TFM

V 0.01 “MVP” | Rafael Venegas | Assembler School

Contenido

Briefing 2

Tabla de organización de la etapa inicial del proyecto 8

Selección de las tecnologías a utilizarse en el proyecto 9

Calendario del proyecto. 10

Cronograma del proyecto. 11

Diagrama de casos de uso 12

Estructura de la base de datos 13

Wireframes 13

Log / diario 13

Estructura de los archivos 14

Registro de incidencias que se han detectado durante la ejecución del proyecto. 14

Registro de lecciones aprendidas. 14

Briefing

Introduction

After much effort at **Assembler School**, the **final master project** has arrived. Now you must **put into practice** everything learned to date to **develop a project of your choice** with which you can demonstrate everything learned through the master.

Keep in mind that the final project can serve as a **portfolio for a company** and it is important that you **follow all good practices** to evaluate this project.

What are the main objectives of the project?

- **Demonstrate all your knowledge** learned in Assembler School
- **Carry your own project** from planning to development
- **Define each task** in the best possible way **to reduce the risk of uncertainty** in the implementation and development phase.
- **Investigate all that is necessary** to reduce the uncertainty index
- Obtain in a consensual way an **evolution in time** of the amount of time **the team needs to develop the project.**

1. Organization

Initial planning should address **how the project will be approached**. The objective is not to postpone the organization of the project until the end and the use of time for other tasks that are after the organization phase. **It is possible that at some point you have carried out a project**, you have encountered situations such as:

- I'm on day 4/5 and I'm not clear on how to technically approach the project
- I have not spoken to a colleague about the project
- I have questions about the project and have not clarified them in time
- In my head I am very clear on how to approach the project but I am not able to write it in a document
- I allocate time to other tasks that are not related to the project since I consider that I have plenty of time but then "the bull catches me".
- He did not take into account that during the week I have to allocate time to other events or master class
- He did not take into account that someday an unforeseen event could arise and he did not reflect it in the planning.

- I didn't know that the project had to cover certain functionalities
- He was too long on a functionality that has prevented me from advancing broadly

That is why it is important that you plan the tasks and check that everything is going as expected. You must also be able to transmit this monitoring to the Project Manager in a clear and orderly manner. Below you can see some examples of tasks that can help you plan the project:

Task	*Phase*	*Time*	*Priority*	*Responsible*
Research on X technology	First stage	3 hours	High	Pau
Perform an analysis of the use cases that the user can perform in your project	First stage	2 hours	High	Mike
Divide the project into phases to facilitate planning and be able to measure the progress of the project	First stage	1 hours	High	Mike
…				

2. General analysis

The final master project will have **two main phases** that you must follow, the first is the **planning phase** and the second is the **implementation phase**:

2.1. Planification

As we mentioned earlier, in this phase of the project **the objective is to obtain a document that reflects how the project will be implemented**. To carry out project planning you must take into account that the following requirements are met:

- **Use cases** design
- **Wireframes** design
- Include all views of which the application will be composed
- **Research** of technologies, libraries and functionalities
- **Planning of the next phase** (Implementation)
- Establish a plan to develop the project. It is important that you deliver your own estimate in time of what you consider necessary to carry out the Project. The estimate will be validated together with the Assembler team.
- **Hosting** and **deployment**

- We recommend deploying your applications on **Heroku** for its speed and simplicity

2.1.1. Applications, Architecture and technologies

As the project is **free to choose**, you must define the structure of your project at the level of **technologies**, **architecture** and the **applications that it is made up** of if there are more than one.

