

# 1 Introducción

Son varios los caminos que han llevado al desarrollo de este proyecto. En este capítulo tendré la oportunidad de poder hablar de cada uno de ellos. Desde empezar a estudiar ingeniería informática hasta la mentalidad que he ido adquiriendo a lo largo de estos años gracias a ella.

## 1.1 Motivaciones

La motivación principal ha sido la ingeniería informática y es que sin ella no sería quién soy ahora. Para bien y para mal, ha ido siendo una evolución de pequeños pasos empezando en primero de carrera hasta llegar a saber una pizca de todo este mundo de tecnología ya acabando cuarto. Todo empezó desde pequeño, desde que vi esos monitores gigantes, los disquetes y el Age of Empires 1 a la edad de cuatro años.

Mi madre tuvo un novio, Martín se llamaba, que nos trajo el mundo de la tecnología a la casa. Ella también era otra fan tecnológica, pues al haberme tenido con tan solo 17 años siempre tuvo esa mentalidad más abierta con todo el tema de nuevas herramientas. Creo que esto me acabó ayudando bastante.

Luego de esta pequeña época vino una un poco más oscura que consistía en una vida de cero tecnología. Influyó bastante el aspecto de vivir en Argentina y que el precio de los aranceles fuera bastante alto influyendo en los productos tecnológicos en su mayoría. Pasar al lado de los escaparates y ver esas pantallas brillantes con juegos, películas, canciones... era un martirio para mí.

Otro aspecto muy importante y el cual también he de agradecer fue la concesión que hizo la Junta de Andalucía allá por 2009 de aquellos pequeños ordenadores verdes. Esos sí que fueron gasolina para mis aspiraciones y para terminar hoy aquí, dentro del grado de Ingeniería Informática de la Universidad de Almería.

Mi segunda motivación principal para el desarrollo de este proyecto fue la beca extracurricular que publicó el Departamento de Informática de la Universidad de Almería, cuyo director en el momento de la concesión y en la fecha actual de redacción de este documento es Juan Francisco Sanjuan Estrada, el tutor de este proyecto.

Gracias a ella me sentí motivado para poder afianzar la actividad desempeñada en aquel pequeño trayecto de tiempo, unos seis meses, a la redacción de este trabajo.

El trabajo no se basa en lo que exactamente realicé en aquella beca, sino en una refactorización/transformación de aquel proyecto para poder convertirla en algo “fresco” podríamos llamarlo.

Y aquí pasamos al tercer y último punto de este apartado. La transformación del proyecto, el aplicar el concepto de ingeniería para poder transformar algo que era un software cerrado, con poca modularidad, difícil de entender y de ampliar en algo que merezca la pena tener. Un software que combine tecnologías y metodologías de hoy en día. Un ejemplo a seguir.

## 1.2 Objetivos

El objetivo principal de este proyecto es la creación de un sistema de gestión de inventario y creación de préstamos para el Departamento de Informática de la Universidad de Almería.

Este sitio web tiene que cumplir con unos requisitos principales, que son:

- Llevar un registro del inventario del Departamento de Informática ubicado en el edificio Científico Técnico III
- Que el estudiantado y el personal docente e investigador realicen solicitudes de préstamos para los distintos elementos ofertados dentro de la página
- Que los técnicos de servicio puedan gestionar estas solicitudes más el seguimiento del inventario dentro del edificio

Por estos puntos entendemos que la motivación principal de la herramienta es la de gestionar y organizar préstamos.

Luego de los requisitos principales que había que cumplir se nos presentaban los secundarios.

- La página web tenía que disponer de un modo adaptativo para las versiones móviles
- Había que incorporar la posibilidad de realizar un importado de datos gracias al procesamiento de una tabla de datos que es lo que antiguamente utilizaban los técnicos del departamento
- Había una serie de datos que tenían que almacenar los objetos añadidos a este sistema los cuales podían ser de tres tipos: inventarios, fungibles o kits
- Se permitiría ver un seguimiento de los préstamos realizados sobre los objetos.

En base a estos objetivos principales y secundarios realizaremos la construcción de la aplicación.

### 1.3 Planificación

La planificación del proyecto se dividirá en cinco grandes grupos: reuniones iniciales con los clientes, planificación y elaboración del desarrollo del proyecto, preparación del entorno de trabajo, construcción de la aplicación, testeo y comprobación de la aplicación y comprobación de errores.

#### 1.3.1 Reuniones iniciales con los clientes

A pesar de basarse en la reconstrucción de una aplicación las reuniones con los clientes son iniciales. Estas se hicieron en un principio del desarrollo y se mantuvieron unas pocas más a lo largo del mismo. Terminando añadiendo nuevas funcionalidades o descripciones de los productos.

#### 1.3.2 Planificación y elaboración del desarrollo del proyecto

En el desarrollo del proyecto no solo se considera la planificación de las diferentes etapas para poder ver y realizar un seguimiento del proyecto sino que también se considera la redacción y especificación de las diferentes actividades realizadas en cada una de estas. Este apartado de la planificación se desarrolla junto a todo el resto de apartados.

#### 1.3.3 Preparación del entorno de trabajo

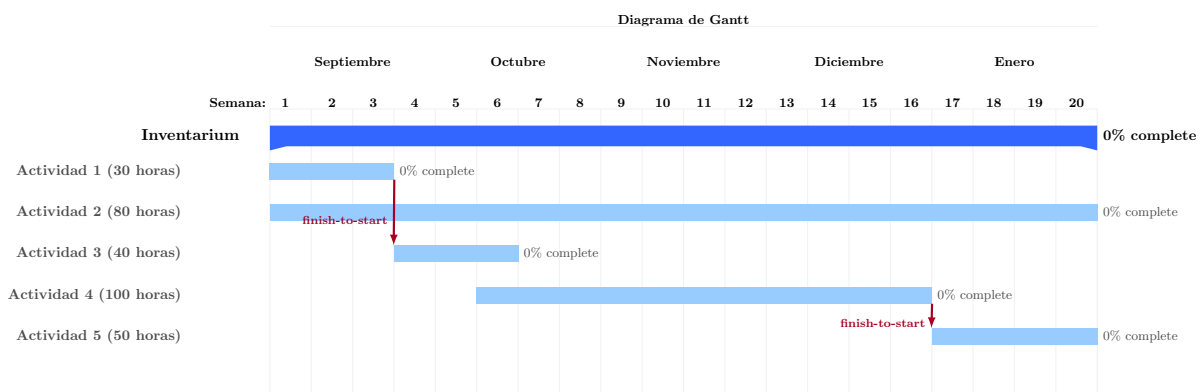
Este apartado me hace especial ilusión ya que considero que gran parte de la actividad que realizas depende del entorno donde te manejas. Los mayores casos de éxito tanto en producción como en mejoras de producto de una empresa se debe a la mejora de su entorno de producción. Gracias a algunas asignaturas del grado, a la investigación, a poder haber estado alrededor de un año en el entorno laboral y a la magnífica herramienta que es internet y todo lo que nos brinda he podido perfeccionar esta creación de entornos de trabajo y cada día siento que puedo ir mejorando un poquito más en cada aspecto que rodea a esta intencionalidad.

