts due to it has so many types of charts and a lot of documentation to include them on your application.



Themify. This library has been used mostly for the icons that are on the whole application, and also, for the font used in the menu.



Chapter 4 - Tests and results

This chapter documents the tests performed to evaluate the software performance, and also the queries and results obtained after evaluating and obtaining feedback with real users.

The results are shown in graphs and images where it is described the performance of the software and a critical analysis of the results is carried out deciding if the system is ready for distribution and, therefore, is valid for other users to use it.

There are two types of tests:

- White box tests: they are performed on the internal functions of a module, class, component etc.
- Black box tests: they check that the functional requirements have been fulfilled and respected. In other words, they allow to obtain sets of entry conditions that fully cover all the requirements.

The white box tests have been carried out at the same time as the project has been implementing.

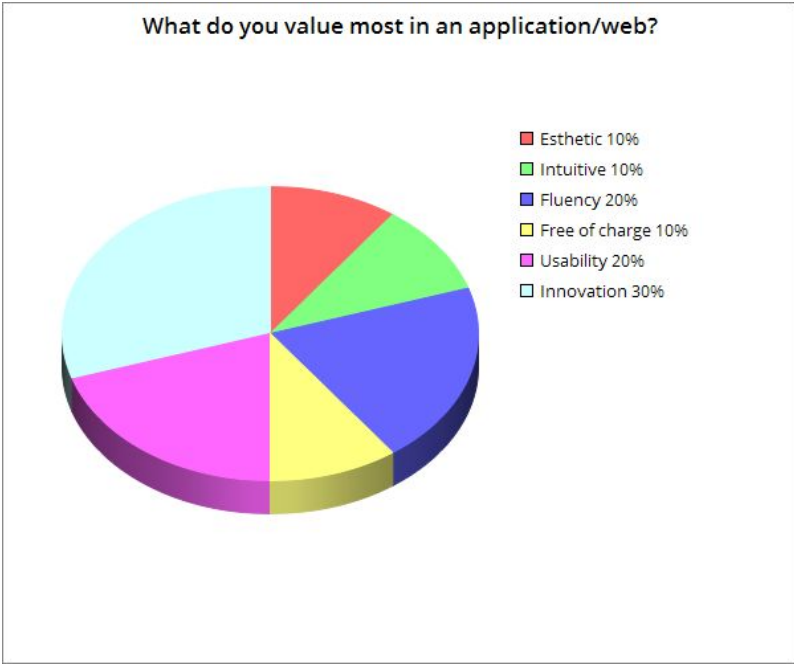
In the other hand, the black box tests, were created and described in the analysis phase (use cases), so in order to not repeat the same information, all the post conditions for each use case, they have been successfully got and completed.

4.1. Initial evaluation

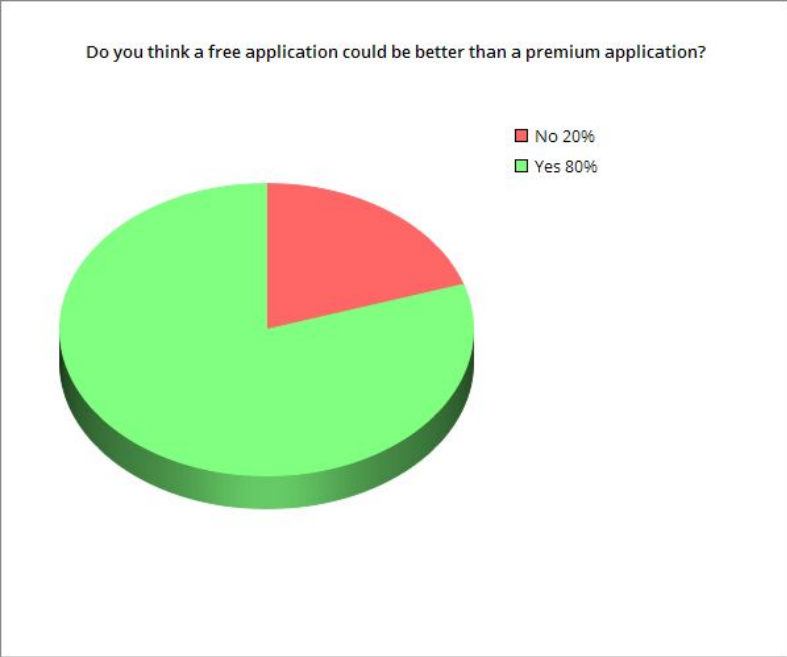
At the beginning of the project, a survey was conducted on different people, to try to get what they value most in an application. In this way, the application would be oriented to one side or other.

The survey was done in Google Doodles and it contained 6 questions without multiple choice. All these people are people that work in jobs related to computers or people that use applications on their smartphones every day.

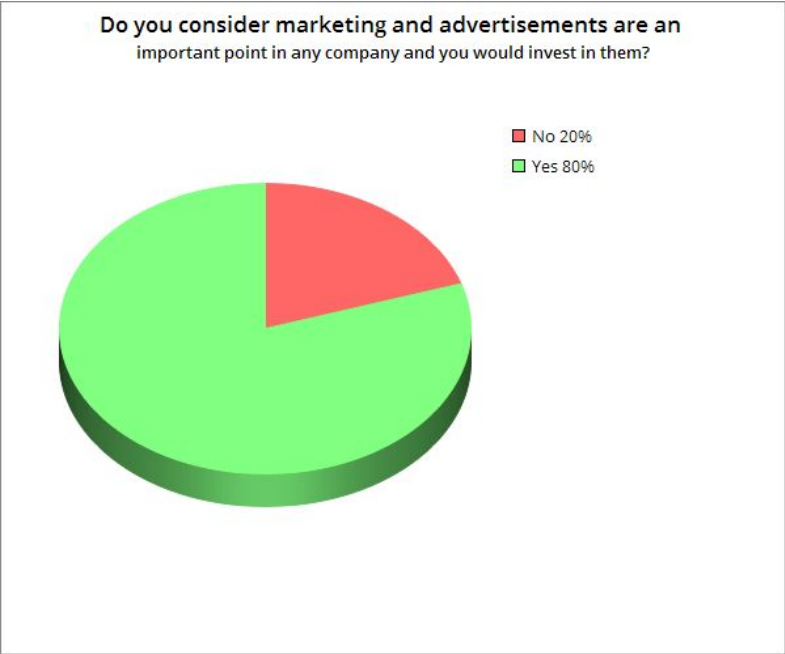
Question 1: What do you value most in an application/web?



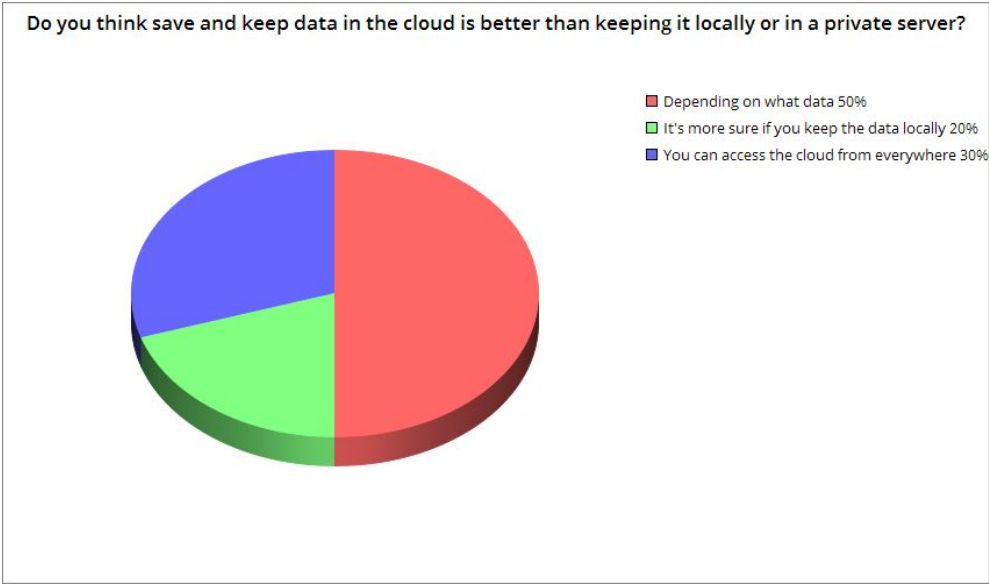
Question 2: Do you think a free application could be better than a premium application?



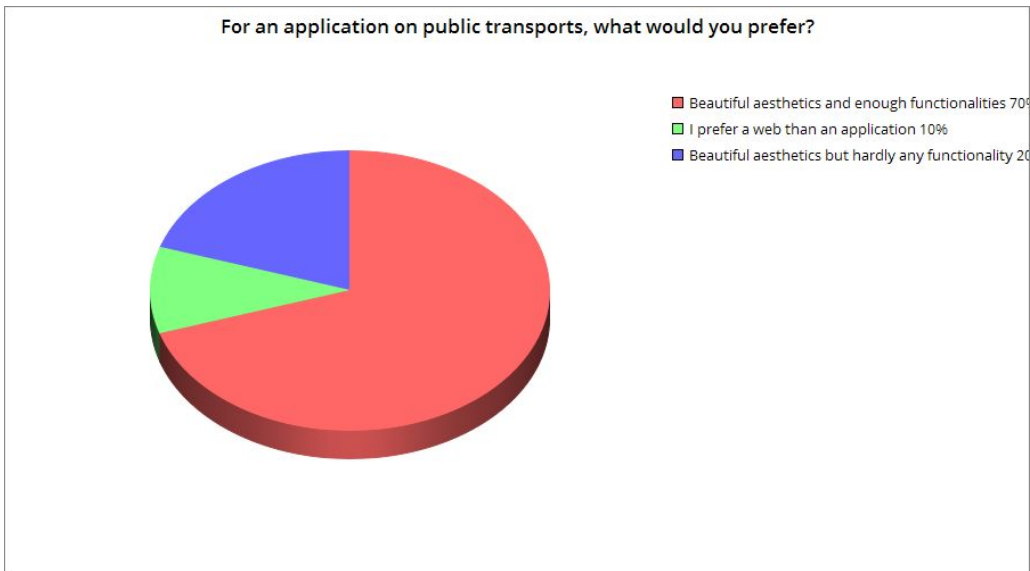
Question 3: Do you consider marketing and advertisements are an important point in any company and you would invest in them?



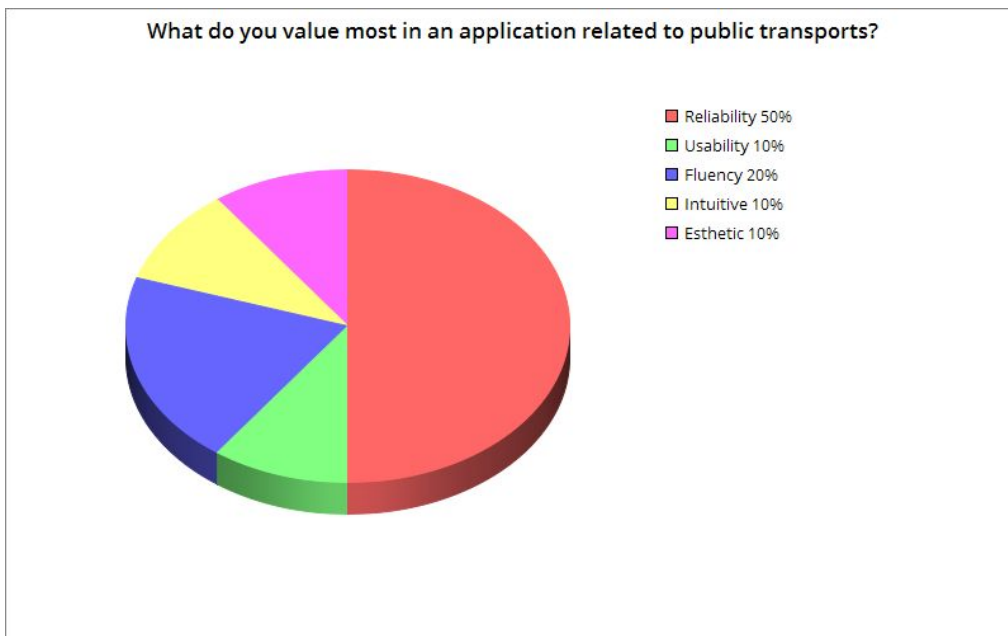
Question 4: Do you think save and keep data in the cloud is better than keeping it locally or in a private server?



Question 5: For an application on public transports, what would you prefer?



Question 6: What do you value most in an application related to public transports?



4.2. Practical evaluation with real users

The objective of the practical evaluation with real users is to check the functionality and usability of the software with a relevant set and the most realistic possible users to ensure that the project can be used without problems. In this way, it is ensured to what extent the requested requirements and functionalities are accomplished.

The second objective is to provide general feedback to the development team so it can make improvements faster. The validation will focus on the implementation of the project, the functionalities that users can see and prove of the software.

The evaluation criteria will be based on the functional requirements from the point of view of the user who will use the platform. The most important variables are listed below, next to a brief description:

- Personal user experience with the platform: customer satisfaction, learning curve, interactivity.
- User productivity: time spent to configure the platform, quality of the results, error rate etc.
- Acceptance: user acceptance rate.

Based on these variables, the platform will be judged according to the observed effect on users behavior and personal experience. The results of these tests are obtained through a questionnaire completed by 5 users that offered themselves to prove the software.

The platform's user interface has been designed to be simple and versatile. The software is given to the use, on the login page and the user can freely interact with the software as he/she wants. Also, he or she receives a paper with the following information:

Information for testing

The objective of this test is that you can play freely with the application and that you can discover the functionalities of it. However, this document is given to you in order to present the application a little.

First of all, this is a prototype, which tries to bring the ideas explained in this project to a real application.

The first screen that appears is the login screen. If you do not currently have an account, you can register in the system by accessing signUp.

After being authorized in the system, you are now in the dashboard of the application (home). On the right side, you can access to the menu in which you have all the functionalities you can try: 'Close Stops', 'Schedule' and 'Is Crowded'. Also, there is an option to ask help, other to check your profile and other to show notifications related to incidents.