The project architecture must have at least the following requirements as minimum:

- The project must have a **frontend part** and a **backend part**
- Use of **frameworks** or **libraries**
- There will be at least **two applications** in different repositories and they must be **deployed independently**
- It will be taken into account for the **evaluation of the project** to use in the **frontend** an asset tool like **webpack** or **similar**
- You will need to use a **linter** for **all the code** on **both layers**
- There must be a **covered test part** (it does not need to be 100%). The use of **dependency injection is recommended** for the correct implementation and decoupling of tests
- The **decoupling of the functionalities** as well as the **order** and **organization** of the **architecture** of your project will be taken into account in the evaluation.
- Assets such as **CSS**, **images**, **etc**. they have to have a **development version** and a **production version**
- The **OOP paradigm** and the **implementation of patterns** such as **MVC** will be **positively valued**
- In the case that you use frameworks like **REACT** and **VueJS** it is recommended to **componentize** the APP
- In the case of using a **database** it will be essential to use an **ORM**
- You will have to make your **CSS** files dynamic by using **SASS**

For the **technologies to choose** for this final project, you can use any technology you have learned during the master, but **if you want to use other technologies**, Assembler School recommends that you use the following, which we can support:

The new technologies are:

- **Google Firebase**
- **Typescript**
- **WebSockets**
- **Symfony**

2.1.2. Analysis of the UX interface

You must design the **wireframes** taking into account the **usability of the portals** for each **type of device.** You have to **take into account** that depending on whether it is **mobile oriented** you will have to make a **specific wireframe**.

2.1.3. About product

You must **include in the documentation and presentation** a section explaining the **product** you are going to develop with its **strengths**, **weaknesses**, **competition analysis** and the **cost of developing this site** in a real environment.

2.1.4. Project examples

Although the project is free to choose, we offer you a list of proposals that may be interesting to develop:

- **Project management software**
  - Real examples:
  - Trello: https://trello.com/
  - Kanbanize: https://kanbanize.com/
  - Monday: https://monday.com/
  - Asana: https://asana.com/
- **Ecommerce**
  - Real examples:
  - Nike: https://www.nike.com/
  - H&M: https://www2.hm.com/es_es/
  - Tous: https://www.tous.com/
- **Home delivery system**
  - Real examples:
  - Glovo: https://glovoapp.com/
  - Just Eat: https://www.just-eat.es/
  - Deliveroo: https://deliveroo.es/es/
- **Social media**
  - Real examples:
  - Facebook: https://es-es.facebook.com/
  - Linkedin: https://es.linkedin.com/

  - Twitter: https://twitter.com/

2.2. Implementation

Once **Assembler School** approves your **project proposal**, you can start developing the project taking into account the documentation and planning you have provided.

2.2.1. Final documentation

You must evolve the document resulting from the first phase to add the following points:

- Learned lessons
- Unplanned changes and unforeseen events
- Conclusions of the project and your evolution respect your knowledge before starting the master
- Daily log with annotations that you consider relevant for the development of the project

2.2.2. Final presentation

Create a small presentation explaining:

- How have you managed the tasks that you have previously raised
- Explain the knowledge learned
- What difficulties have arisen during the project
- Conclusions of the project and your evolution respect your knowledge before starting the master

3. Requirements

- Create a clear and orderly directory structure
- Both the code and the comments must be written in English
- Use the camelCase code style for defining variables and functions
- In the case of using HTML, never use inline styles
- In the case of using different programming languages always define the implementation in separate terms
- Remember that it is important to divide the tasks into several sub-tasks so that in this way you can associate each particular step of the construction with a specific commit
- For the project documentation a PDF version is required within the repository

- Assets generated from source code (such as sass, typescript, etc.) must be compiled during the upload process
- You should try as much as possible that the commits and the planned tasks are the same
- Delete files that are not used or are not necessary to evaluate the project
- It is mandatory that the developed applications are available in production, in the event that they are not in production, the evaluation of the project will not be possible and therefore it will be suspended

4. Testing and compatibility

Finally, you will have to verify that the project functionalities are compatible with different browsers, attaching screenshots. Don't forget to validate your code!