### 1.3.4 Construcción de la aplicación

Uno de los apartados más difíciles en cierto momento, por lo que conllevaba la creación de la lógica de la aplicación, las interacciones que realiza la interfaz con el usuario y las distintas funcionalidades que tiene que presentar la misma. Esta sección unos dos años atrás pudo haber sido de las que más tiempo podrían llevar pero gracias a los distintos frameworks que tenemos hoy en día para poder reutilizar componentes software y ahorrar pasos intermedios se ha ido volviendo más pequeña, que no quiere decir menos importante. Gracias a Angular 12 y a algunas tecnologías más que utilizaremos veremos como la construcción no es tan complicada como en un principio parecía.

### 1.3.5 Testeo y comprobación de la aplicación y comprobación de errores

¿Qué es de una aplicación bien planificada, con una interfaz bien elaborada y unos requisitos satisfechos que presente errores? No es nada, y es por ello que un testeo intensivo en el momento que la aplicación finalmente haya sido publicada hará que mejore en esos apartados que anteriormente podría haber presentado fallos.



- **Actividad 1** (30 horas) Reuniones iniciales con los clientes
  - Reunión inicial con el director del proyecto
  - Reunión con los técnicos para investigar el dominio de la aplicación
  - Presentar primer diseño de la aplicación junto a una interpretación del dominio
- **Actividad 2** (80 horas) Planificación y elaboración del desarrollo del proyecto
- **Actividad 3** (40 horas) Preparación del entorno de trabajo
  - Creación del entorno en la nube Google Cloud
  - Creación del repositorio en GitHub
  - Creación del entorno virtual en Visual Studio Code
  - Contenerización de los distintos sistemas que se utilizarán en el desarrollo de la aplicación
- **Actividad 4** (100 horas) Construcción de la aplicación
  - Elaboración y creación de la base de datos
  - Desarrollo de la API
  - Desarrollo del sitio web
- **Actividad 5** (50 horas) Testeo, comprobación de la aplicación y comprobación de errores

## **1.4 Estructuración del documento**

La estructura de este documento de trabajo fin de grado es la siguiente:

### **1.4.1 Herramientas utilizadas**

Donde haremos un pequeño resumen de todas las herramientas tanto físicas como tecnológicas utilizadas para la construcción de este software.

### **1.4.2 Desarrollo de las fases del proyecto**

Donde se describirán y detallarán todas las fases de desarrollo del proyecto. Esta es la más extensa.

### **1.4.3 Conclusiones y posibles mejoras**

Evaluaremos las distintas fases de desarrollo, su desempeño en ellas y posibles mejoras que se hubieran podido realizar.

### **1.4.4 Bibliografía**

Hablaremos sobre la bibliografía en la que se han fundamentado cada uno de los pasos de elaboración de este proyecto.

## 2 Herramientas utilizadas

En este capítulo hablaré sobre las herramientas utilizadas durante el desarrollo del proyecto. Me parece bastante interesante debido a la inclusión de nuevas herramientas que irán en conjunción a nuevas metodologías de trabajo.

Al igual que las tecnologías van avanzando las empresas también lo hacen, y la amplia gama de tecnologías gratuitas que están a nuestro alcance conlleva que el conocimiento de las mismas sea de vital importancia en el momento de construcción de una aplicación.

### 2.1 Hardware

Dentro del apartado de hardware disponemos de dos ordenadores. Su procesador no conlleva relevancia en el desarrollo de la aplicación debido a la utilización de servicios en la nube.

#### 2.1.1 Torre de PC

La cual contiene como procesador un Xeon E5-2620 V3. 16GB de RAM DDR3. Una tarjeta gráfica RTX 570 de 4GB DDR5. Tiene 256GB de memoria SSD y 1TB de memoria HDD.

#### 2.1.2 Un portátil

Es un MacBook Air M1 de 2020 con 8GB de RAM y 256GB de almacenamiento.

### 2.2 Software

#### 2.2.1 Entorno de desarrollo

Nuestro entorno de desarrollo y desde donde haremos casi absolutamente todo será desde Visual Studio Code. Este es un editor de código desarrollado por Microsoft que soporta varias distribuciones de sistemas operativos, entre ellas: Windows, Mac Os y Ubuntu.

Una de las características de esta herramienta que la hacen la predilecta de varios desarrolladores es el gran soporte que tiene por parte de la comunidad. Tiene un mercado de plugins bastante grande que apoya la creación continua de código para todos los desarrolladores.

Dispone de integraciones con Git, resaltado en errores de sintaxis, finalización de código y hasta conexión remota a otros entornos de trabajo mediante SSH.

Otra enorme ventaja que presenta es el consumo de memoria que tiene, bastante pequeño. Es un programa para ordenadores de todos los tamaños y precios, un software gratuito y una herramienta increíblemente potente al alcance de todos.

#### 2.2.2 Redacción del documento

Estas líneas están siendo escritas ahora mismo desde LaTeX. LaTeX es un sistema de composición de textos que está formado mayoritariamente por órdenes construidas a partir de comandos TeX. En un principio no estaba seguro de qué herramienta utilizar, ya que la posición de varios profesores respecto a esta herramienta era bastante férrea pero Word siempre había ido agarrado a mi mano desde comienzos del instituto.

Luego de pasar de Word a LaTeX y de LaTeX a Word bastantes veces no fue hasta que mi

profesora Rosa, en una de mis visitas matinales a su despacho me dijo: Yo hice mi TFG en LaTeX.

No me lo podía creer y al comprobar la fecha de publicación de este programa de procesamiento de textos me sorprendí al ver que su lanzamiento oficial fue en 1980. “Si el programa ha durado tanto es que algo de importante tendrá” pensé. Y aquí me hallo redactando este documento con un programa que facilita el control de versiones de Git de una manera asombrosa. Facilita también los procesos de documentación y disposición de las diferentes subsecciones. Y, lo que más me gusta sin lugar a dudas, que puedo realizar una separación de cada capítulo por documentos separados y es que a mí, el tener las cosas descompuestas, me puede.

### 2.2.3 Diseño y creación de la base de datos

La base de datos ha sido diseñada con MySQL Workbench. Esta es una herramienta visual de diseño de bases de datos, capaz de administrar, diseñar, gestionar y mantener bases de datos. Su primera versión fue publicada en 2005.