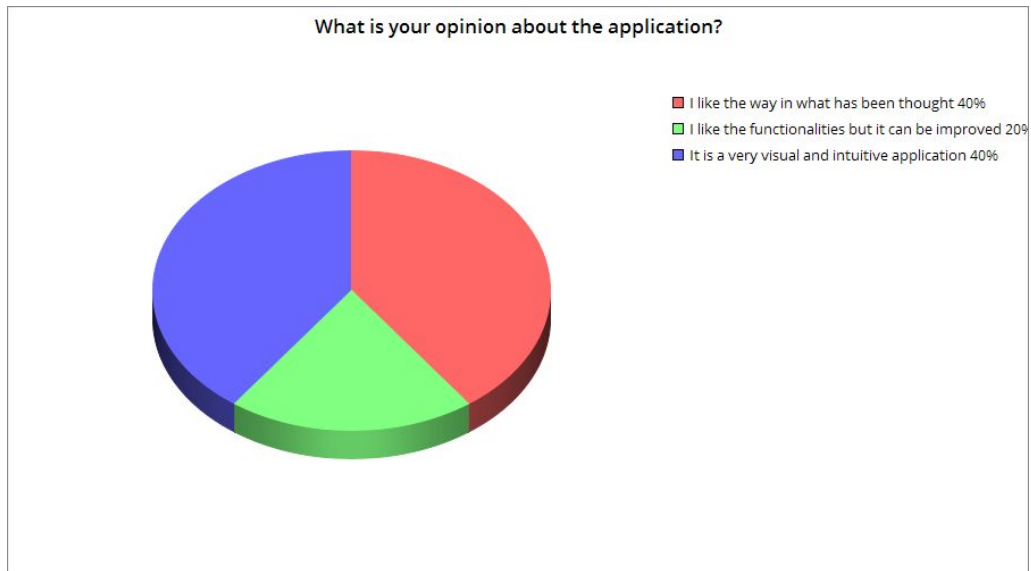
- Close stops: considering the location where you are, the closest bus stops are shown.
- Schedule: it gives you the possibility to check and save, the buses that are coming (line, time to arrive etc.) to a specific bus stop introduced by the user.
- Is Crowded: introducing the name of one bus stop, it shows you how many people have checked it.

Feel free to tell us comments, suggestions, doubts etc. while you are trying our software.

4.3. Results of the practical evaluation

The users were asking to do a survey after the testing. The results are attached below, grouping them with all the comments, suggestions, doubts etc. obtained by the users in each functionality while they were playing.

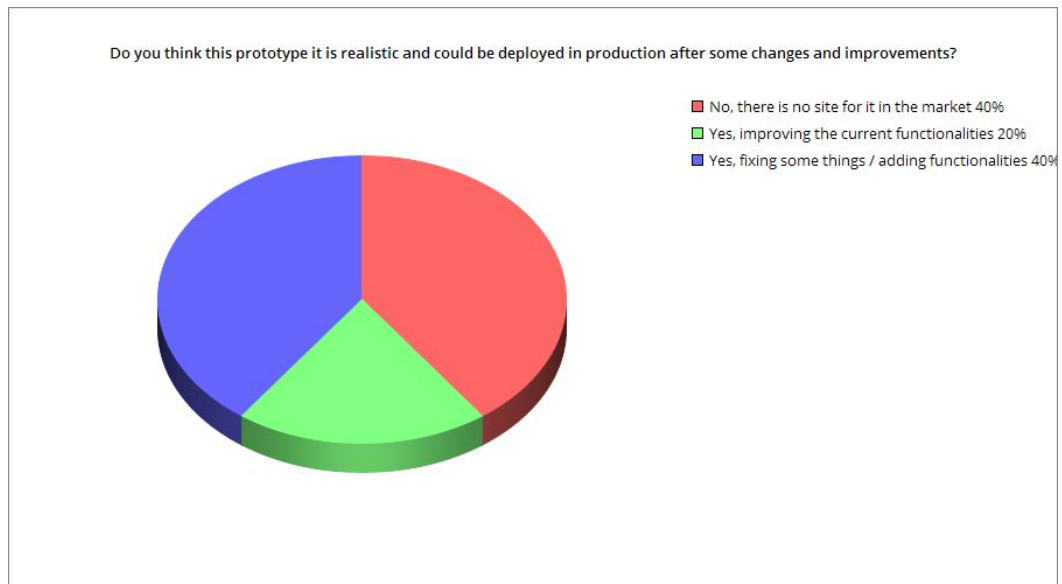
Q1- What is your opinion about the application?



Comments:

- It is so easy to use it and I like the aspect of it.

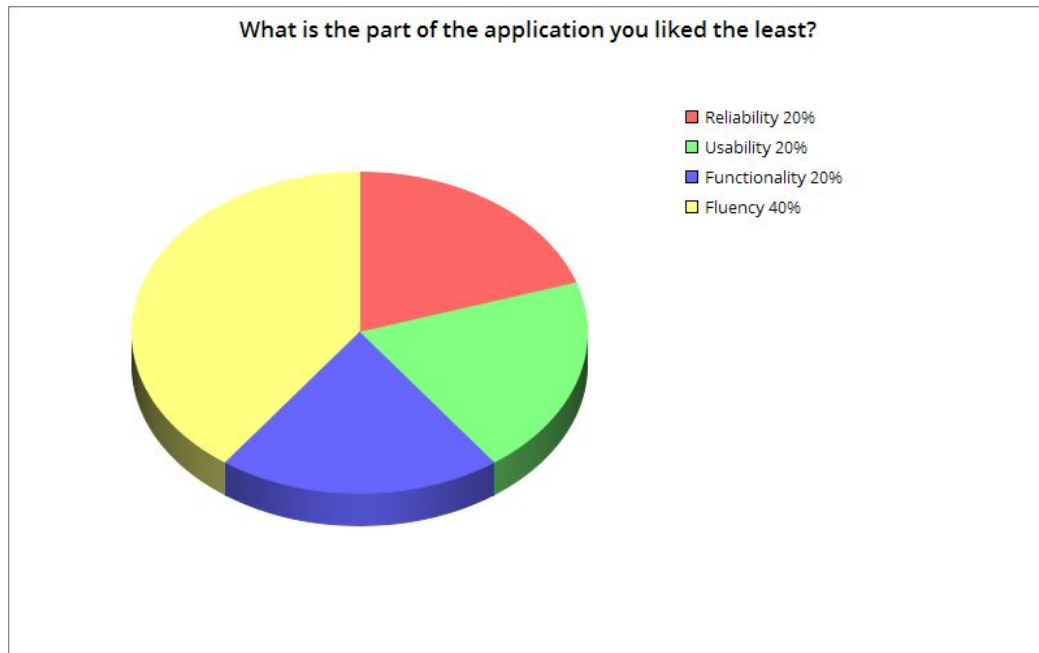
Q2- Do you think this prototype it is realistic and could be deployed in production after some changes and improvements?



Comments:

- It is different than other transports app, the style is more modern.

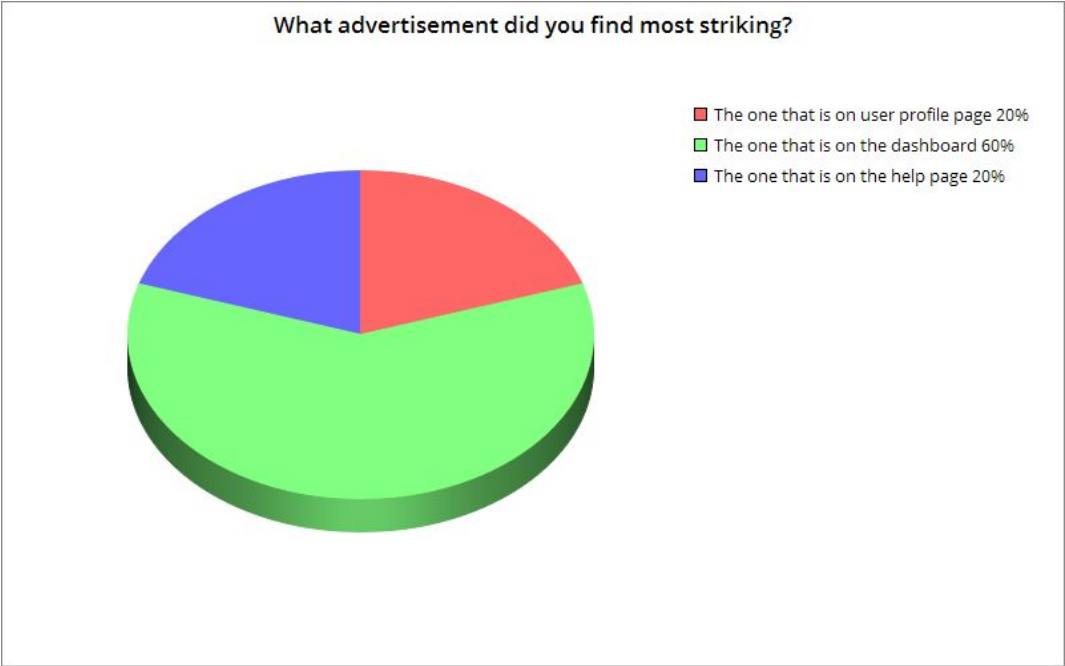
Q3- What is the part of the application you liked the least?



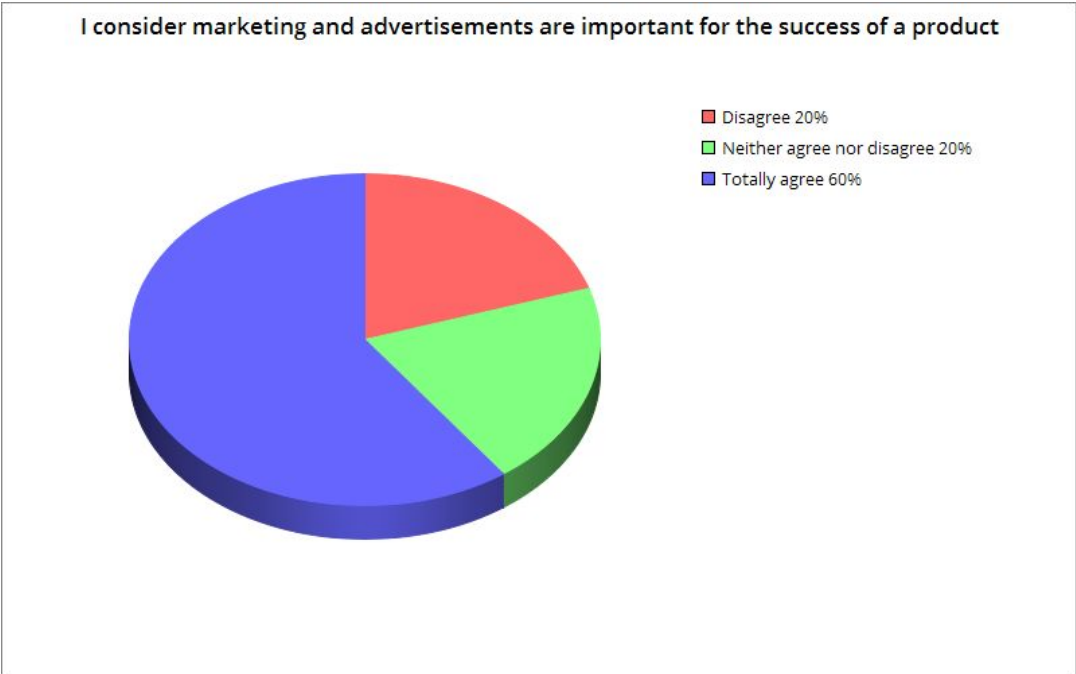
Comments:

- The esthetic is pretty nice!
- I do not think the functionalities are bad ideas but maybe they need a plus.

Q4 – What advertisement did you find most striking?



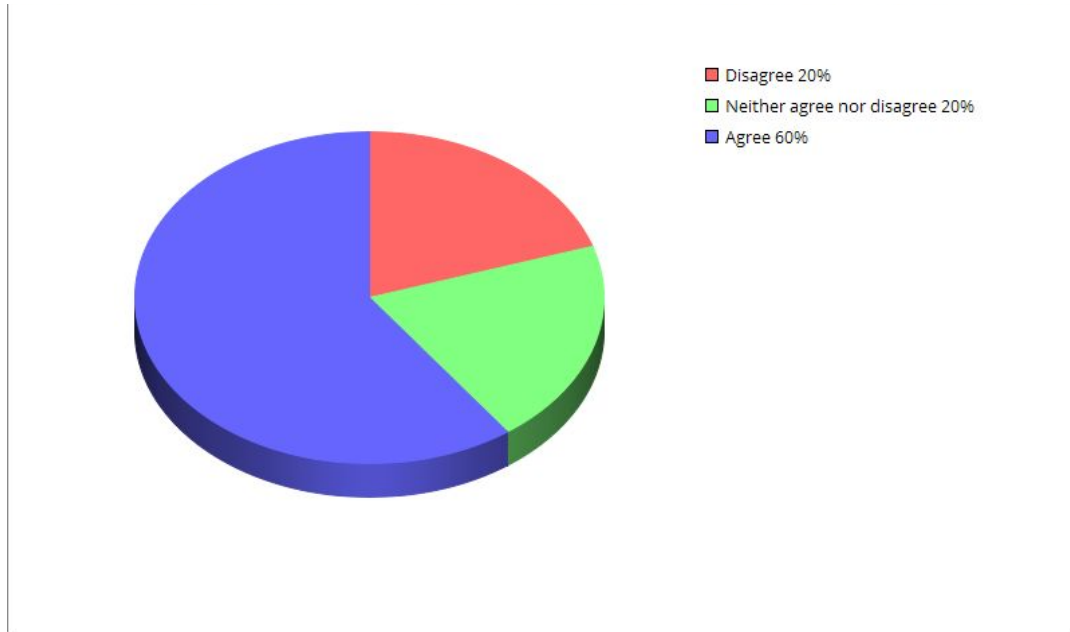
Q5- I consider marketing and advertisements are important for the success of a product



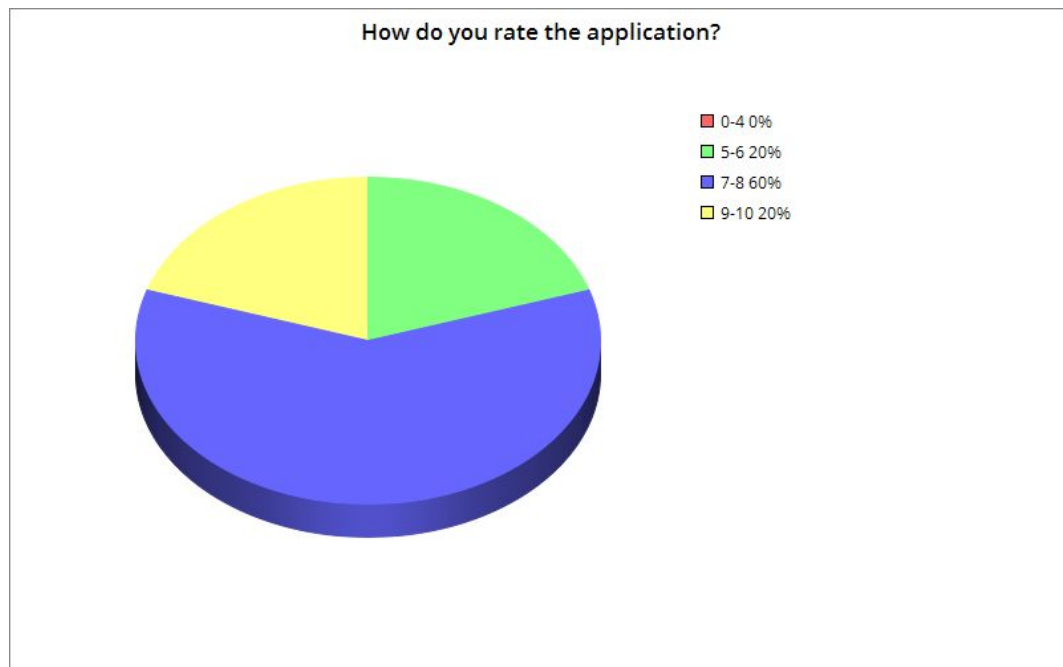
Comments:

- Using Google Analytics you will get an idea of how advertisements works and you can know for what advertisement the clients will pay more.