- Internet Explorer 11 or higher
- Safari in one of its latest versions
- Firefox in one of its latest versions
- Chrome in one of its latest versions
- Check HTML and CSS code to detect possible issues
- Verify all your unit tests are ok

5. Evaluation criteria

Concept	***% of eval.***
Choice of technologies used in the project	10%
Structure and approach of the project	20%
Project planning and documentation	20%
Approach of project delivery vs proposal	25%
Organization, structure, code simplicity, architecture, good practices and code validation	25%

6. Deliverables

- Presentation of how you will approach the project that includes:
  - Summary of the points mentioned above
  - The **presentation** must be timed with a **maximum** duration of **15 minutes**
- Final presentation at the end of the project
- Complete document with all project information

- For the development part of the project you must deliver:
  - Code Assembler repository link
  - You must create a correctly documented README file in the root directory of the project (see guidelines in Resources)

7. Key points

- Try to deploy a production version of your applications as soon as possible
- Try to be cautious when sizing your project
- In the case that you use new technologies try to write down the uncertainties in the first phase of the project
- Remember that at Assembler School you have a team of peers and counseling from mentors with whom you can ask for support
- Good project planning and organization increases your chances of success
- Remember to read this document and the requirements to understand what the evaluation criteria are

8. Resources

- Dependency injection: https://www.youtube.com/watch?v=0X1Ns2NRfks

Tabla de organización de la etapa inicial del proyecto

	Descripción	*Prioridad*	*Dificultad*	*Tiempo estimado*
Análisis de los requisitos del cliente	Lectura y análisis del briefing del proyecto	Media	Baja	1 hora
Búsqueda previa de información	Aprendizaje a través de tutoriales y ejemplos	Media	Alta	6 horas
Organización inicial	Análisis del proceso a seguir para llevar a cabo el proyecto	Alta	Media	1 hora
Documentación	Creación del archivo de la documentación	Media	Media	2 horas