En un principio la base de datos fue exportada para que el deploy también se realizará en un servidor MySQL pero el versionado de estos scripts para la creación de las bases de datos me llevan dando problemas tres años. Grata fue mi sorpresa al descubrir MariaDB y que ofrece una compatibilidad perfecta con MySQL, además, es software libre.

MariaDB es un sistema de gestión de bases de datos derivado de MySQL con licencia GPL. Fue escrito por el mismo creador que MySQL, y esto ¿por qué? Porque vendió su producto (MySQL) a Oracle, dejando este de ser software libre.

Para la gestión de la base de datos he utilizado PHPMyAdmin. Un gestor de bases de datos que se utiliza desde páginas web. Facilita cualquier tipo de inspección que un técnico tenga que realizar por no poder hacerlo desde la aplicación web de inventarium.

### 2.2.4 Diseño del sitio web

Para realizar el diseño de la aplicación utilicé Adobe XD. Adobe XD es un editor de gráficos vectoriales desarrollado y publicado por Adobe Inc para diseñar un prototipo de la experiencia del usuario para páginas web y aplicaciones móviles.

Adobe XD apoya a los diseño vectoriales y a los sitios web wireframe creando prototipos simples e interactivos con un solo click.

### 2.2.5 Diseño de diagramas

El diseño del funcionamiento de la aplicación y el análisis de requisitos fue hecho mediante Visual Paradigm Professional. Visual Paradigm es una herramienta CASE, Ingeniería de Software Asistida por Computación, que nos brinda un gran abanico de herramientas para la ayuda del diseño, creación, planificación, análisis, documentación y un largo etcétera más de proyectos informáticos.

Esta herramienta ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo de un software a través de la representación de todo tipo de diagramas.

### 2.2.6 Diseño y modificación de imágenes

Para el diseño y modificación de imágenes utilizaremos Adobe Photoshop 2021. Adobe Photoshop es un editor de fotografías desarrollado por Adobe Systems Incorporated. Usado principalmente para el retoque de fotografías y gráficos.

### 2.2.7 NodeJS

NodeJS es un entorno de tiempo de ejecución de JavaScript. Este entorno de tiempo de ejecución en tiempo real incluye todo lo que se necesita para ejecutar un programa escrito en JavaScript. Gracias a él podremos utilizar varias herramientas que explicaré a continuación.

### 2.2.8 Levantamiento del servidor web y de la Interfaz de Programación de Aplicaciones(API)

El servicio Web se levanta desde Express. Express es framework back-end para NodeJS que está diseñado para levantar sitios webs y APIs. Es la herramienta predilecta para levantar servicios web dentro del entorno de Node.

### 2.2.9 Desarrollo del sitio web

El desarrollo del sitio web se realizará con Angular 12. Angular es sin duda el punto más importante de esta subsección.

Angular es un framework para aplicaciones web desarrollado en TypeScript, de código abierto, mantenido por Google, que se utiliza para crear y mantener aplicaciones web de una sola página. Su objetivo es aumentar las aplicaciones basadas en navegador con capacidad de Modelo Vista Controlador (MVC), en un esfuerzo para hacer que el desarrollo y las pruebas sean más fáciles. Angular se basa en clases tipo Componentes, cuyas propiedades son las usadas para hacer el binding de los datos.

Angular es la evolución de AngularJS (la versión de Angular que usaba JavaScript) aunque incompatible con su predecesor.

Este framework para mí ha sido como el descubrimiento de América. Y es que en la anterior versión de la aplicación había utilizado únicamente PHP, Javascript y CSS. Es más, había conseguido desarrollar un modelo basado en jQuery para que el sitio web se ubicara en una única página y fuera refrescando los datos de forma interactiva y fluida.

Pero Angular fue un cambio total de mentalidad, la gestión de componentes es lo que le da la vida en su totalidad y es que, ¿qué sería un desarrollo software sin que presente una modularidad en sus componentes? Angular te lo ofrece, y te da más. Podemos definir todo el dominio de la aplicación dentro de ella y crear los servicios para que estén listos para usarse para contactar con la API. Luego podemos definir el sistema de rutas que deba tener la aplicación junto al resto de componentes que tienen que interactuar con el usuario.

No podría imaginarme un desarrollado planificado, con sus diagramas de requisitos, de base de datos, diseño de componentes y estructuración de vistas sin este framework.

## 2.3 Servicios

### 2.3.1 Sistema de control de versiones

El sistema de control de versiones se realizará gracias a GitHub. GitHub es una plataforma de desarrollo colaborativo para alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de aplicaciones.

El software que opera GitHub fue escrito en Ruby on Rails. Desde enero de 2010, GitHub opera bajo el nombre de GitHub, Inc. Anteriormente era conocida como Logical Awesome LLC. El código de los proyectos alojados en GitHub se almacena normalmente de forma pública.

El 4 de junio de 2018 Microsoft compró GitHub por 7500 millones de dólares. Al principio, el cambio de propietario generó preocupaciones y la salida de algunos proyectos de este repositorio, sin embargo no fueron representativos. GitHub continúa siendo la plataforma más importante de colaboración para proyectos Open Source.

### 2.3.2 Gestión y creación de máquinas virtuales

Este apartado no creí que lo llegaría a utilizar. En la Universidad de Almería tenemos OpenStack un servicio que nos ofrece el poder crear y destruir máquinas virtuales además de poder ubicarlas en una infraestructura de red. El problema que tuve con OpenStack fue la propia configuración de entorno de red de la universidad. Y es que había unos determinados puertos que por mucho que los abriese en la máquina virtual el cortafuegos de la Universidad me impedía acceder a ellos. Como veremos más adelante a lo largo de estas páginas esto no tuvo por qué haber sido un problema pero lo descubrí más tarde.

Debido a los problemas anteriormente mencionados con OpenStack me dispuse a buscar las soluciones que me brindaban las distintas “empresas top” del sector.

Las iré evaluando por precio, rendimiento e interfaz. Las he probado todas.

#### **Amazon Web Services**

En precio se encuentra en un rango medio. Creo que de las tres es la que mejor se enfocaba en lo que buscaba. Disponía de buena documentación y la interfaz era compleja pero fácil de usar. La conectividad y el acceso a las máquinas era rápido pero los planes y servidores eran algo limitados.

#### **Microsoft Azure**

El precio es alto, bastante a mi parecer. El manejo y creación de base de datos se complica por el plan de precios que presentan que están almacenados en unos tipos de monederos. No creo que sean malas ideas pero sí que están mal implementadas dificultando el aprendizaje inicial. Los servidores y la configuración de los mismos son más limitados que los de Amazon Web Services.