Q6 – Keeping statistics of where your users click more, it will give you an opportunity to improve the software.



Q7- How do you rate the application?



Comments:

- Good idea to show how some new tools are used.

The objective of this chapter is to describe the evaluation: the evaluation criteria and mechanisms have been detailed, essentially surveys that combine different questions, to verify if the requirements of the project and the needs of the users are satisfied with the results.

From the analysis of the surveys and the experience of the real users, it can be extracted several ideas:

- Users believe that the use of this app in the future can be useful due to its modern aspect and its intuitive interface and the functionalities are different than other apps.
- Most users really appreciate the innovation on the application and the use of technologies and tools that are not commonly used.
- Most users think that the prototype has fulfilled their expectations, but there is still a big room for improvement. Among the characteristics demanded, it can be highlighted the

necessity to bring this project to a real situation and finish all the functionalities completely. Anyway, the users think that it has been a really close approach to a real application.

As a final conclusion and from the experience that the users have experienced, it can be stated that the overall feedback was positive, the main functionalities and the application, in general, were well received by the users

Chapter 5 - Conclusions and future work

5.1. Conclusions

During the development of this project, a main objective has been pursued:

Create a prototype that approximates a real application, focused on the use of all technologies, tools, protocols, etc. that have been explained during the course of it.

This objective has been successfully achieved, but without forgetting that many other secondary ones have been obtained for this: analysis and study of the technologies used, planning, implementations, etc. Once this project is finished, we can draw several brief (subjective) conclusions regarding the technologies used:

Vue. It is a very powerful framework capable of achieving splendid results. It is true, that it is totally an adventure to use it, since it is quite new, and the documentation is barely null. It is believed that in the not too distant future it will be a reference of hybrid programming.

Firebase. Very surprising this tool, and not very well known by the community. Thanks to it, processes such as user authentication are greatly simplified, and also, having a real-time database in the cloud is a great advantage. But that's not all, Firebase has many more features, which have not been used in this master thesis, but can be very useful, such as data storage or hosting.

Analytical Google. Another very useful which is used by many companies in order to achieve better results in their businesses. Thanks to it, and without having a lot of marketing knowledge, you can get great performance to statistics and advertisements in your application.

Google Spreadsheet. It is really quite complicated to use, talking about implementation, with a fairly reprehensible security. The idea of using Google Spreadsheets as a database is ridiculous but it can be useful when you want to export data and work with it, depending on the case.

EMT Api. Not very recommended, most micro services are usually down and you have to contact their support team many times in order to get an answer. In addition, the response time is very large and much of the data is corrupt (duplicates).

Apart from the technologies used, facing the application and taking into account the opinion of the users who have tested it, it can be said that it is very viable for computer companies, to invest in programmers that develop hybrid applications, since both costs as times decrease. In turn, investing in both analytics/statistics (Google Analytics) and innovation, in order to simplify different processes typical of an application (Firebase), is something that will eventually be reflected in their numbers in a very positive way. It should not be forgotten, that this project has been carried out completely free of charge, so obtaining benefits from a project of this nature, including some improvements, would be relatively easy.

5.2. Future Work

Within this project there are several possibilities for expansion. The first way it consists in extending the scope and another way would be to add new functions and features to the solution.

The first path focuses on resolving the small problems that this application actually have, like format the answers of the EMT api to don't get duplicates, fix small details like the dropdown's performance etc. Further, it would be nice if the functionalities are rethought and they have a more useful meaning and performance.

Also, changes for example, in what format you can download the schedule, and not only have the option of Google Spreadsheets, or use Google Analytics for a wider solution, like study how to make campaigns focusing in a specific type of advertisements or users.

Once the bugs are fixed and the improvements are done, the next step would be to deploy the application for both platforms (Android and iOS) and bring it to production (generating the apks).

The second block of extensions focuses on adding new features to the solution. The most important would be to include different roles (at least two). The admin role, in order to control all the statistics and advertisements from the application. Another one would be to improve the software usability and offer a wider support to the user.

Regardless of the technologies that have been used, it could be possible to include others in order to simplify some process or, for example, to get a better security.

There is a wide range of improvement in this application, because it is just a prototype and a proof of concept to demonstrate how these technologies work and how can be used together.

Chapter 6 - Introducción

En este momento, en el planeta Tierra, hay aproximadamente 7650 millones de personas repartidas en todos los países y enmarcadas en diferentes edades.

En el último año, ha habido más de 3000 millones de usuarios de teléfonos móviles inteligentes, pero gracias a varios estudios, esta cifra aumentará en los próximos años, llegando a una cantidad cercana a los 3800 millones de usuarios, una verdadera barbaridad.

Es muy raro ver a una persona en la calle que no tenga un teléfono con conexión a internet desde el cual consultar una cantidad innumerable de servicios, por lo que desarrollar software que no se ajusta a dispositivos móviles, actualmente, es una pérdida importante de clientes.

Además, las aplicaciones están por delante de las páginas web, en casi todos los aspectos relevantes: son más fáciles de usar y su navegación es más rápida, tienen acceso más inmediato al contenido, se adaptan mejor a las necesidades del cliente y recuerdan sus preferencias de navegación, por lo que desarrollar solo una página web y no desarrollar una aplicación, es un impacto negativo para una empresa.

La tecnología ha mejorado mucho durante los últimos años (siempre lo ha hecho, y cada vez más rápido), y por lo tanto, el campo de la programación también lo ha hecho.

Actualmente hay muchas formas de programar el mismo software: usando un lenguaje de programación distinto, diferentes herramientas etc. Pero lo que está claro es que debemos realizar un estudio antes de comenzar a desarrollar, para ver qué herramientas se ajustan mejor al resultado esperado.

Un aspecto muy importante es la innovación y la creatividad, por lo que utilizar tecnologías innovadoras es apostar por resultados que marquen la diferencia con la competencia, y esto, es un punto crucial en el desarrollo de un nuevo proyecto.

Esto es lo que quiero lograr en este proyecto: estudiar una serie de tecnologías innovadoras, que no han sido utilizados por tanta gente, para tratar de lograr un resultado exitoso, plasmandolo en un pequeño software.

Las tecnologías más importantes que se han utilizado en este proyecto son: Vue (framework), herramientas de Google (Google Analytics, Firebase, Google Maps, Google Spreadsheets) y protocolos (Measurement Protocol, OAuth2).

Como se puede verificar en la bibliografía, hay muchas referencias de estas tecnologías, por separado, pero cuando todas se unen, hay problemas para encontrar referencias, principalmente porque Vue es un framework bastante nuevo.

Se ha elegido la API EMT Madrid para trabajar con sus datos con el fin de mostrar cómo estas tecnologías funcionan con datos reales y para tratar de plasmar cómo se pueden utilizar para acercarse a un ejemplo real de aplicación. La información que ofrece esta API está relacionada con las líneas de autobuses, el uso de bicicletas públicas y aparcamientos, en Madrid.

La aplicación que se ha implementado se centra en la información ofrecida en las líneas de autobús y utiliza esta para crear algunas funcionalidades, como consultar las paradas más cercanas a la ubicación del usuario, verificar cuántas personas han consultado una determinada parada o saber qué autobuses llegarán a una parada.

La solución propuesta debe entenderse como un punto de partida para avanzar en esta dirección y como prueba de concepto de cómo estas tecnologías pueden trabajar juntas.

Chapter 7 - Conclusiones y trabajo futuro

7.1. Conclusiones

Durante el desarrollo de este proyecto, se ha perseguido un objetivo principal:

Crear un prototipo que se aproxime a una aplicación real, enfocado al uso de todas las tecnologías, herramientas, protocolos etc. que se han explicado durante el transcurso del mismo.

Este objetivo se ha conseguido con éxito, pero sin olvidar que para ello se han tenido que conseguir otros muchos secundarios: análisis y estudio de las tecnologías usadas, planificaciones, implementaciones etc.

Una vez ya finalizado este proyecto, podemos sacar varias breves conclusiones (subjetivas) en cuanto a las tecnologías usadas:

Vue. Es un framework muy potente capaz de conseguir resultados espléndidos. Es cierto, que es totalmente una aventura el utilizarlo, ya que es bastante nuevo, y la documentación es apenas nula, pero viene pisando muy fuerte y se cree que en un futuro no muy lejano será un referente de la programación híbrida.

Firebase. Muy sorprendente esta herramienta, y poco conocida por una gran parte de la comunidad. Gracias a ella, procesos como el de autenticación de usuarios se simplifican mucho, y el tener una base de datos en tiempo real en la nube, es una gran ventaja. Pero eso no es todo, Firebase cuenta con muchas más funcionalidades, las cuales no se han utilizado en esta tesis, pero que pueden resultar muy útiles, como almacenamiento de datos o hosting.

Google Analytics. Otra herramienta muy útil, que es usada por bastantes empresas de cara a conseguir mejores resultados en sus negocios. Gracias a ella, y sin tener muchos conocimientos de marketing, puedes sacar un gran rendimiento a estadísticas y anuncios en tu aplicación.

Google Spreadsheets. Realmente es bastante complicada de utilizar de cara a la implementación con una seguridad bastante reprochable. La idea de utilizar Google Spreadsheets como base de

datos es totalmente ridícula pero puede resultar útil a la hora de querer exportar datos y trabajar con ellos, dependiendo del caso.

EMT Api. No muy recomendable, la mayoría de los microservicios suelen estar caídos y tienes que contactar cada 2 por 3 para que los arreglen. Además, el tiempo de respuesta es muy grande y buena parte de los datos están corruptos (duplicados).

A parte de las tecnologías empleadas, de cara a la aplicación y teniendo en cuenta la opinión de los usuarios que la han probado, se podría decir que es muy viable de cara a las empresas informáticas, invertir en programadores que desarrollen aplicaciones híbridas, ya que tanto los costes como los tiempos disminuyen. A su vez, invertir tanto en analíticas/estadísticas (Google Analytics) como en innovación, de cara a simplificar diferentes procesos típicos de una aplicación (Firebase), es algo que a la larga, se verá reflejado en sus números de manera muy positiva. No se debería de olvidar, que este proyecto se ha realizado de forma totalmente gratuita, por lo que obtener beneficios de un proyecto de estas características, incluyendo algunas mejoras, sería relativamente fácil.

7.2. Trabajo futuro

Dentro de este proyecto hay varias posibilidades de expansión. Una de ellas consistiría en ampliar el alcance, y otra forma, sería agregar nuevas funciones y características a la solución.

El primer camino se centra en resolver los pequeños problemas que esta aplicación tiene, como dar formato a las respuestas de la API de EMT para no obtener duplicados, corregir pequeños detalles como el rendimiento del menú desplegable, etc. Además, sería bueno si las funcionalidades se dieran una vuelta y encontrar un significado y un rendimiento más útiles.

Además, cambios en, por ejemplo, en qué formato se puede descargar la tabla de horarios, y no solo tener la opción de Google Spreadsheet, o usar Google Analytics para una solución más amplia, como estudiar cómo hacer campañas centradas en un tipo específico de anuncios o usuarios.

Una vez que se corrijan los errores y se realicen las mejoras, el siguiente paso sería desplegar la aplicación para ambas plataformas (Android e iOS) y llevarla a producción (generando los apks).

El segundo bloque de extensiones se enfoca en agregar nuevas características a la solución. Lo más importante sería incluir diferentes roles (al menos dos). El rol de administrador, para controlar todas las estadísticas y anuncios de la aplicación, sin la necesidad de saber cómo la consola de Google Analytics funciona, y el rol de usuario.

Otra opción, sería mejorar el usabilidad del software y ofrecer un soporte más amplio para el usuario.

Independientemente de las tecnologías que se hayan utilizado, podría ser posible incluir otras para simplificar algún proceso o, por ejemplo, para obtener una mayor seguridad.

Hay una amplia variedad de mejoras en esta aplicación, ya que es solo un prototipo y una prueba de concepto para demostrar cómo funcionan estas tecnologías y cómo se pueden usar juntas.

Bibliography

- [1] Serrano, Nicolas, Josune Hernantes, and Gorka Gallardo. "Mobile web apps." *IEEE software* 30.5 (2013): 22-27.
- [2] Jobe, W. (2013). Native Apps vs. Mobile Web Apps. *International Journal of Interactive Mobile Technologies*, 7(4).
- [3] SKÓLSKI, Paweł. "Single-page application vs. multiple-page application." *Luettu* 2 (2016): 2018.
- [4] Wang, Q. S., & Cao, L. (2015). *U.S. Patent Application No. 14/032,184*.
- [5] Satrom, B. (2018). Choosing the right JavaScript framework for your next web application. *Vitbok RITM0012054*. Progress Software Corporation.
- [6] Kyriakidis, Alex, and Kostas Maniatis. *The Majesty of Vue. js*. Packt Publishing Ltd, 2016.
- [7] Nelson, Brett. "Getting to Know Vue. js." (2018).
- [8] Wohlgethan, E. (2018). *Supporting Web Development Decisions by Comparing Three Major JavaScript Frameworks: Angular, React and Vue. js* (Doctoral dissertation, Hochschule für Angewandte Wissenschaften Hamburg).
- [9] Hong, Philippe. *Practical Web Design: Learn the fundamentals of web design with HTML5, CSS3, Bootstrap, jQuery, and Vue. js*. Packt Publishing Ltd, 2018.
- [10] Moroney, Laurence, Moroney, and Anglin. *Definitive Guide to Firebase*. Apress, 2017.
- [11] Rochelle, Jonathan, et al. "Collaborative online spreadsheet application." U.S. Patent No. 8,307,119. 6 Nov. 2012.
- [12] Clark, D. J., Nicholas, D., & Jamali, H. R. (2014). Evaluating information seeking and use in the changing virtual world: the emerging role of Google Analytics. *Learned publishing*, 27(3), 185-194.
- [13] Ledford, Jerri L., Joe Teixeira, and Mary E. Tyler. *Google analytics*. John Wiley and Sons, 2011.