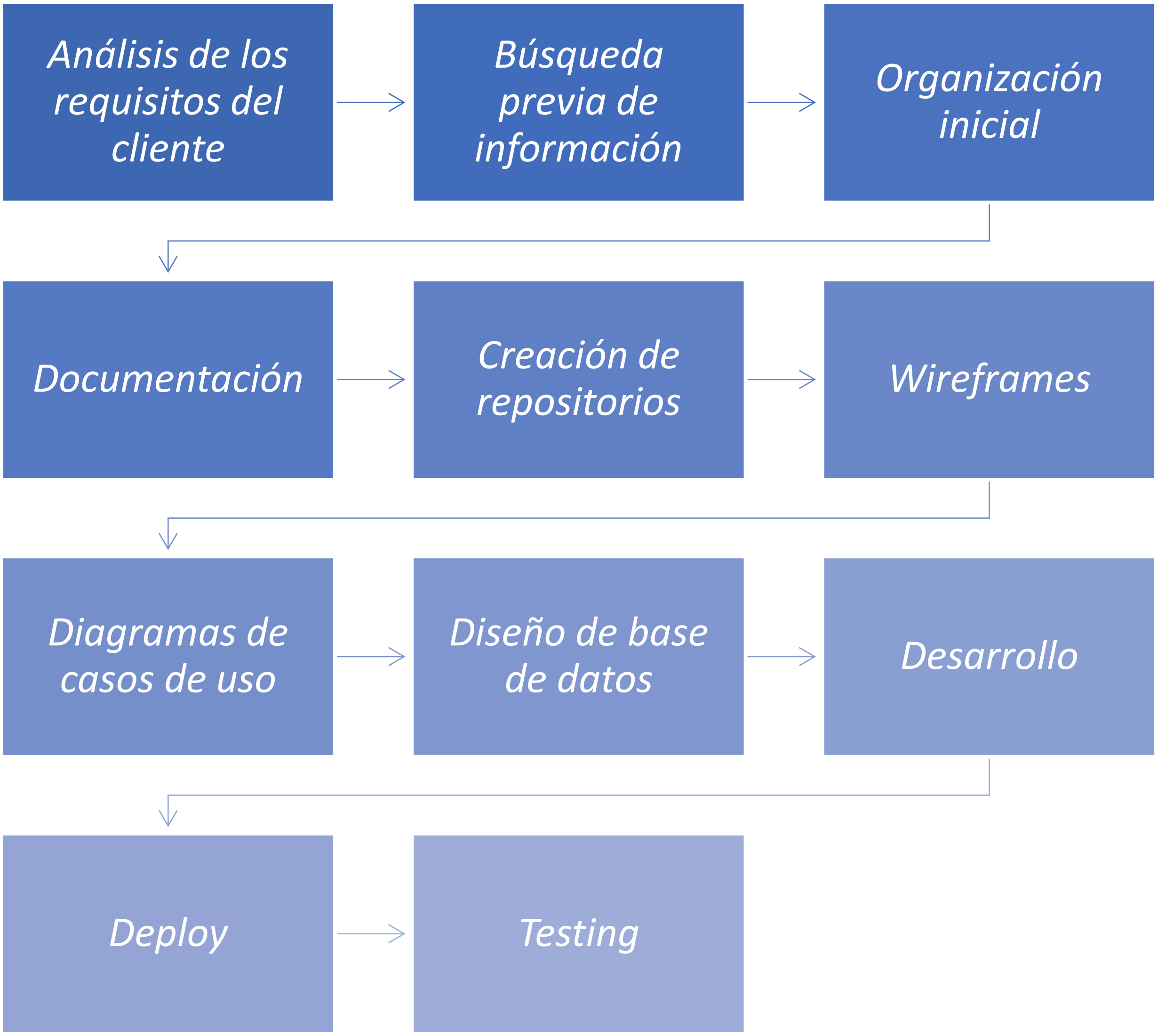
Creación de repositorios	Creación de los repositorios en Github	Baja	Baja	30 min
Wireframes	Creación de wireframes en Figma y exportación a Zeplin	Media	Alta	4 horas
Diagramas de casos de uso	Diagramación de los casos de uso para todos los usuarios	Media	Media	1 hora
Diseño de base de datos	Diseño de la estructura de la base de datos	Media	Media	1 hora
**Desarrollo*	Desarrollo de la API, BD, y la app en vuejs	Media	Alta	2 semanas
Deploy	Deploy de las apps en Heroku	Baja	Media	1 hora
Testing	Testing con JEST de varias funcionalidades	Media	Alta	4 horas

***Nota:** Los detalles de la etapa de desarrollo será descrita en los logs diarios.

Selección de las tecnologías a utilizarse en el proyecto

- Backend:
  - Firebase
  - Laravel
- Frontend:
  - Vuejs
  - Typescript

Calendario del proyecto.



Cronograma del proyecto.