#### **Google Cloud**

Ofrecen el mejor precio de la competencia. El problema es que la primera vez que entras te encuentras con muchas soluciones que ofrece Google con servicios en la nube. Es una sobreinformación de cosas que realmente no necesitaba para desplegar una simple máquina virtual. Una vez te ubicas que tienes que moverte en el entorno de “Compute Engine” todo se vuelve mucho más fácil. Las máquinas virtuales son fáciles de desplegar y de personalizar y presentan una alta gama de servidores disponibles, al fin y al cabo Google está en todo el mundo.

Mi elección fue Google Cloud. Sin duda para el proyecto iba a ser la mejor opción.



## 3 Fases previas a la construcción de la aplicación

La temporalidad del proyecto no es exacta del todo ya que los puntos 3.1 y 3.2 se hicieron aquellos dos primeros meses de prácticas extracurriculares mientras que los otros son totalmente nuevos.

Para el seguimiento de las fases y tareas se han estado utilizando la herramienta Trello y las nuevas funcionalidades de GitHub que incorpora tableros parecidos a los de Trello.

### 3.1 Reuniones iniciales con los clientes

La reunión inicial fue con Juan Francisco Sanjuan Estrada donde me explicó los distintos objetivos que se tenía pensado para la aplicación y los puntos de vista de los técnicos. La organización, gestión y préstamos del inventario se realizaba mediante la utilización de hojas de cálculo.

Dentro de las hojas de cálculo se hacía un detallado exhaustivo del objeto y también se indicaba su ubicación actual dentro del Edificio Científico Técnico III.

El problema es que esto no estaba tan organizado como parecía en un principio debido a que la gestión no la realizaba únicamente un técnico sino que eran tres. Es decir, había tres filosofías distintas a la hora de ir gestionando una parte del inventariado del Departamento de Informática. Después de estas aclaraciones Juan me comentó que no tenían un diseño en mente en el Departamento por lo que las propuestas iniciales de diseño iban a ser libres siempre y cuando se satisficieran los requisitos principales de esta como la adición y eliminación del inventario y la concesión de préstamos.

Luego de esta primera reunión acordamos en que podría visualizar los archivos Excel de los técnicos. Estos me compartieron sus ficheros y desde ahí pude empezar a realizar la definición de los diferentes campos que irían ligados a cada tabla en la base de datos. Desde esa primera definición de campos se haría una propuesta inicial.

La propuesta inicial fue esta:

La lógica de la aplicación consistiría en generar un grupo de objetos de un determinado tipo, inventario o fungible, 1 o 0. Dentro de este se contendrían objetos que irían vinculados a una ubicación. También los objetos irían vinculados a una posible configuración.

Esta configuración ¿para qué serviría? Resulta que los técnicos aparte de manejar el inventariado disponible en la universidad también manejan las claves y accesos a esos determinados objetos, como puede ser el armario donde se ubiquen, sus direcciones mac o ip, las bocas de conexión con las regletas y más.

Este objeto tendría la posibilidad de tener varios préstamos, aunque es verdad que sería uno activo por persona, siempre dejando un registro de los anteriores usos que se hayan realizado del mismo. Este préstamo tendría que ir ligado obligatoriamente a un usuario. Contiene un campo de “retiradoPor” en el caso de que un docente o investigador quiera realizar un préstamo para su alumno.

La propuesta inicial fue aceptada por el grupo de los técnicos y por el director del proyecto así que podíamos continuar hacia adelante. Más adelante se realizaron unas modificaciones en los datos que manejarían los objetos por lo que las columnas de la tabla son las definitivas.

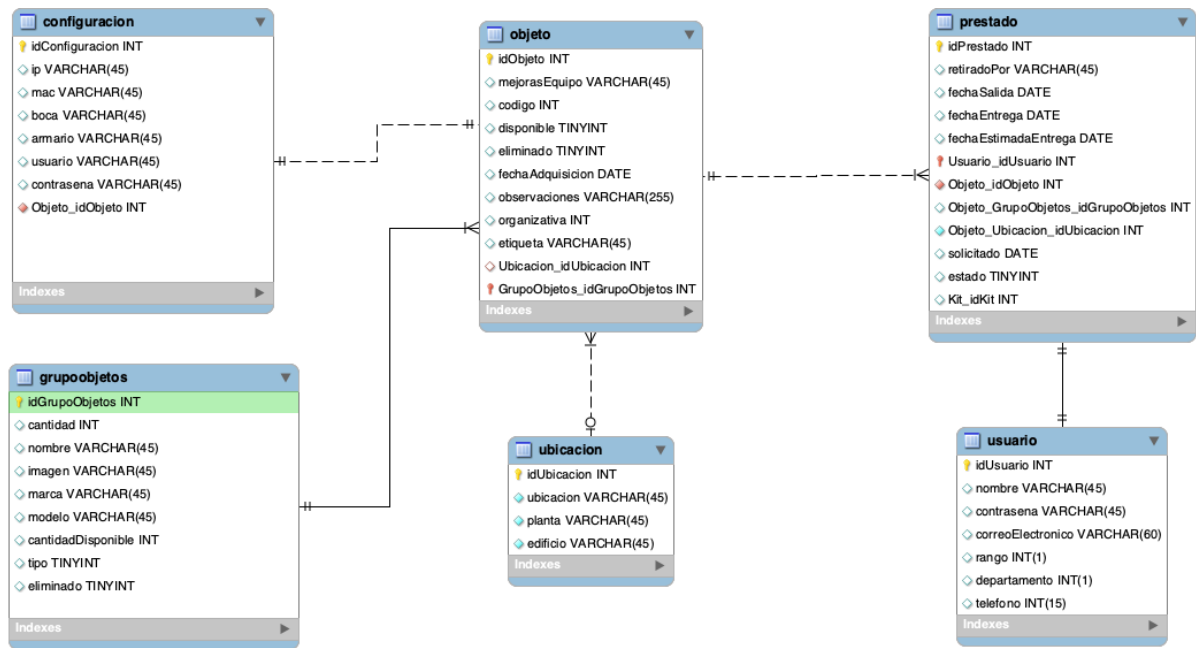


Figure 3.1: Diseño principal de la base de datos

Preparé las propuestas iniciales de diseño mediante la utilización de la herramienta de Adobe XD. El aprendizaje de la herramienta es rápido y fácil de usar, aparte de que ya la había empezado a utilizar unos años atrás. Adobe XD le dió un toque de frescura y profesionalidad a las propuestas de diseño.

Las propuestas iniciales fueron ligadas a un diseño móvil ya que desde el primer momento se buscó que fuera responsive el sitio web. Esto influyó luego a la hora de realizar el diseño ya que se basó en un funcionamiento de “cards”, es decir, componentes web con forma de carta y con fácil adaptabilidad a diseños móviles.