- [14] Svennerberg, Gabriel. *Beginning Google Maps API 3*. Apress, 2010.
- [15] Sucasas, V., Mantas, G., Radwan, A., & Rodriguez, J. (2016, May). An OAuth2-based protocol with strong user privacy preservation for smart city mobile e-Health apps. In *2016 IEEE International Conference on Communications (ICC)* (pp. 1-6). IEEE.
- [16] Zanella, A., Bui, N., Castellani, A., Vangelista, L., & Zorzi, M. (2014). Internet of things for smart cities. *IEEE Internet of Things journal*, 1(1), 22-32.
- [17] Cockburn, A. (2000). *Writing effective use cases*. Addison-Wesley Professional.
- [18] Hinojosa, M. A. (2003). Diagrama de Gantt. *Producción, procesos y operaciones*.
- [19] Medium dedicated to comparison between *Single Page Application* and *Multiple Page Application*. (<https://medium.com/@NeotericEU/single-page-application-vs-multiple-page-application-2591588efe58>)
- [20] Official Website of *Vue.js*. (<https://vuejs.org/>)
- [21] Official Website of *Firebase* and *Google Analytics*. (<https://firebase.google.com/?hl=es-419> , <https://analytics.google.com/analytics/web/>)
- [22] *Google Analytics Measurement Protocol* explained in great detail - *Optimizes Smart*.
- [22] *EMT Madrid MobilityLabs* Website. (<https://www.emtmadrid.es/Home>)

Appendix A - Set up Vue

To use NativeScript-Vue is necessary to set up the system to compile the apps.

System Characteristics

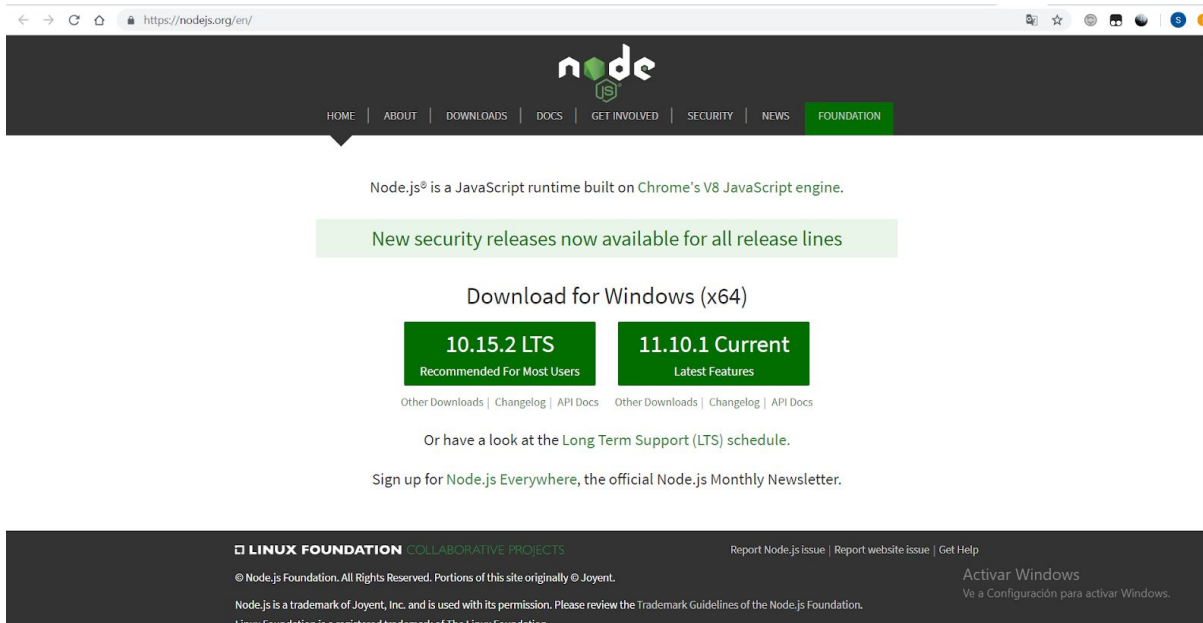
It has been used a laptop with the following characteristics:

- OS: Windows 10 Home
- Processor: Intel(R) Core(TM) i7-8750H CPU @ 2.20 Ghz
- System's type: OS-64 bits, processor-x64
- RAM memory: 16 GB

Node

Node.js is an open-source, cross-platform Javascript run-time environment that executes Javascript code outside of a browser.

Download and install the latest version of Node.js from <https://nodejs.org/> . The version used is the '10.15.2 LTS'.



Verify if the installation was successful by opening the terminal console and running ‘node --version’.

NativeScript CLI

NativeScript is an open-source framework to develop apps on the Apple iOS and Android platforms.

Open the terminal and run: ‘npm install -g nativescript’. Verify if the installation was successful by opening the terminal console and running ‘tns’.

Visual Studio Code

It is necessary an editor/environment to programming. It has been used Visual Studio Core but it can be used others like Eclipse.

Visual Studio Code is a source code editor developed by Microsoft that includes support for debugging, syntax highlighting, snippets etc..

It is possible to download it via the following link: <https://code.visualstudio.com/>

Ap.A.1. Building an app locally

Let’s try to build our first app locally, the typical example of ‘Hello world’.

First of all, it is necessary to create a folder/project where we are going to save all the documents, plugins etc..

The easiest way is to open the terminal console and type the following codes:

- npm install -g @vue/cli @vue/cli-init
- vue init nativescript-vue/vue-cli-template ProjectName

The first command installs the latest Vue CLI and support for Vue CLI 2.x templates through the @vue/cli-init add-on.

```
cn Símbolo del sistema
Microsoft Windows [Versión 10.0.17763.316]
(c) 2018 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Sergio>npm install -g @vue/cli @vue/cli-init
```

The second command creates a project using the vue-cli-template.

```
cn Símbolo del sistema
Microsoft Windows [Versión 10.0.17763.316]
(c) 2018 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Sergio>vue init nativescript-vue/vue-cli-template 'thesis'
```

In the ‘creation process’, one has to fill some fields, like the project name, project description etc...

```
cn Símbolo del sistema - vue init nativescript-vue/vue-cli-template 'thesis'
Microsoft Windows [Versión 10.0.17763.316]
(c) 2018 Microsoft Corporation. Todos los derechos reservados.

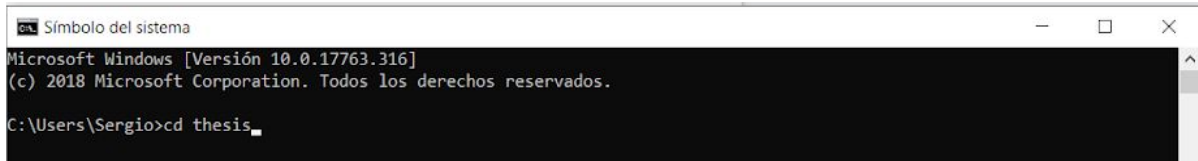
C:\Users\Sergio>vue init nativescript-vue/vue-cli-template 'thesis'

"git" no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.
? Project name thesis
? Project description A native application built with NativeScript-Vue
? Application name NativeScript-Vue Application
? Unique application identifier org.nativescript.application
? Project version 1.0.0
? Author
```

The next step is open the file .vue with an editor that is in the folder ‘app/components’ inside the folder created with the last command. There is a simple code in this file, with the typical example of ‘Hello World’.



In the terminal console, it is necessary to change the folder to the ones that we created in the last step (projectName). To do it, use the command ‘cd’.



```
Símbolo del sistema
Microsoft Windows [Versión 10.0.17763.316]
(c) 2018 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Sergio>cd thesis_
```

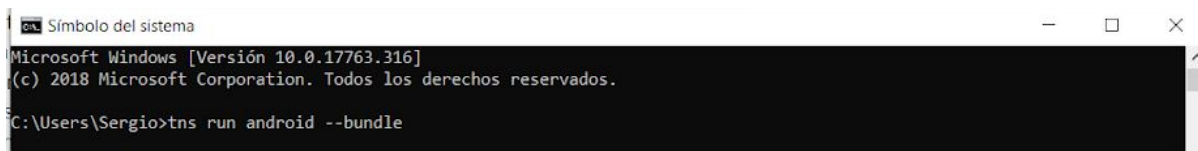
Let's introduce the next command: 'tns run android --bundle'

With this command, the app will execute for the platform android, if you change 'android' for 'ios' the app will execute for the platform ios.

There are many options to execute the app. Once you introduced the command, you can choose between some possibilities, the ones that I prefer are the next:

1- Sync with playground: in this one, one QR Code will appear in the console terminal, and you can download the app 'Native PlayGround' in your mobile phone and scan the QR Code, if all goes good, you will can pre visualize the result of the app in your mobile phone.

2- Configure for local builds: with this option, the computer configure your system to be ready to launch the executions of the app without doing nothing more.



```
Símbolo del sistema
Microsoft Windows [Versión 10.0.17763.316]
(c) 2018 Microsoft Corporation. Todos los derechos reservados.

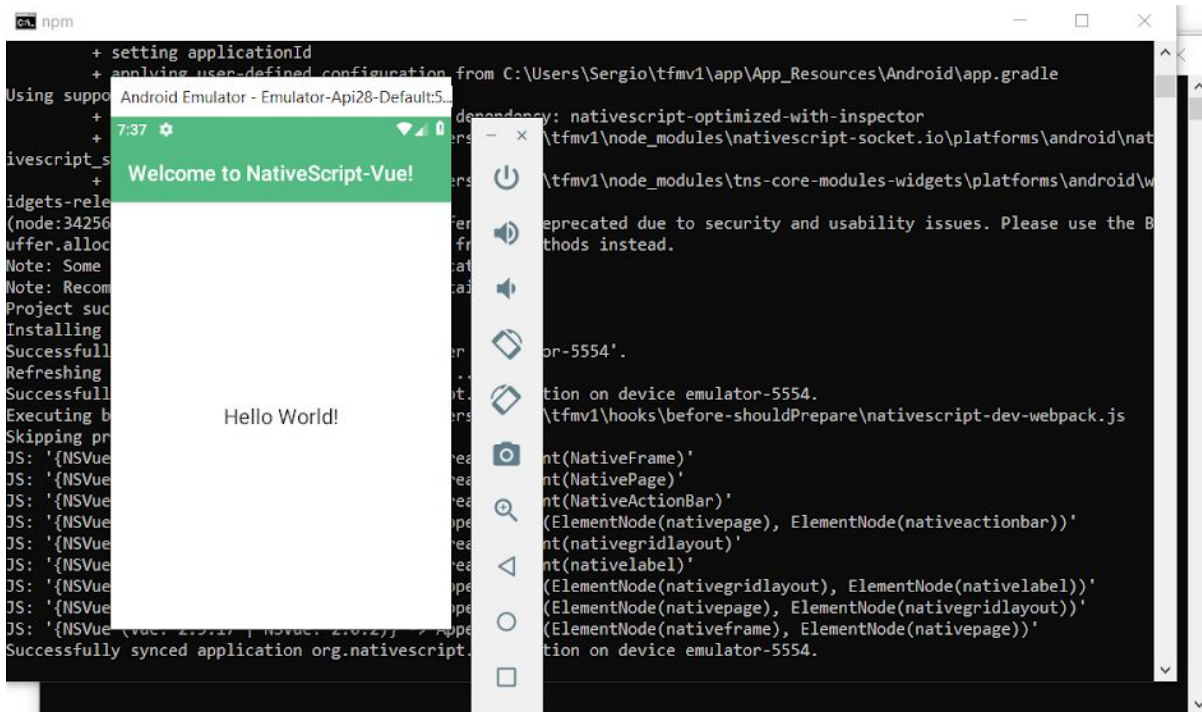
C:\Users\Sergio>tns run android --bundle
```

I used the second option, even the first one is easier. It takes such a long time configuring your own computer, but you just have to press 'Y' every time the installation requires you to do it. Also, it gives you the possibility to install an 'Android emulator', which is a really good thing. After finishing the installation, it could be possible that you have to reboot the computer, and re type the command.

It is required 'Gradle'. Gradle is an open-source build automation tool that is designed to be flexible enough to build almost any type of software. It is a general-purpose build tool and the core model is based on tasks.

It is necessary to build the android APK, so the first time we build the app, gradle will be installed automatically.

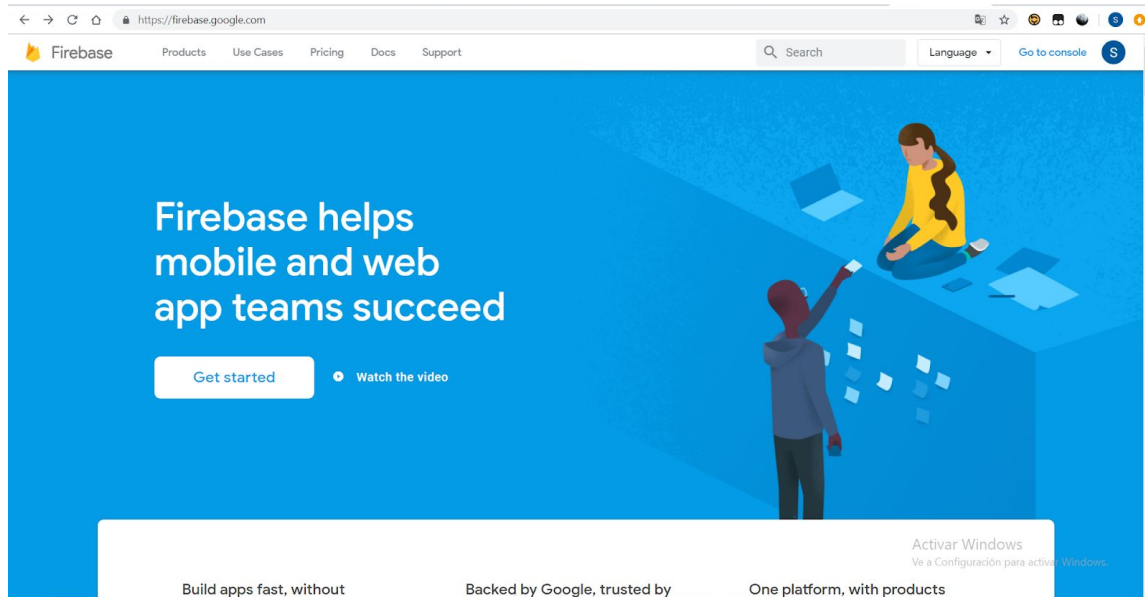
After all this configurations and requirements, this has to be the result:



Possible problem: it is said the module 'update-ns-webpack' usually gives problems, so if the building/execution did not work and throws an exception like this one: 'System.err: TypeError: Cannot read property 'create' of undefined', deleting all the files related to this module should solve the problem. (The modules are always in the folder 'node_modules'.