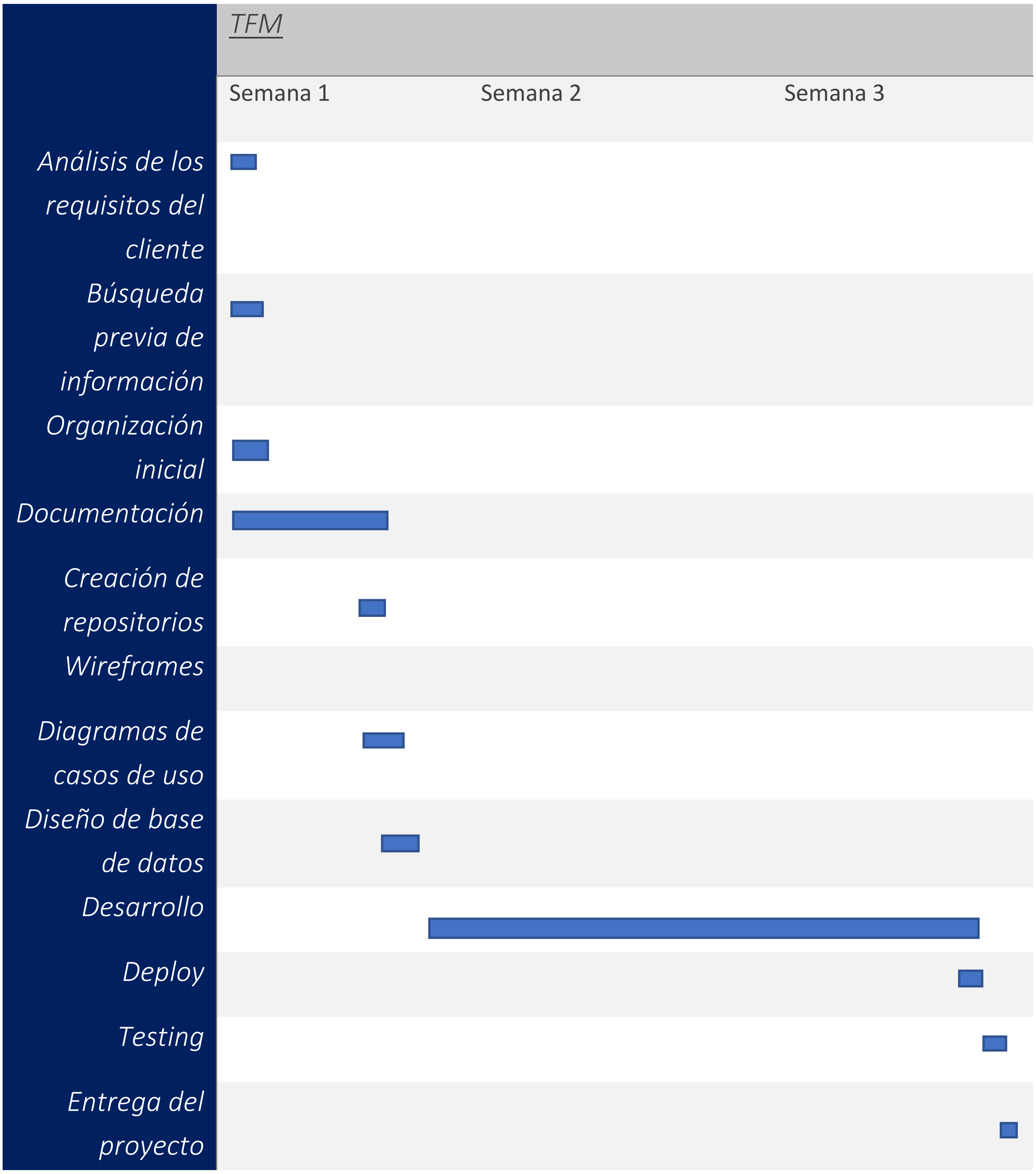
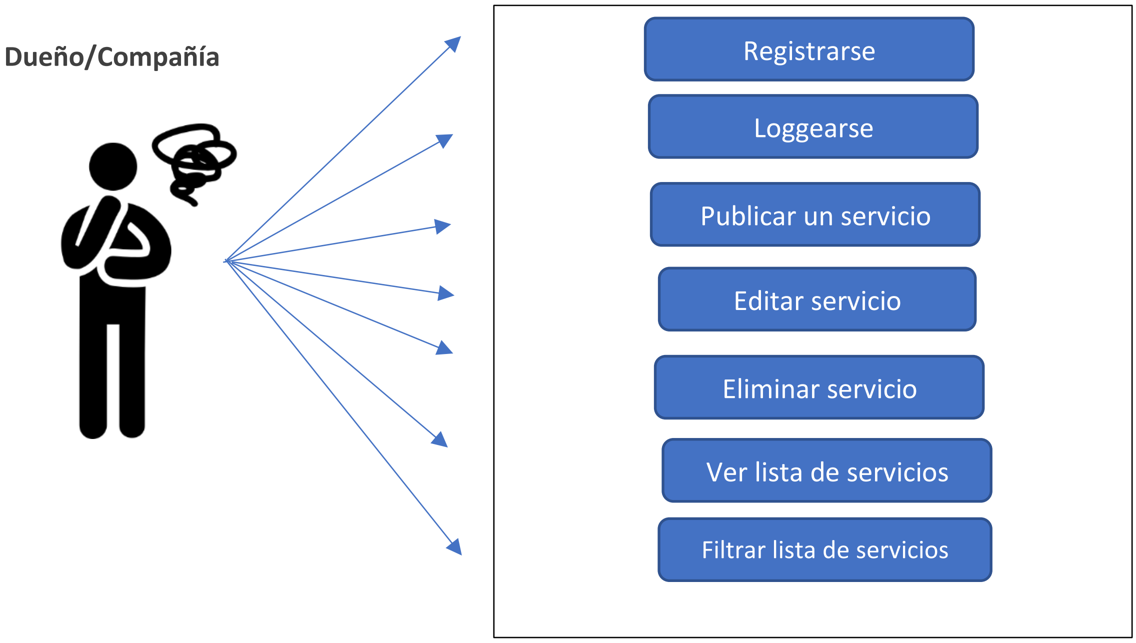