El esquema de navegación de la propuesta inicial nos quedaría tal como podemos ver en la Figura 3.2.

Tenemos dos roles de usuario, un personal docente o investigador y un técnico. Después del análisis hecho en la entrevista y correos mantenidos más tarde con el cliente teníamos el siguiente resultado en la Figura 3.3.

#### 3.1.1 Inicio de sesión y registro

El registro del usuario no conllevaba un registro instantáneo del mismo ya que este tiene que ser dado de alta por un técnico del sistema.

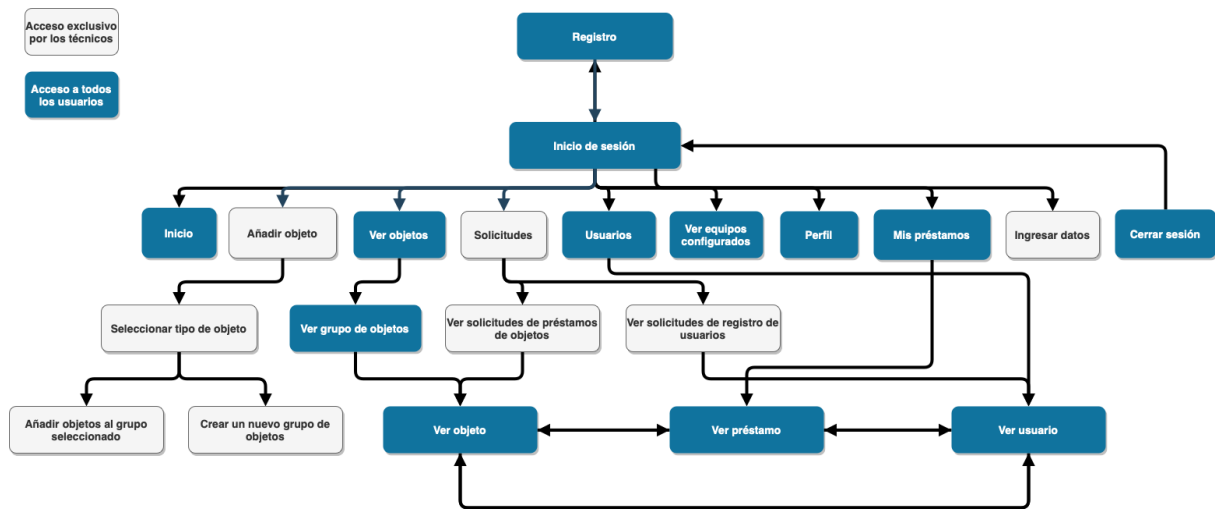


Figure 3.2: Diseño principal del sistema de navegación

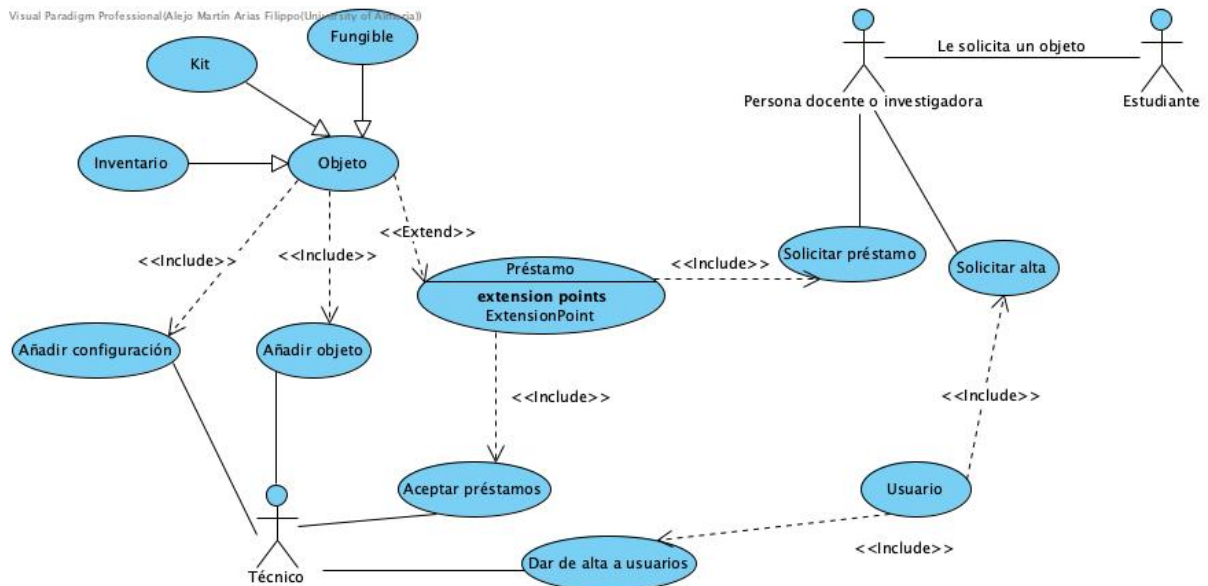


Figure 3.3: Diagrama de casos de uso del funcionamiento principal de la herramienta Inventarium

UAL-Inventarium

Correo electrónico

Contraseña

Iniciar sesión

Solicitar acceso

Solicitar acceso

Nombre completo

Correo electrónico institucional

Contraseña

Repetir contraseña

Departamento

Teléfono

Figure 3.4: Diseño del inicio de sesión y del registro

Dentro del inicio de sesión vemos un campo interesante a considerar y es el de departamento. Una de las solicitudes que hizo el director del proyecto era poder agrupar a los usuarios dentro de departamentos. En este caso refiriéndose a dentro del personal docente e investigador que pertenecen al Departamento de Informática.

Los valores del departamento en la base de datos pueden ser los siguientes:

- 0 = Departamento de informática (valor por defecto)
- 1 = Ingeniería de sistemas y automática
- 2 = Lenguaje y sistemas informáticos
- 3 = Ciencias de la computación e inteligencia artificial
- 4 = Arquitectura y tecnología de computadores

Otro campo para comprobar es el del correo electrónica institucional teniendo que ser este con la extensión **@inlumine.ual.es** o **@ual.es**.

#### 3.1.2 Barra de navegación

La barra de navegación vertical tuvo una pequeña modificación que era incorporar una pequeña sección de inicio donde aparecieran información pertinente y relevante para el usuario. Esto será explicado más adelante en la parte de desarrollo.