Appendix B - Set up Firebase

First of all, to configure Firebase we have to access to the following url:
firebase.google.com



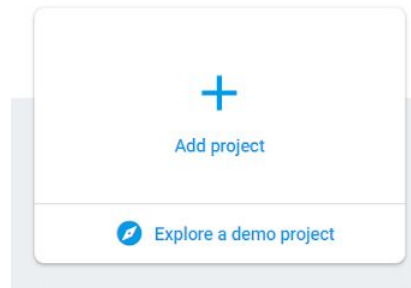
It is possible use the standard Google account to start using this tool. Clicking get started, it is possible to add a new project.

Welcome to Firebase!

Tools from Google for developing great apps,
your users and earning more through mobile

[Learn more](#) [Documentation](#) [Support](#)

Recent projects



It is necessary to set a name to the project, the Analysis location and check both conditions at the bottom.

Add a project [X]

Project name: TFMv1 [Tip: Projects span apps across platforms]

Project ID: tfmv1-a87d5 [edit]

Analytics location: Poland

Cloud Firestore location: eur3 (europe-west)

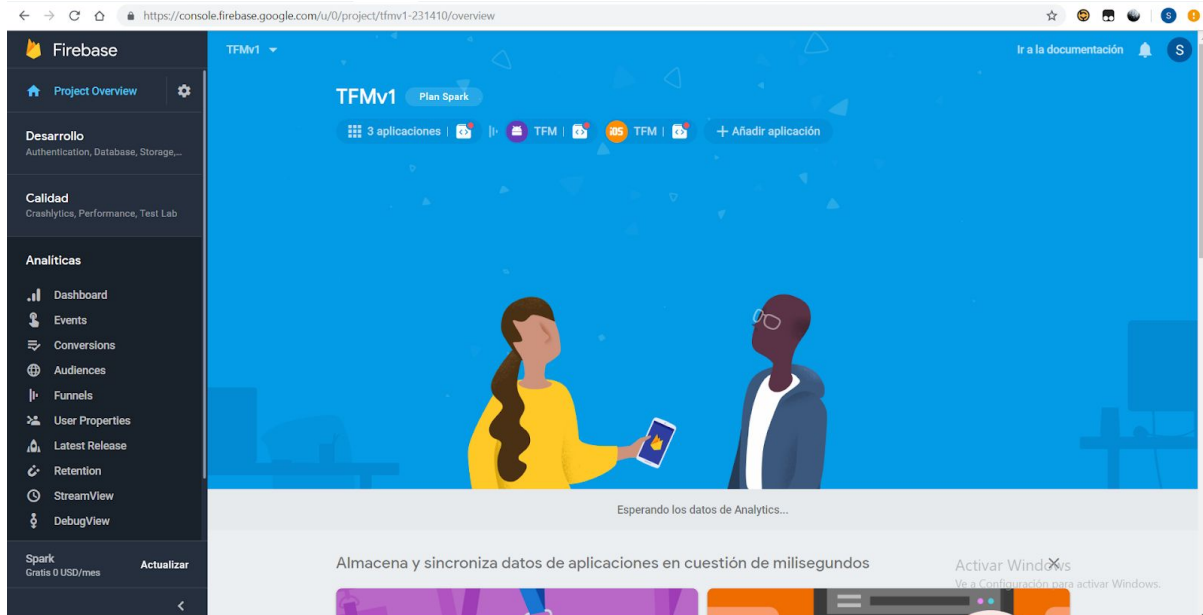
Use the default settings for sharing Google Analytics for Firebase data

- Share your Analytics data with all Firebase features
- Share your Analytics data with Google to improve Google Products and Services
- Share your Analytics data with Google to enable technical support
- Share your Analytics data with Google to enable Benchmarking
- Share your Analytics data with Google Account Specialists

I accept the [controller-controller terms](#). This is required when sharing Analytics data to improve Google Products and Services. [Learn more](#)

Cancel **Create project**

Clicking create project, we have already configured our project and this should be our main screen:



Ap.B.1. Firebase as user's authentication

One of the good products that Firebase offers is the user's authentication. In this chapter, we are going to learn how to authenticate users using Firebase. This is needed for example, in the sign up and sign in process that are used on most of the websites.

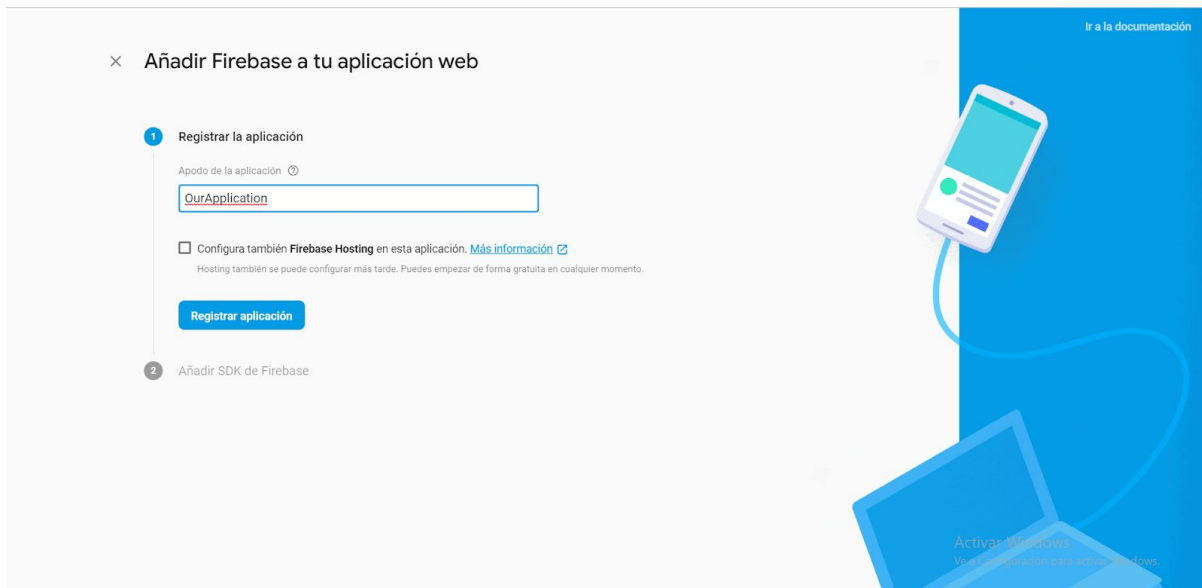
The first step is to create an application:



Clicking this button, we have to decide for what platform we want to create the application. In our case, we select 'web platform' (third button):



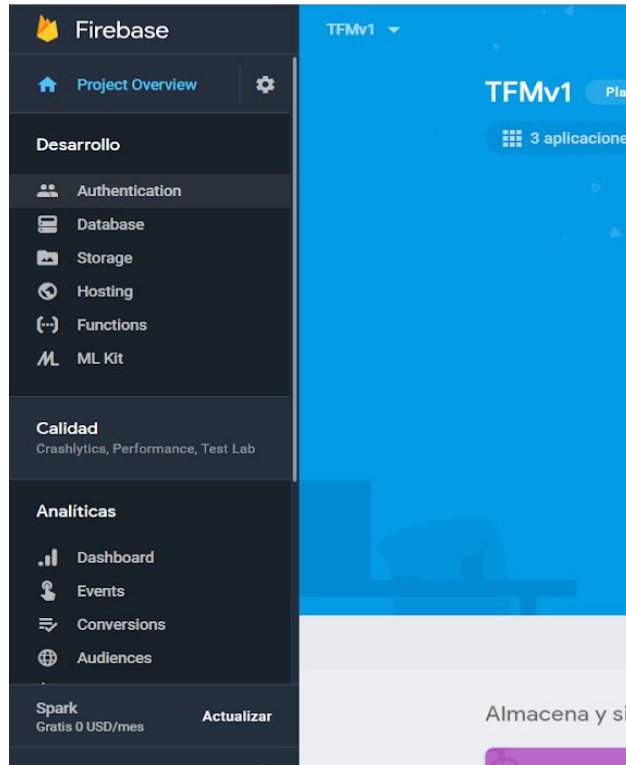
In the next screen, we just have to write a name for our app and register the application.



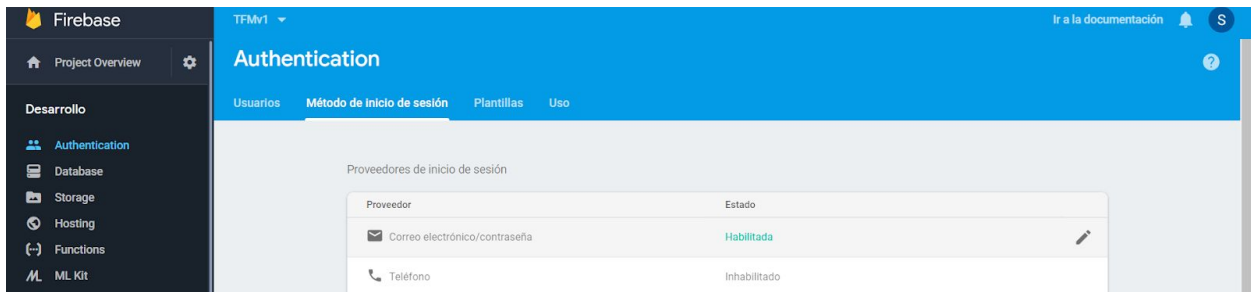
Once we have done it, we have to copy the code that appears in the next screen, because we will have to use it in our code to connect with Firebase:



Now it is necessary to select the authentication's method. Being in the Firebase's main screen, we have to click in Authentication (located in development).



Then, we select 'Login method' at the top. Our choose is to authenticate the users with their email and password, so we have to enable that method.



Let's switch to implement this in our code. As usual, we have to install one library to use firebase with Vue: 'npm install --save firebase'.

In our main.js, we are going to initialize Firebase. To do this, we have to copy the code we mentioned before. Don't forget to import the library.

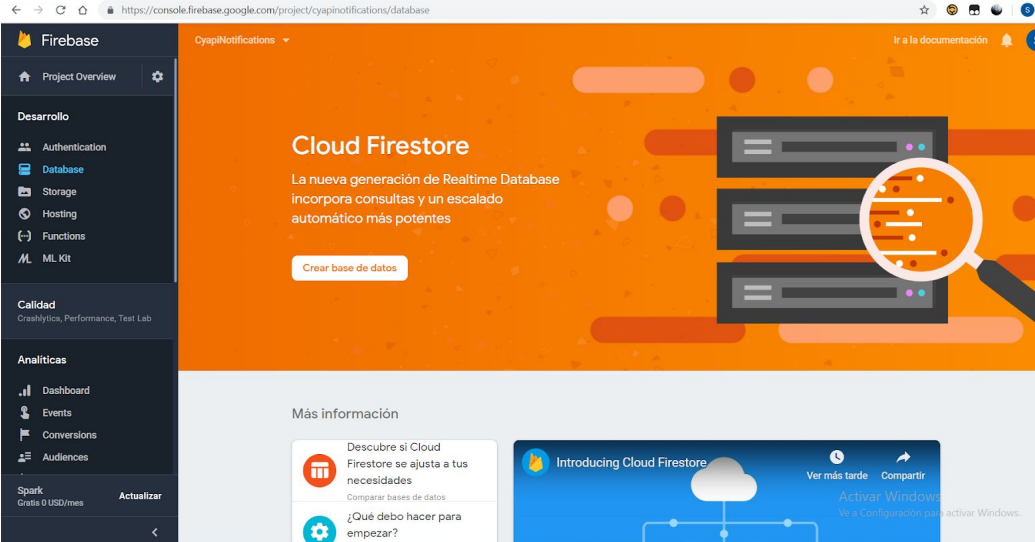
```
import firebase from 'firebase'
import 'firebase/firestore'
```

```
29 // Your web app's Firebase configuration
30 var firebaseConfig = {
31   apiKey: 'AIzaSyAdy4Tio9rLLY7UpvCe_G_vPZm7tYByl8',
32   authDomain: 'tfmv1-231410.firebaseio.com',
33   databaseURL: 'https://tfmv1-231410.firebaseio.com',
34   projectId: 'tfmv1-231410',
35   storageBucket: 'tfmv1-231410.appspot.com',
36   messagingSenderId: '354521227503',
37   appId: '1:354521227503:web:29ee727c11b5a099'
38 }
39 // Initialize Firebase
40 firebase.initializeApp(firebaseConfig)
```

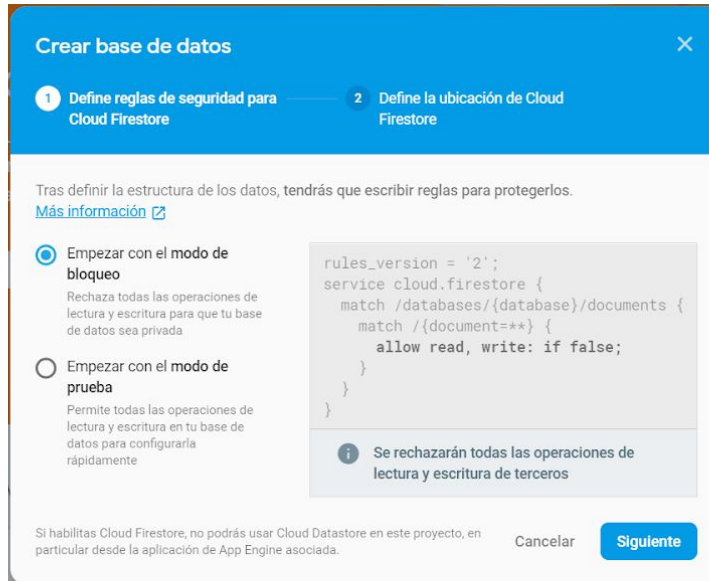
Ap.B.2. Firebase as real-time database

Another product that Firebase offers is the real-time database. In this chapter, we are going to learn how to create a real-time database using Firebase.