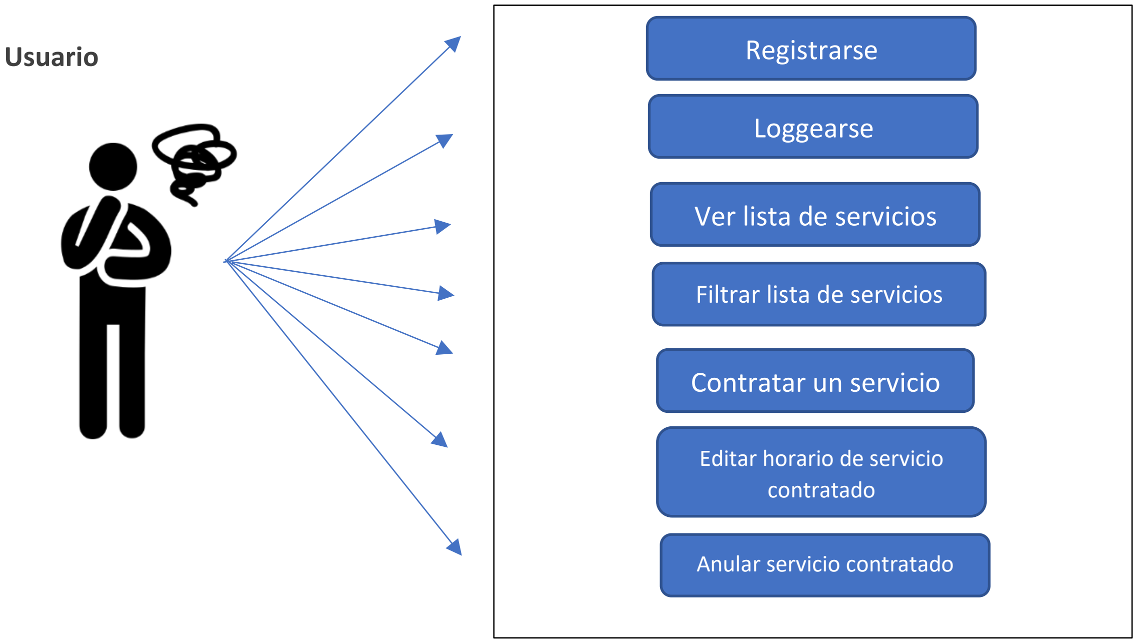


Diagrama de casos de uso



Dueño/Compañía

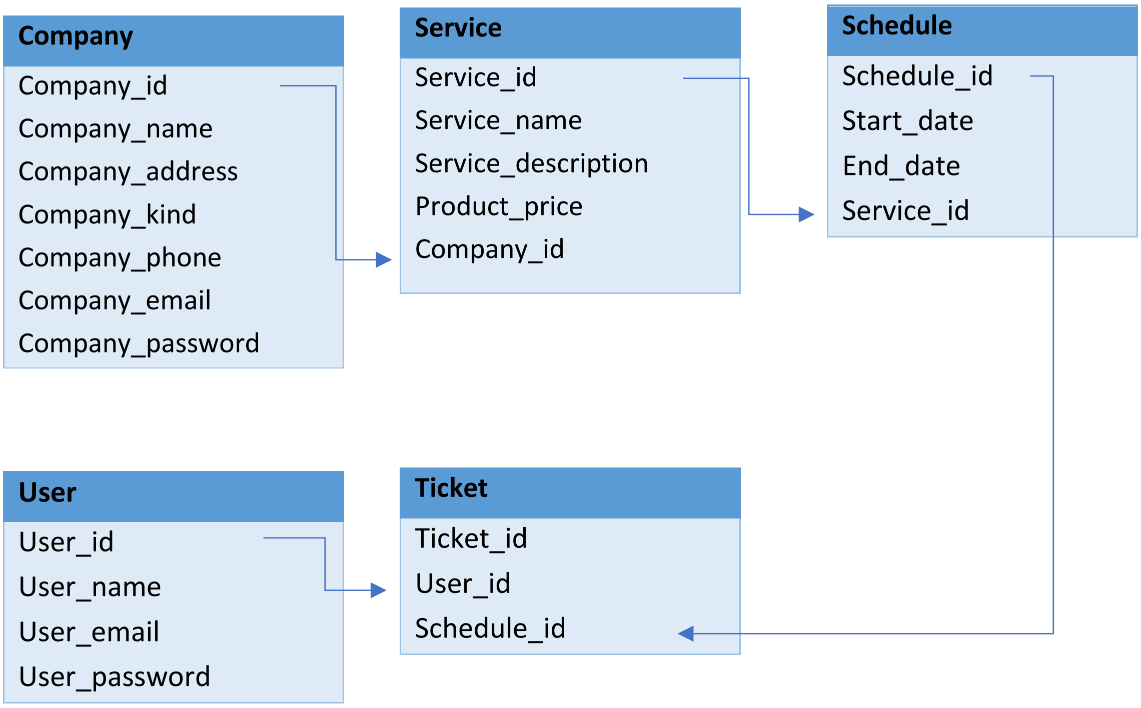
- Registrarse
- Loggearse
- Publicar un servicio
- Editar servicio
- Eliminar servicio
- Ver lista de servicios
- Filtrar lista de servicios



Usuario

- Registrarse
- Loggearse
- Ver lista de servicios
- Filtrar lista de servicios
- Contratar un servicio
- Editar horario de servicio contratado
- Anular servicio contratado

Estructura de la base de datos



Company

Company_id
Company_name
Company_address
Company_kind
Company_phone
Company_email
Company_password

Service

Service_id
Service_name
Service_description
Product_price
Company_id

Schedule

Schedule_id
Start_date
End_date
Service_id

User

User_id
User_name
User_email
User_password

Ticket

Ticket_id
User_id
Schedule_id

Wireframes

Los wireframes se realizaron en figma, y se exportaron a zeplin para su revisión desde el siguiente enlace:

https://zpl.io/bJJeejx

Log / diario

- Día 1:
  - Comienzo de la documentación
  - Selección de las tecnologías a utilizarse
  - Comienzo de los Wireframes
  - Diagramas de casos de uso
  - Estructura de la base de datos
- Día 2:
  - Finalización de los wireframes

  - Creación del repositorio
  - Estructura de las carpetas
- Día 3:
  - Finalización de la versión inicial de la documentación
  - Investigación sobre firebase y la integración en laravel
  - Inicialización de la base de datos en firebase

Estructura de los archivos

Registro de incidencias que se han detectado durante la ejecución del proyecto.

Registro de lecciones aprendidas.