Figure 3.5: Diseño de la barra lateral de búsqueda

Dentro de ella podemos ver cinco apartados que derivan en varios componentes visuales:

- Añadir objeto: un acceso rápido donde poder añadir objetos dentro de grupos de objetos. Sección de menú solo visible para los técnicos.
- Ver objetos: apartado donde se visualizan los grupos de objetos, que contienen tanto el inventariado, fungibles y kits.
- Solicitudes: un menú donde se visualizan las solicitudes tanto de alta de usuarios como de objetos que le puedan llegar a los técnicos.
- Usuarios: componente visual donde cargan los usuarios registrados que hay dentro de la aplicación.
- Ver equipos configurados: aquí cargan únicamente objetos a los que se les haya aplicado una configuración. En caso de que no, no aparecerían.

### 3.1.3 Añadir objeto

Añadir objeto

Inventario

Fungible

Nombre \*

Imagen \*

Cámara

Ubicación \*

Ubicación

Marca

Modelo

Mejoras en el equipo

Cantidad

2

+

-

Código del inventario 1 \*

Código del inventario 2 \*

Configuración

Aplicar

Añadir

Añadir objeto

Inventario

Fungible

Nombre \*

Imagen \*

Cámara

Ubicación \*

Ubicación

Marca

Modelo

Mejoras en el equipo

Cantidad

1

+

-

Código del inventario 1 \*

Código del inventario 2 \*

Configuración

Aplicar

Añadir

Figure 3.6: Diseño de la característica para poder añadir fungibles e inventarios

La única diferencia entre añadir un objeto de tipo inventario y otro fungible es que en el de inventario hay que registrar su código. Siempre que vayamos a añadir uno de estos elementos tiene que ser dentro de un **grupo de objetos** que debemos haber creado anteriormente. Por el resto son los dos idénticos y se le puede añadir el mismo tipo de información. No solamente tenemos fungibles e inventarios sino que también tenemos kits.

Un **kit** podemos considerarlo como otro nivel de agrupación. Ya que un kit contiene más objetos dentro suyo que llamaremos “objetos kit”. Para entender los niveles de agrupación aquí tenéis un esquema explicándolas.

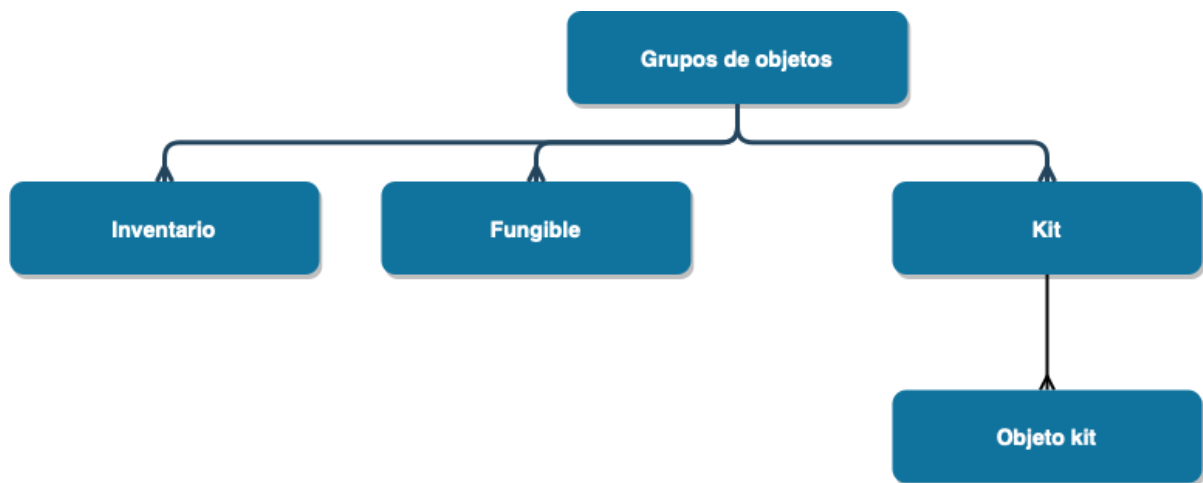


Figure 3.7: Jerarquía de elementos que derivan de grupo de objetos

### 3.1.4 Grupo de objetos y objetos

Dentro de la sección de “ver objetos” de la barra de navegación nos llevará a la visualización de los grupos de objetos de la aplicación. En las cuales habrá también objetos. Estos dos conjuntos se pueden modificar y además se pueden pedir solicitudes para el inventariado, fungibles y kits.

Figure 3.8: Grupo de objetos a la izquierda y objetos que contiene a la derecha

### 3.1.5 Usuarios

En este apartado podremos visualizar los usuarios que están registrados en la aplicación. Podemos acceder a cada uno de los perfiles para ver sus respectivos datos y sus solicitudes o historial de préstamos.

The figure shows two forms side-by-side. The left form, titled 'Usuarios', contains two text boxes with the names 'Juan Francisco Sanjuan Estrada' and 'José Andrés Moreno Ruiz'. Below these is the text '20 usuarios máximos por página'. At the bottom are three buttons: '1', '2', and 'Siguiente'. The right form, titled 'Usuario', contains four text boxes: 'Nombre: Juan Francisco Sanjuan Estrada', 'Correo electrónico: jsanjuan@ual.es', 'Departamento: Informática', and 'Teléfono: 950 214017'. Below these is a dropdown menu labeled 'Objetos prestados' with a downward arrow. The dropdown is open, showing two options: 'Objeto' and 'Objeto'.

Figure 3.9: Grupo de usuarios a la izquierda y usuario único a la derecha

#### 3.1.6 Préstamos

Los préstamos se pueden encontrar en tres diferentes estados:

- Préstamo en curso: una solicitud de préstamo aprobada por un técnico y que está en vigencia.
- Préstamos solicitado: donde aparecerá una solicitud en espera a ser aprobada por un técnico, en el caso de que el objeto solicitado esté actualmente con un préstamo en curso no se permitirá conceder el préstamo solicitado hasta que finalice el actual.
- Préstamo finalizado: un préstamo en curso que ha sido devuelto, la aprobación de devolución la tiene que conceder un técnico.
- Préstamo vencido: un préstamo en curso que ha sobrepasado su fecha máxima que se había comprometido a devolver. En caso de que quiera solicitar una ampliación tiene que hacerlo nuevamente generando una nueva petición.

Cada solicitud irá con un color representativo dependiendo de cuál sea su estado, una utilidad tanto para el usuario como para el técnico que van a realizar las acciones de ir solicitando y concediendo préstamos.

The figure shows a form for requesting a loan. It has a text box labeled 'Nombre completo del destinatario\*'. Below it is a text box labeled 'Cantidad' with the value '1' and a dropdown arrow. Below that is a text box labeled 'Fecha estimada de entrega \*' with a button labeled 'Seleccionar'. At the bottom are two buttons: 'Cancelar' and 'Solicitar'. A small note above the date field says 'Se podrá seleccionar para que sea indefinida'.

Figure 3.10: Solicitud del préstamo de un objeto



### 3.1.7 Configuraciones

Un objeto puede ir ligado a una información extra o no. Esta información no es algo que vuelva más descriptivo el objeto sino que es una utilidad que se solicitó en el momento del desarrollo de la aplicación. Su objetivo es poder acceder a información a la que suelen recurrir los técnicos de forma bastante frecuente. Como la IP o dirección MAC de un ordenador o las credenciales para poder acceder a él. Gracias a esta distinción podemos hacer que la información de los equipos que disponen de una configuración en particular estén en una sección para ellos solos y facilitarles el acceso a los mismos a los técnicos.

Figure 3.11: Visualización de los campos de configuración a la izquierda y creación de estos a la derecha

## 3.2 Planificación y elaboración del desarrollo del proyecto

Esta sección implica dos apartados de relevante importancia que es de la forma en que la subdividiremos. Por una parte tenemos:

### 3.2.1 Redacción del proyecto

Para la redacción del proyecto como he explicado anteriormente se está utilizando la herramienta LaTeX. Esta serie de archivos viene incorporado al repositorio principal del proyecto dentro de una carpeta llamada documentación.

Nuestra carpeta de documentación a la vez está subdividida en varios apartados:

#### Build

Carpeta que viene por defecto incorporada dentro de la “compilación” del proyecto que realiza LaTeX. Dentro de ella se genera nuestro documento PDF que gracias a Visual Studio Code se va generando cada vez que se guarda un archivo del proyecto que vaya ligado o afecte al main.tex

#### **Bibliografía**

En este apartado añadimos la bibliografía utilizada durante la realización del proyecto.

#### **Capítulos**

Dentro de esta carpeta guardaremos cada uno de los capítulos del documento. En el caso de que un capítulo presente apartados demasiados extensos generamos una carpeta con el nombre del capítulo y metemos las secciones dentro de este. Gracias a hacer esto la modificación de cada zona del documento se hace de una forma mucho más cómoda.

#### **Diagramas**

Aquí almacenamos los diagramas del proyecto. Estos pueden estar en archivos de imágenes aunque también pueden estar generados mediante LaTeX. Debido a la unicidad que presentan los diagramas estos no están separados en diferentes carpetas por capítulos tengamos.

#### **Imágenes**

Como su propio nombre indica almacenamos las imágenes del proyecto. Estas generalmente están divididas por carpetas con el nombre de los capítulos. En el caso de que sea demasiado extensa también se contempla la posibilidad de realizar el almacenaje mediante secciones.

#### **Include**

Aquí añadiremos los archivos que generen inclusiones dentro de nuestro documento. Dependiendo de para qué sean estos añadidos irán con unas determinadas agrupaciones u otras.

#### **main.tex**

Esto no es un directorio pero es un archivo de relevancia. Su contenido es escaso debido a la generalización que presentamos en nuestra disposición de la documentación. Dentro de él nos podemos encontrar las inclusiones al principio del documento. La adición del índice, los capítulos y de la bibliografía al final de la página. Es el archivo que utiliza LaTeX para generar nuestro documento PDF.

### **3.2.2 Elaboración del desarrollo del proyecto**

Esta sección consiste en la explicación y descomposición de pasos que vamos a hacer para la elaboración de nuestra aplicación. Una vez hemos hecho la lógica de la aplicación, su dominio y sus casos de uso procederemos a la maquetación de nuestra aplicación:

#### **Generación de la base de datos**

Generaremos nuestra base de datos, para ello utilizaremos MariaDB para el despliegue de esta.

#### **Construcción de la Interfaz de Programación de Aplicaciones (API)**

La creación de nuestra API hay que hacerla de forma muy cuidadosa, ya que tiene que incorporar todas nuestras reglas de negocio y cualquier paso adicional que haya que añadir en cada una de las peticiones. Estos pueden consistir en que cuando un objeto es creado hay que sumar una unidad dentro del campo de “objetos” y “objetosDisponibles” de su respectivo grupo de objetos.

## Construcción de la aplicación

Detallaremos cada uno de los pasos más adelante debido a que la construcción de esta depende en pequeña medida al framework utilizado que en este caso es Angular. En todo caso el orden de creación de ficheros sería:

1. Creación de interfaces
2. Creación de servicios
3. Creación de componentes visuales

## 3.3 Preparación del entorno de trabajo

La preparación del entorno de trabajo es una de las secciones que me hacen especial ilusión de este trabajo. Como decía Robert Owen, un reconocido empresario galés, en el siglo XVIII:

Mejorando el entorno se mejora al hombre.

En nuestro caso en vez de hombre podría ir mi nombre, ya que me ha ahorrado mucho tiempo esta preparación previa, o, el proyecto.

Esta sección irá estructurada en consonancia a la cronología de creación de cada parte del entorno. Empezamos por la semilla de todo: Github.

### 3.3.1 GitHub

Realizaremos la creación de nuestro proyecto en GitHub. Esto se hace de forma muy rápida y fácil.

1. Nos dirigimos a la dirección de `github.com`

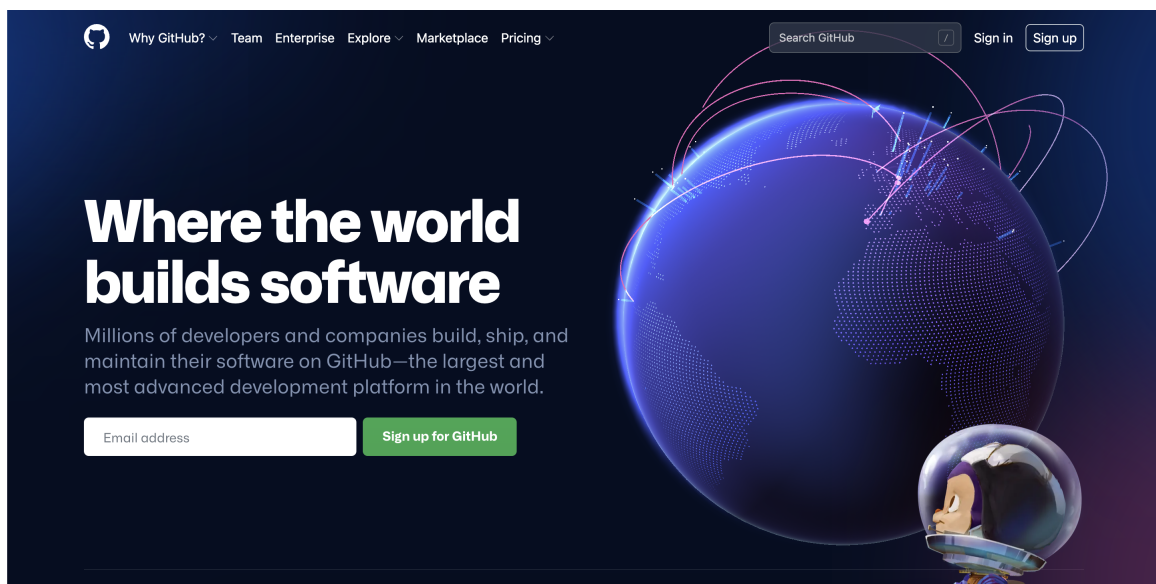
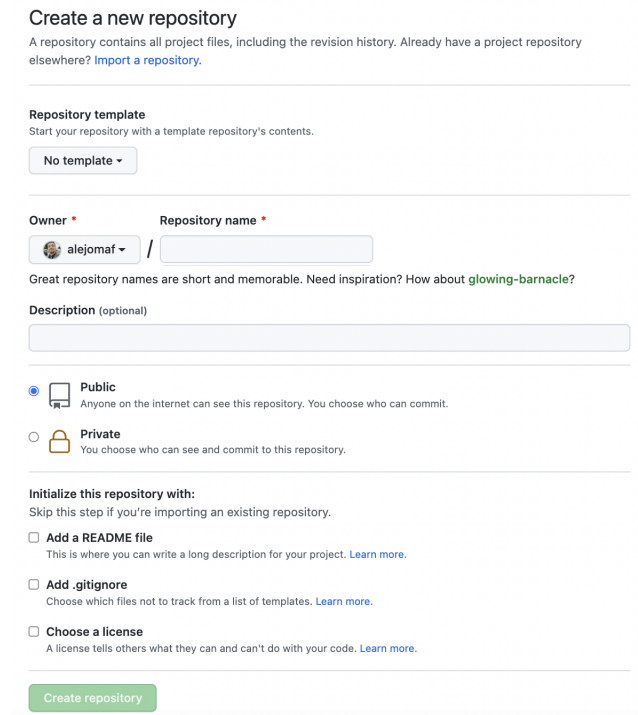


Figure 3.12: Página principal de GitHub

2. En la parte superior derecha nos aparece “Sign up” donde clicaremos y nos permitirá registrarnos dentro de la web.

3. Una vez registrados volvemos a [github.com](https://github.com) y clicamos a la izquierda de “Sign up” que pone “Sign in” donde nos registraremos.
4. Cuando hayamos iniciado sesión nos dirigiremos a la parte superior izquierda y clicaremos en “new” para crear un nuevo repositorio.
5. Rellenamos los campos que nos solicitan para crear un nuevo repositorio. Le damos el nombre de Inventarium y seleccionamos la opción para que el repositorio sea privado. También procedemos a añadirle un readme al archivo para poder describir partes del proyecto dentro de este. Por último pulsamos el botón “Create repository”.



The screenshot shows the GitHub 'Create a new repository' page. At the top, it says 'Create a new repository' and explains that a repository contains all project files. Below this, there's a 'Repository template' section with a 'No template' button. The 'Owner' is set to 'alejomaf' and the 'Repository name' field is empty. A note suggests repository names should be short and memorable. The 'Description (optional)' field is also empty. Under the 'Public/Private' section, 'Public' is selected with a radio button. The 'Initialize this repository with:' section has three checkboxes: 'Add a README file' (unchecked), 'Add .gitignore' (unchecked), and 'Choose a license' (unchecked). At the bottom, there is a green 'Create repository' button.

Figure 3.13: Creando un nuevo repositorio dentro de GitHub

Con estos sencillos pasos ya tendríamos creado nuestro repositorio.

#### 3.3.2 Visual Studio Code y Google Cloud, los mejores amigos

Hoy en día lo nuevo y a lo que nos dirigimos es la nube. Dentro de ella se pretenden que se realicen todos los procesos. La mayoría de herramientas y servicios que se nos ofrece hoy en día cumplen un modelo de caja negra. Es decir, interactuamos con él y recibimos respuestas pero no vemos qué procesos ocurren dentro de aquella cajita.

Por esta razón cada vez se empieza a disponer menos de un software como tal y se empiezan a pasar a servicios en línea. Esto no quiere decir que se dejen de usar programas o aplicaciones móviles. Pero la realidad es que sin internet; la mayoría no funcionaría.

Esto presenta desventajas siendo la principal que se depende constantemente de una conexión en línea que puede parecer que pasa en todos sitios pero en lugares remotos lejos de la ciudad como son los pueblos de montaña el internet no es el mejor compañero por decirlo de una manera. Por suerte este no es mi caso.

Las ventajas son enormes aunque solo nos centraremos en las que para mí implican mayor relevancia.

El tener ya un repositorio generado con nuestro proyecto subido dentro de él ya nos aporta

bastante autonomía ya que podemos acceder a él y descargar nuestros datos desde cualquier lugar. Pero, ¿y configurarlo?

Aquí viene una problemática que no nos resuelve un entorno de repositorios. El tener que configurarlo todo cada vez que queremos trabajar desde un sitio distinto. Esto nos lo resuelven las máquinas virtuales en la nube. El generar un directorio de trabajo donde poder conectar nuestro Visual Studio Code y olvidarnos de preocupaciones.

Lo segundo que me parece más importante es el ahorro de memoria tanto de RAM como de disco y de procesos que se origina al hacer esto. No es lo mismo tener que trabajar con un portátil conectado todo el día a un enchufe que poder ir llevándotelo contigo por el escaso consumo de su batería. Peor si es una torre, solo puedes trabajar desde un único sitio, con la problemática también de que siempre se te puede cortar la luz, y más en una zona de pueblo de sierra como es la mía, suele ocurrir una vez cada dos semanas.

Estas ventajas expuesta son las que personalmente me favorecen a mí, pero ¿y si trabajara con un equipo? El poder tener varios compañeros de un mismo equipo trabajando en el mismo proyecto, tocando los distintos componentes que se están desarrollando dentro de tu aplicación es fantástico.

#### **Crear una máquina virtual en Google Cloud**



## **4 Construcción de la aplicación**

**4.1 Generación y puesta en marcha de la base de datos**

**4.2 Creación y puesta en marcha de la Interfaz de Programación de Aplicaciones**

**4.3 Creación de Inventarium**





## **5 Conclusiones y posibles mejoras**



# References

- Biot, M. A. (1962). Mechanics of deformation and acoustic propagation in porous media. *Journal of applied physics*, 33(4), 1482–1498.
- Heinz, M., Carsten, & Hoffmann, J. (2014, March). *The listings package, march 2014*. <http://texdoc.net/texmf-dist/doc/latex/listings/listings.pdf>. Retrieved 12/12/2014, from <http://texdoc.net/texmf-dist/doc/latex/listings/listings.pdf>