Once we have created the account, we click in the option ‘Database’ and later, in ‘Create Database’:



In the next screen, we select the ‘locked mode’ because we do not want that everyone can edit our database:



Finally, we select the location of our database, in what part of the world (server) we want to save our data:



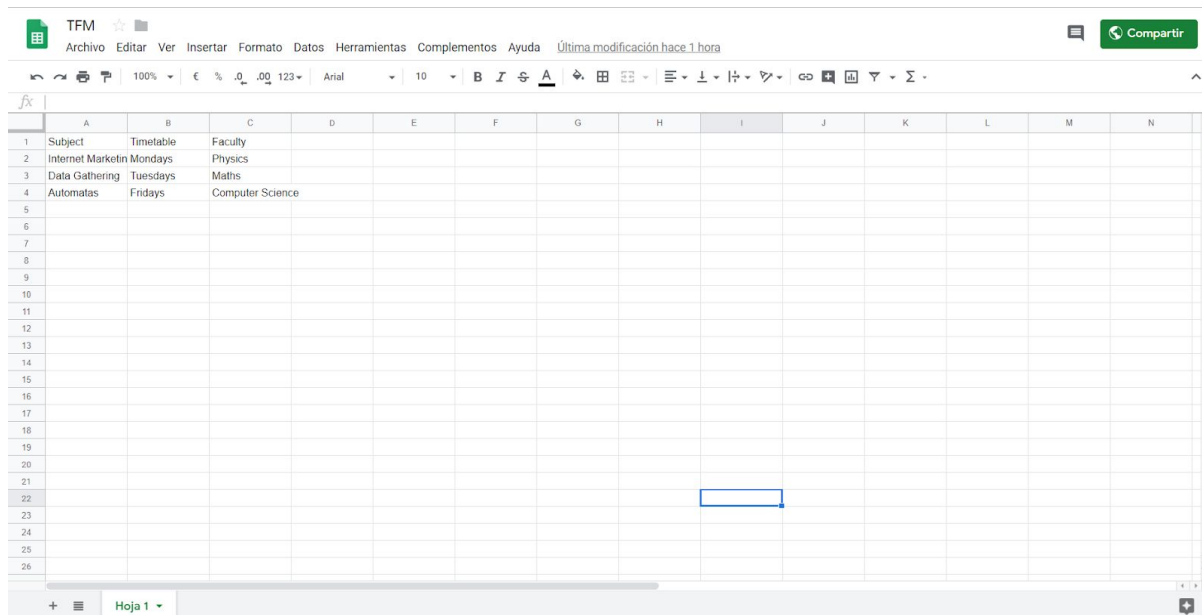
Our database is already created and configured to be used.

Appendix C - Set up Google Spreadsheet + OAuth2

Ap.C.1. Google Spreadsheet without authentication

It is not possible to create, edit or delete a spreadsheet without being authenticated on Google. In this example, we just read one public spreadsheet.

First of all, it is necessary to create it (or find one on internet), in my case this is a simple spreadsheet:



The screenshot shows a Google Spreadsheet interface. The title bar reads 'TFM' and the menu bar includes 'Archivo', 'Editar', 'Ver', 'Insertar', 'Formato', 'Datos', 'Herramientas', 'Complementos', and 'Ayuda'. The spreadsheet grid contains the following data:

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N |
|----|-------------------|-----------|------------------|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Subject | Timetable | Faculty | | | | | | | | | | | |
| 2 | Internet Marketin | Mondays | Physics | | | | | | | | | | | |
| 3 | Data Gathering | Tuesdays | Maths | | | | | | | | | | | |
| 4 | Automatas | Frídays | Computer Science | | | | | | | | | | | |
| 5 | | | | | | | | | | | | | | |
| 6 | | | | | | | | | | | | | | |
| 7 | | | | | | | | | | | | | | |
| 8 | | | | | | | | | | | | | | |
| 9 | | | | | | | | | | | | | | |
| 10 | | | | | | | | | | | | | | |
| 11 | | | | | | | | | | | | | | |
| 12 | | | | | | | | | | | | | | |
| 13 | | | | | | | | | | | | | | |
| 14 | | | | | | | | | | | | | | |
| 15 | | | | | | | | | | | | | | |
| 16 | | | | | | | | | | | | | | |
| 17 | | | | | | | | | | | | | | |
| 18 | | | | | | | | | | | | | | |
| 19 | | | | | | | | | | | | | | |
| 20 | | | | | | | | | | | | | | |
| 21 | | | | | | | | | | | | | | |
| 22 | | | | | | | | | | | | | | |
| 23 | | | | | | | | | | | | | | |
| 24 | | | | | | | | | | | | | | |
| 25 | | | | | | | | | | | | | | |
| 26 | | | | | | | | | | | | | | |

The ID of our spreadsheet is necessary, so we have to get it via the url:



I'm attaching in the same folder the code we are going to use:


```
File Edit Selection View Go Debug Terminal Help AppVue - Visual Studio Code
AppVue x
1 <template>
2 <Page class="page">
3 <ActionBar class="action-bar" title="Spreadsheet"></ActionBar>
4 <GridLayout>
5 <ListView for="entry in entries" class="list-group">
6 <template>
7 <StackLayout class="list-group-item">
8 <Label :text="entry.title.$*" />
9 </StackLayout>
10 </template>
11 </ListView>
12 </GridLayout>
13 </Page>
14 </template>
15
16 <script>
17 import * as http from "http";
18 export default {
19   data() {
20     return {
21       entries: []
22     };
23   },
24   mounted() {
25     http.getJSON("https://spreadsheets.google.com/feeds/list/1r5hLrN1jwQbJLkWRQ3D2bIR9QK2tk3v8NvSy3xx2xxY/1/public/values?alt=json").then(result => {
26       this.entries = result.feed.entry;
27     }, error => {
28       console.log(error);
29     });
30   }
31 };
32 </script>
33
34 <style scoped></style>
```

This is a simple code in which we are going to show the columns's titles of our spreadsheet.

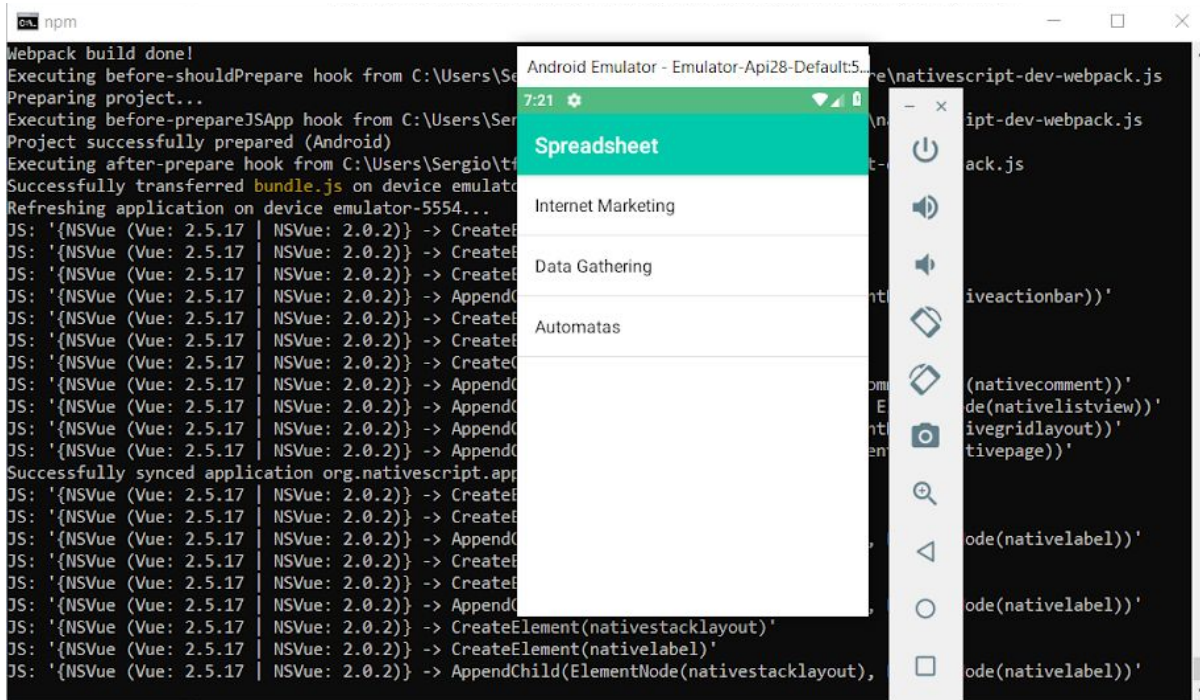
In the script section, there is a call to the spreadsheets API, in which we put the spreadsheet's ID we got before (after 'list' and before the '1').

This is the answer that we receive:

```
← → ↻ 📄 https://spreadsheets.google.com/feeds/list/1r5hLrN1jwQbJLkWRQ3D2bIR9QK2tk3v8NvSy3xx2xxY/1/public/values?alt=json
{"version": "1.0", "encoding": "UTF-8", "feed": [{"xmlns": "http://www.w3.org/2005/Atom", "xmlns:opensearch": "http://a9.com/~spec/opensearchrss/1.0/", "xmlns:gssx": "http://schemas.google.com/spreadsheets/2006/extended", "id": {"$t": "https://spreadsheets.google.com/feeds/list/1r5hLrN1jwQbJLkWRQ3D2bIR9QK2tk3v8NvSy3xx2xxY/1/public/values"}, "updated": {"$t": "2019-02-01T17:42:04.273Z"}, "category": [{"scheme": "http://schemas.google.com/spreadsheets/2006", "term": "http://schemas.google.com/spreadsheets/2006#list"}], [{"rel": "alternate", "type": "application/atom+xml", "href": "https://docs.google.com/spreadsheets/d/1r5hLrN1jwQbJLkWRQ3D2bIR9QK2tk3v8NvSy3xx2xxY/pubhtml"}, {"rel": "http://schemas.google.com/g/2005/feed", "type": "application/atom+xml", "href": "https://spreadsheets.google.com/feeds/list/1r5hLrN1jwQbJLkWRQ3D2bIR9QK2tk3v8NvSy3xx2xxY/1/public/values"}, {"rel": "http://schemas.google.com/g/2005/post", "type": "application/atom+xml", "href": "https://spreadsheets.google.com/feeds/list/1r5hLrN1jwQbJLkWRQ3D2bIR9QK2tk3v8NvSy3xx2xxY/1/public/values"}, {"rel": "self", "type": "application/atom+xml", "href": "https://spreadsheets.google.com/feeds/list/1r5hLrN1jwQbJLkWRQ3D2bIR9QK2tk3v8NvSy3xx2xxY/1/public/values?alt=json"}, {"name": "mrsergiogf", "email": {"$t": "mrsergiogf@gmail.com"}, "openSearch$totalResults": {"$t": "3"}, "openSearch$startIndex": {"$t": "1"}, "entry": [{"id": {"$t": "https://spreadsheets.google.com/feeds/list/1r5hLrN1jwQbJLkWRQ3D2bIR9QK2tk3v8NvSy3xx2xxY/1/public/values/cokur"}, "updated": {"$t": "2019-02-01T17:42:04.273Z"}, "category": [{"scheme": "http://schemas.google.com/spreadsheets/2006", "term": "http://schemas.google.com/spreadsheets/2006#list"}], "title": {"type": "text", "$t": "Internet Marketing"}, "content": {"type": "text", "$t": "timetable: Mondays, faculty: Physics"}, "link": [{"rel": "self", "type": "application/atom+xml", "href": "https://spreadsheets.google.com/feeds/list/1r5hLrN1jwQbJLkWRQ3D2bIR9QK2tk3v8NvSy3xx2xxY/1/public/values/cokur"}, {"gssx$subject": {"$t": "Internet Marketing"}, "gssx$timetable": {"$t": "Mondays"}, "gssx$faculty": {"$t": "Physics"}], {"id": {"$t": "https://spreadsheets.google.com/feeds/list/1r5hLrN1jwQbJLkWRQ3D2bIR9QK2tk3v8NvSy3xx2xxY/1/public/values/cpz4"}, "updated": {"$t": "2019-02-01T17:42:04.273Z"}, "category": [{"scheme": "http://schemas.google.com/spreadsheets/2006", "term": "http://schemas.google.com/spreadsheets/2006#list"}], "title": {"type": "text", "$t": "Data Gathering"}, "content": {"type": "text", "$t": "timetable: Tuesdays, faculty: Maths"}, "link": [{"rel": "self", "type": "application/atom+xml", "href": "https://spreadsheets.google.com/feeds/list/1r5hLrN1jwQbJLkWRQ3D2bIR9QK2tk3v8NvSy3xx2xxY/1/public/values/cpz4"}, {"gssx$subject": {"$t": "Data Gathering"}, "gssx$timetable": {"$t": "Tuesdays"}, "gssx$faculty": {"$t": "Maths"}], {"id": {"$t": "https://spreadsheets.google.com/feeds/list/1r5hLrN1jwQbJLkWRQ3D2bIR9QK2tk3v8NvSy3xx2xxY/1/public/values/cre11"}, "updated": {"$t": "2019-02-01T17:42:04.273Z"}, "category": [{"scheme": "http://schemas.google.com/spreadsheets/2006", "term": "http://schemas.google.com/spreadsheets/2006#list"}], "title": {"type": "text", "$t": "Automatas"}, "content": {"type": "text", "$t": "timetable: Fridays, faculty: Computer Science"}, "link": [{"rel": "self", "type": "application/atom+xml", "href": "https://spreadsheets.google.com/feeds/list/1r5hLrN1jwQbJLkWRQ3D2bIR9QK2tk3v8NvSy3xx2xxY/1/public/values/cre11"}, {"gssx$subject": {"$t": "Automatas"}, "gssx$timetable": {"$t": "Fridays"}, "gssx$faculty": {"$t": "Computer Science"}]}]}]}
```

That is why we are accessing to 'result.feed.entry' to get the data we need. Finally, in the template section, we go through a loop to get all the possibles titles and we show them.

This has to be the final result in our app:



Ap.C.2. Google Spreadsheet with OAuth2

Unlike the previous, in this case we need the authentication to work with one spreadsheet (create, edit, delete and also, read).

Google gives us an access to lots of welfare with the Google API. The API enable to use Google services from our own applications.

First of all, it is needed to have an account in Google, because we would like to use the Google's APIs and the Google Developer Console.

The process is divided in 3 steps:

1. Create the project and the credentials

2. Get authorization tokens
3. Enable the APIs

Before we start with all the process, we are going to introduce the RESTful APIs because we are going to use them, and also Postman, to check these requests.

Restful APIs

An API is an application programming interface. It is a set of rules that allow programs to talk to each other. The developer creates the API on the server and allows the client to talk to it.

REST stands for Representational State Transfer. It is a software architecture style that relies on a stateless communications protocol, most commonly, HTTP.

REST structures data in XML, YAML, or any other format that is machine-readable, but usually JSON is most widely used. [1]

The basic REST HTTP requests are: POST, GET, PUT, and DELETE. [2]

| | |
|--------|---------------------------------|
| GET | Read or retrieve data |
| POST | Add new data |
| PUT | Update data that already exists |
| DELETE | Remove data |

Postman

Postman is an API client tool which helps to test the API's. It enables us to test the same request against different environments with environment specific variables. [7]

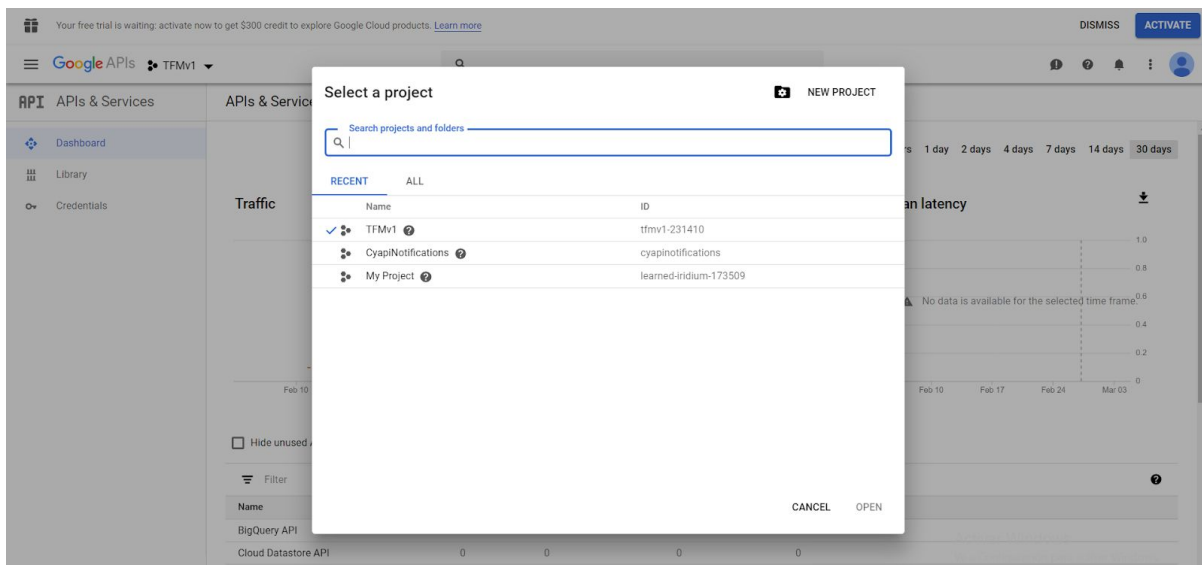
It is possible to download Postman from this url: <https://www.getpostman.com/>

Ap.C.2.1. Create the project and the credentials

Via the following url, it is possible to access to the Google Developer Console:

<https://console.developers.google.com>

The following screen dump represents the initial screen that appears on the Google Developer Console:



It is necessary to create a new project, so clicking on new project, you have to put a project name and a location.

Your free trial is waiting. activate now to get \$300 credit to explore Google Cloud products. [Learn more](#)

Google APIs

New Project

Warning: You have 15 projects remaining in your quota. Request an increase or delete projects. [Learn more](#)
[MANAGE QUOTAS](#)

Project Name *
My Project 65637

Project ID: horizontal-tube-233618. It cannot be changed later. [EDIT](#)

Location *
No organization [BROWSE](#)

Parent organization or folder

[CREATE](#) [CANCEL](#)

The next step is click on ‘credentials’ and create a ‘OAuth 2.0’ authentication.

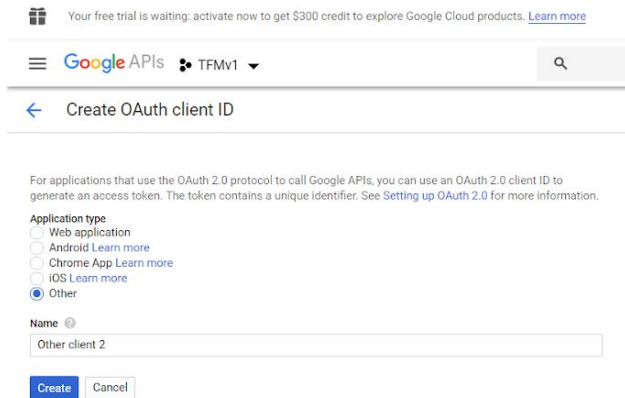
APIs Credentials

You need credentials to access APIs. [Enable the APIs you plan to use](#) and then create the credentials they require. Depending on the API, you need an API key, a service account, or an OAuth 2.0 client ID. For more information, see the [authentication documentation](#).

[Create credentials](#)

- API key**
Identifies your project using a simple API key to check quota and access
- OAuth client ID**
Requests user consent so your app can access the user's data
- Service account key**
Enables server-to-server, app-level authentication using robot accounts
- Help me choose**
Asks a few questions to help you decide which type of credential to use

In this case, we have to select ‘Other’ in application type because it is a hybrid application.



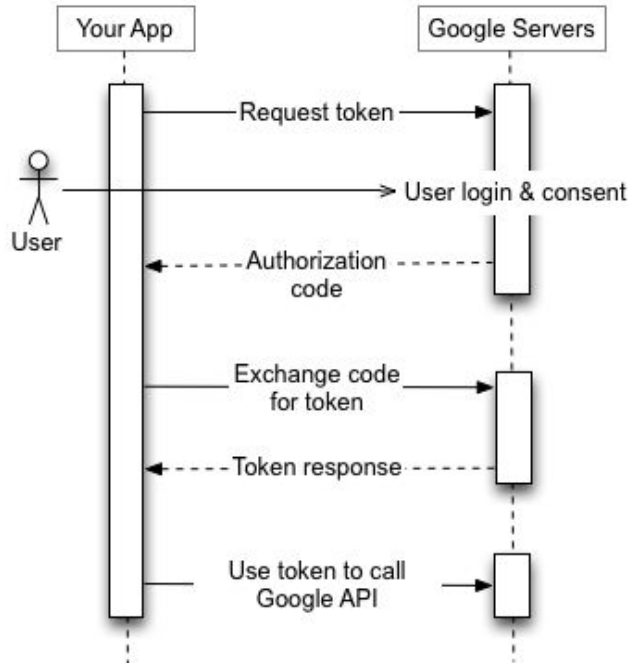
Here is the result:



There is a client ID, and also, it is possible to download the authentication in .JSON format.

Ap.C.2.2. Get authorization tokens

There are 2 steps to get authorization tokens for the project. First is a request for an authorization code and then there is the code confirmation. After the process you will have an `access_token` and a `refresh_token`. You need both of them if your app is offline app, in other case `access_token` is enough. `Refresh_token` is useful to renew `access_token` after it expires.



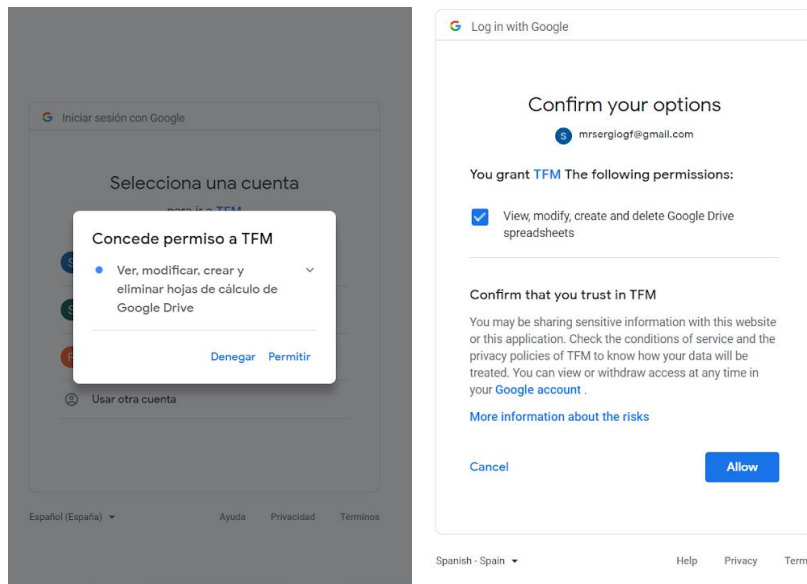
To request permissions for the app, we have to use the browser, with the next url followed by some GET parameters: <https://accounts.google.com/o/oauth2/v2/auth>

1. client_id: previously acquired
2. response_type: put here code, it is the simplest way to authenticate by requesting the code.
3. scope: each api has its own scope. In this case, we will use: spreadsheets
4. redirect_uri: <http://localhost>
5. access_type: use it with value offline, only if it is needed refresh_token.

Our request URI should look like that:

https://accounts.google.com/o/oauth2/v2/auth?client_id=354521227503-olf2d47914rt30lf74qb0oledebu03eb.apps.googleusercontent.com&response_type=code&scope=https://www.googleapis.com/auth/spreadsheets&redirect_uri=http://localhost:8080&access_type=offline

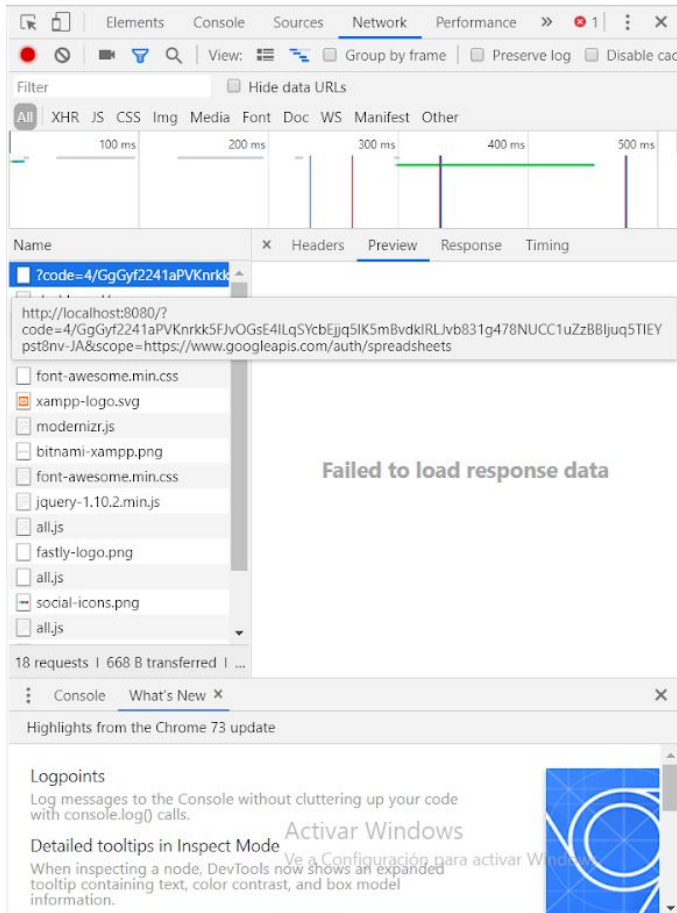
After the call, browser shows question about granting permissions to the app:



Redirect URI looks as follows:

<http://localhost/?code=4/AACbpkMFarNdMwz1qVPV0mWcnfjSt0zMcNcUogSMgr2lcZU2G7qjf7B-f1lmTkhRpfqXFBwxzd9ad3vRD1Oymgk#>

Sometimes the URI does not look like that, so it is necessary to open the developer tools interface. Pressing 'F12' it is possible to open it. Clicking in 'Network' the URI appears:



We can extract code from this. It is:

4/GgGyf2241aPVKnrk5FJvOGsE4ILqSYcbEjjq5IK5mBvdklRLJvb831g478NUCC1uZzBBIjuq5TIEYpst8nv-JA

The next step is to post code with some other parameters on: <https://www.googleapis.com/oauth2/v4/token>

1. code: previously acquired
2. client_id: previously acquired
3. client_secret: previously acquired
4. grant_type: use authorization_code
5. redirect_uri: use <http://localhost>

Now, we are going to use Postman to do the POST. The POST request should be similar to the next one:

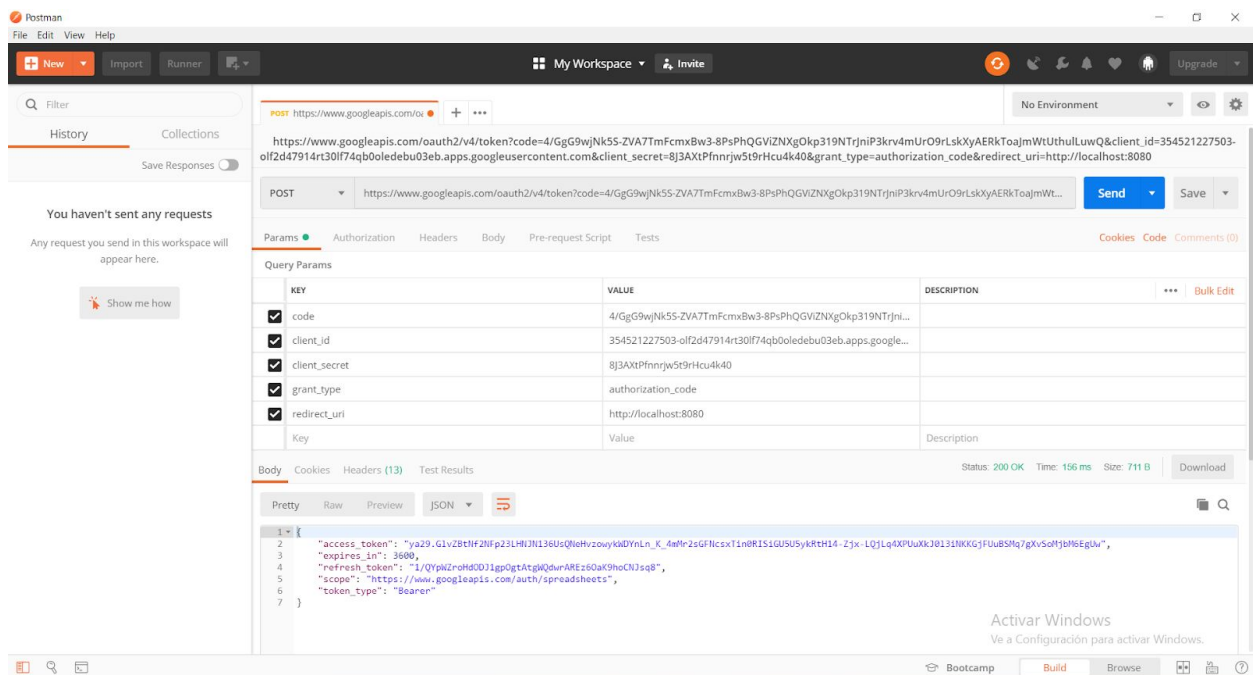
Headers:

Content-Type: application/x-www-form-urlencoded

Body:

code=4/AACbpbkMFarNdMwz1qVPV0mWcnfjSt0zMcNcUogSMgr2lcZU2G7qjf7B-f1lmTkhRpfgXFBwxzd9ad3vRD1Oymgk#&client_id=187637922392-nm8r2q89o9gub1ftmuos32coutiumkt1.apps.googleusercontent.com&client_secret=aspzni4WpLp3pv_ixszax_pQ&grant_type=authorization_code&redirect_uri=<http://localhost:8080>

The answer gives us access_token and refresh_token:

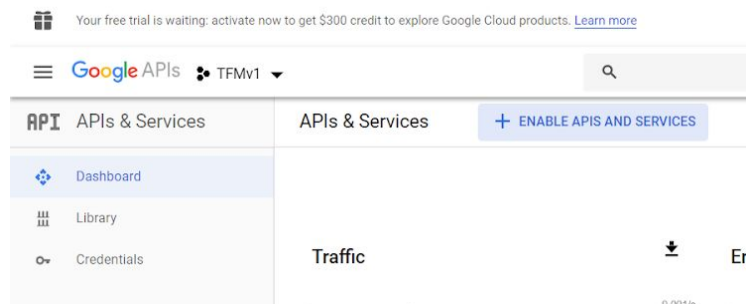


```
1 {
2   "access_token": "ya29.GlvZBtNF2NFp23LHjN136UsQNeHvzowykDYNLn_K_4mMr2sGFncsXTin0RISiGU5U5ykRtH14-Zjx-LQjLq4XPUuXk1013iNKKGjFUuBSMq7gXvScMjbM6EgUw",
3   "expires_in": 3600,
4   "refresh_token": "1/QypwZroHdODJ1gpOgtAtgWQdwrAREz60aK9hoCNjsq8",
5   "scope": "https://www.googleapis.com/auth/spreadsheets",
6   "token_type": "Bearer"
7 }
```

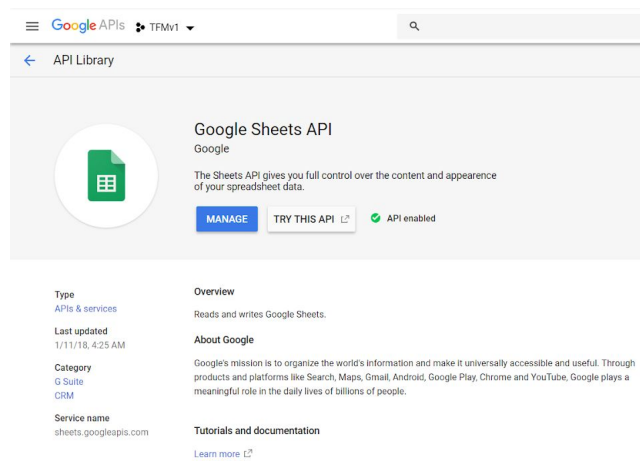
Activar Windows
Ve a Configuración para activar Windows.

Ap.C.2.3. Enable the APIs

The next step is enable the APIS & Services we would like to use. First, clicking on ‘Dashboard’ and then on ‘Enable APIS and Services’.



Typing ‘Google Sheets’, the API that we want to use appears. Clicking on it, there is one button at the top that says ‘Enable API’. Once clicked, the API is ready to be used.



Appendix D - Set up Google Analytics

Ap.D.1. Tracking with Library

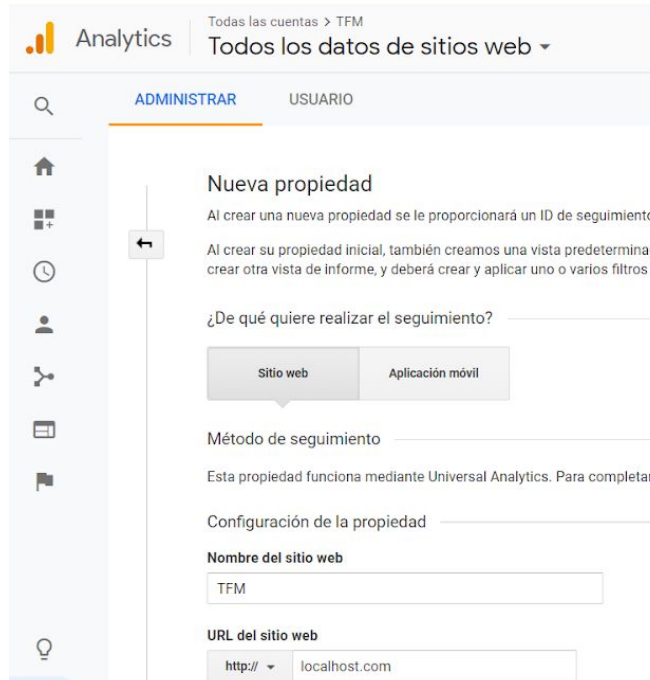
There are many resources that say it is not possible to run Google Analytics locally, but nowadays it is possible and it is an easy process (once you have read about the topic in many documentations). Probably, it is because Google Analytics changed that.

As it is explained in the point below this one, it seems an easy process but using Vue, it is harder.

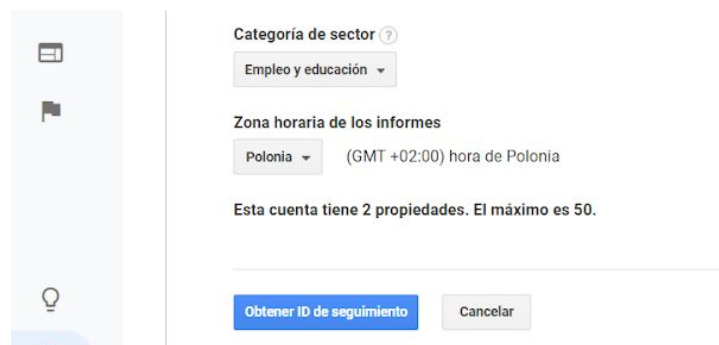
You can use your standard Google account to access Google Analytics. If you don't have an account, you can create one by going to the Google Analytics website at: analytics.google.com



Google Analytics will first ask whether you want to track a website or a mobile app. In our case, we have to select web site.

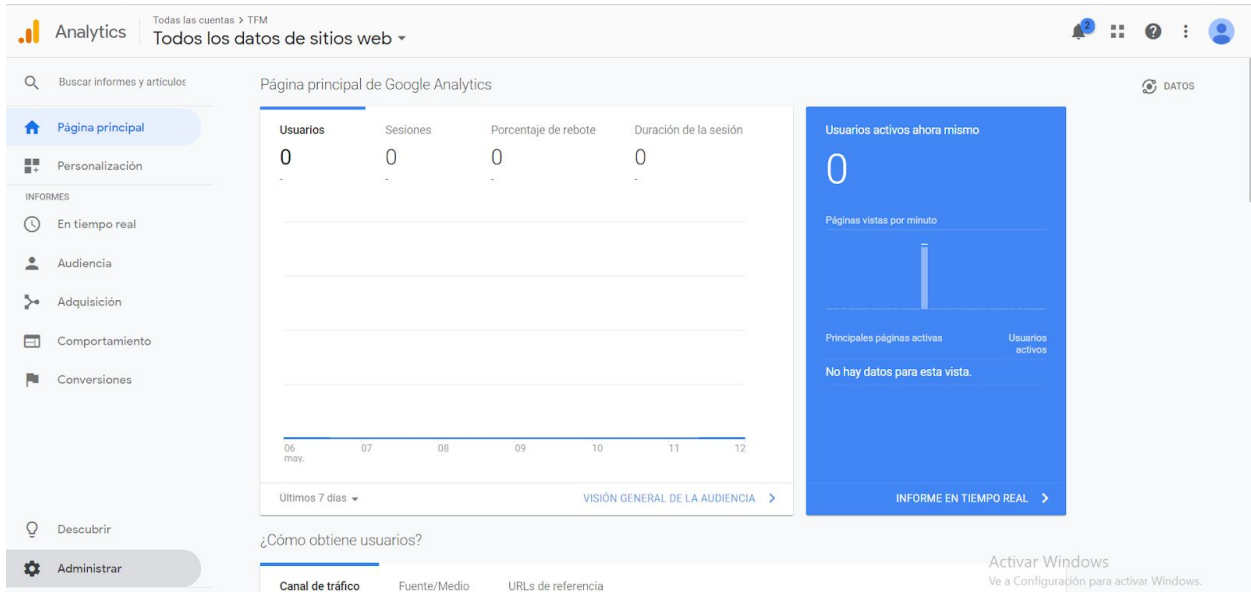


Also, it is necessary to introduce one name for the website and the URL of the website. It is important to write 'localhost.com', if we put just 'localhost' it would not work, because Google Analytics need a full domain.

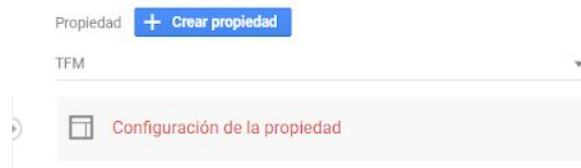


Further, we need to choose our category of sector, it could be 'occupation and education' and the zone time for the reports. Once we have filled all the requirements, we click in 'Get the tracking ID'.

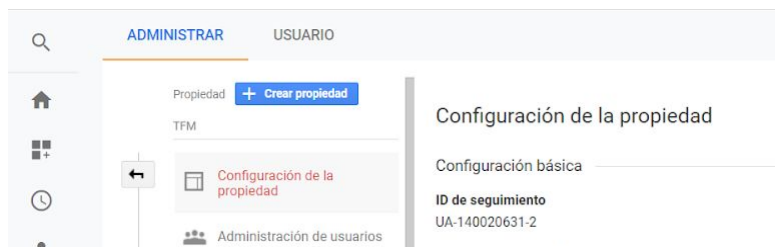
Now we are in the home page, and we have to click in the configuration/administration button, is in the left corner at the bottom.



After it, there are 3 divisions: account, property and view. We go to the property division and select 'Property configuration'.



Here, we have the tracking ID. We need to copy it because we will need it later.



Our project is already configured to be used by our code, so to use Google Analytics in our Vue project, we are going to use one library to simplify this process: 'vue-analytics' [16]

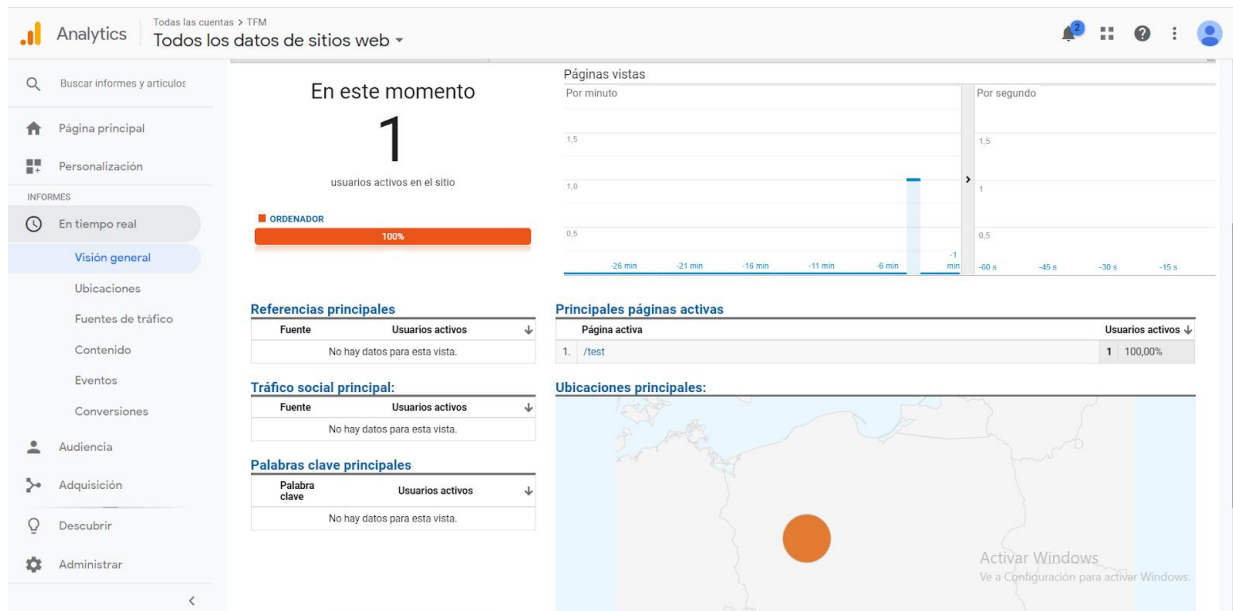
To install it, we just need to use the following command: 'npm install vue-analytics'. In our main.js we have to include this library, using the tracking ID we got in the last step.

```
import VueAnalytics from 'vue-analytics'
Vue.use(VueAnalytics, {
  id: 'UA-140020631-2'
})
```

Now we are going to test this. With the following example code, we track when a component is used:

```
mounted () {
  this.$ga.page('/test')
}
```

In our Google Analytics Project, we click in reports in real time, and we already can see that there is a user that has used the component. As you can see, in this first screen, there is information about the user (location) and also what pages are actives (/test)



Ap.D.2. Tracking with Measurement Protocol

Now is the moment to try to track one page using the Measurement Protocol. To make a `http.post` we need to use the following library: 'vue-resource'.

```
import Vue from 'vue'
import VueResource from 'vue-resource'
Vue.use(VueResource)
```

With the next code, we are going to make the post:

```
handleMP () {
  this.$http.post('https://cors-anywhere.herokuapp.com/http://www.google-analytics.com/collect?v=1&t=pageview&tid=UA-140020631-2&cid=555&dp=%2FTrackingWithMP')
}
```

Let's analyze it:

- `handleMP()`: it is just a function that handle one button that we have created. When we click that button, we launch the request.
- `https://cors-anywhere.herokuapp.com`: this is totally a trick. At present, Vue's team is working on this, but we have to use this trick to overstep the CORS security.
- `http://www.google-analytics.com/collect`: the host we are going to call.
- `v = 1`: version
- `t = pageview`: pageview hit type
- `tid = UA-120020631-2` : tracking ID / property ID
- `cid = 555` : anonymous client ID
- `dp = %2FTrackingWithMP` : page's name

Once we have visited the page, we can refresh our Google Analytics to see how has changed:

