



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA Y SISTEMAS DE TELECOMUNICACIÓN

PROYECTO FIN DE GRADO

TÍTULO: Diseño e implementación de una Aplicación Web Full Stack para la organización y gestión de equipos.

AUTOR: Sergio Herrero Romero

TITULACIÓN: Telemática

TUTOR: Juan Manuel Meneses Chaus

DEPARTAMENTO: Ingeniería telemática y electrónica

VºBº

Miembros del Tribunal Calificador:

PRESIDENTE: César Díaz (IAC)

TUTOR: Juan Manuel Meneses (DTE)

SECRETARIO: Aurelio Berges (DTE)

Fecha de lectura: 29 Junio 2022

Calificación:

El Secretario,

Resumen

En los últimos años, el trabajo en equipo se ha convertido en una pieza angular para que cualquier proyecto fluya de forma óptima. Por ello, es indispensable que las empresas utilicen herramientas para organizar el trabajo en equipo. Con esta idea nació el software HERPO, una aplicación web capaz de solventar los problemas que puedan existir en la organización de un equipo de trabajo.

El objetivo principal de este proyecto es diseñar y desarrollar una aplicación web que consiga ahorrar tiempo, mejorar la organización y producción, crear competitividad dentro de un equipo de trabajo y ayudar a los jefes de equipo a tener una visión mas global de las tareas que se realizan dentro de su área de trabajo. Para ello, se ha necesitado un desarrollo full stack, que implica dos objetivos secundarios: por un lado la parte del servidor y por otro la interfaz que permitirá al usuario interactuar con la aplicación.

El resultado ha sido una aplicación web desarrollada con las tecnologías Angular 10+ en la parte Front end y Spring 5 con Java 8 en la parte Back end, tecnologías modernas para el desarrollo de aplicaciones web en la actualidad.

La aplicación permite realizar diferentes acciones, como son: crear, modificar y eliminar miembros de equipo; crear, modificar y eliminar tareas; asociar tareas a usuarios y añadir detalles y tiempos de entrega a las tareas

Abstract

In recent years, teamwork has become a cornerstone for any project to flow optimally. Therefore, it is essential that companies use tools to organize teamwork. With this idea, the HERPO software was born, a web application capable of solving the problems that may exist in the organization of a work team.

The main objective of this project is to design and develop a web application that manages to save time, improve organization and production, create competitiveness within a work team and help team leaders to have a more global vision of the tasks to be carried out. performed within their work area. For this, we have needed a full stack development, which implies two secondary objectives: on the one hand the server part and on the other the interface that will allow the user to interact with the application.

The result has been a web application developed with Angular 10+ technologies on the Front end and Spring 5 with Java 8 on the Back end, modern technologies for developing web applications today.

The application allows us to perform different actions, such as: create, modify and delete team members; create, modify and delete tasks; associate tasks with users and add details and lead times to tasks.

Lista de acrónimos

- POO: programación orientada a objetos.
- Npm: node package manager.
- RCP: Rich Client Plataform.
- SPA:Single page application.
- JDK: Java Development Kit.
- JRE: Java Runtime Environment.
- BBDD: base de datos.
- JAR: Java Archive.
- JWT: Java Web Token.
- WAR: Web Application Archive.

Índice de contenido

Resumen.....	5
Abstract.....	7
Lista de acrónimos	9
Índice de contenido.....	11
Lista de figuras.....	13
1 Introducción	17
2 Marco tecnológico	19
2.1 Angular 10+	19
2.2 Node.....	19
2.3 TypeScript.....	20
2.4 Atom	20
2.5 Bootstrap.....	21
2.6 Eclipse.....	21
2.7 SpringBoot.....	22
2.8 Api Rest	23
2.9 REST.....	23
2.10 JSON WEB TOKEN (JWT).....	24
3 Especificaciones y restricciones de diseño	27
4 Descripción de la solución propuesta	29
4.1 Iniciación en Angular.....	29
4.1.1 Crear nuestra aplicación con Angular.....	30
4.1.2 Integrar Bootstrap con Angular	31
4.2 Backend: Spring API REST.....	33
4.2.1 Crear proyecto Spring API REST.....	35
4.2.2 Configurar Datasource a MySQL	36
4.2.3 Instalación MySQL.....	37
4.2.4 Creación de la base de datos	42
4.3 Diagrama de clases de la aplicación Spring.....	42
4.3.1 Diagrama de clases del paquete Entity	42
4.3.2 Diagrama de clases del paquete Services.....	44
4.4 Modelo de datos.....	45

5 Resultados	47
5.1 Acceso modo invitado.....	47
5.2 Acceso modo usuario	48
5.3 Acceso modo administrador	51
6 Presupuesto	57
6.1 Recursos humanos	57
6.2 Recursos de desarrollo	57
6.3 Resumen del presupuesto.....	58
7 Conclusiones	59
7.1 Conclusiones principales	59
7.2 Trabajos futuros	60
8 Referencias.....	61

Lista de figuras

<i>Ilustración 1 - Token JWT</i>	25
<i>Ilustración 2 - Casos de uso</i>	28
<i>Ilustración 3 - Prototipo por defecto en entorno local</i>	31
<i>Ilustración 4 - package.json</i>	32
<i>Ilustración 5 - SpringUtils</i>	34
<i>Ilustración 6 - Nuevo proyecto Spring (I)</i>	35
<i>Ilustración 7 - Nuevo proyecto Spring (II)</i>	35
<i>Ilustración 8 - Instalación MySQL (I)</i>	38
<i>Ilustración 9 - Instalación MySQL (II)</i>	39
<i>Ilustración 10 - Instalación MySQL (III)</i>	40
<i>Ilustración 11 - Ventana inicio Workbench (I)</i>	40
<i>Ilustración 12 - Ventana inicio Workbench (II)</i>	41
<i>Ilustración 13 - Instancia MySQL</i>	41
<i>Ilustración 14 - Diagrama de clases paquete Entity</i>	43
<i>Ilustración 15 - Diagrama de clases paquete Services</i>	44
<i>Ilustración 16 - Modelo de datos</i>	45
<i>Ilustración 17 - Vista modo invitado</i>	47
<i>Ilustración 18 - Pantalla de inicio de sesión</i>	48
<i>Ilustración 19 - Modal acceso usuario correcto</i>	48
<i>Ilustración 20 - Modal acceso usuario erróneo</i>	48
<i>Ilustración 21 - Pantalla principal modo usuario</i>	49
<i>Ilustración 22 - Modal detalle usuario (modo usuario)</i>	50
<i>Ilustración 23 - Pantalla detalle tarea</i>	51
<i>Ilustración 24 - Pantalla principal modo administrador</i>	52
<i>Ilustración 25 - Pantalla creación nuevo miembro</i>	52
<i>Ilustración 26 - Modal nuevo miembro creado con éxito</i>	53
<i>Ilustración 27 - Nuevo miembro en la tabla</i>	53
<i>Ilustración 28 - Modal detalle miembro</i>	54
<i>Ilustración 29 - Modal imagen subida</i>	55
<i>Ilustración 30 - Pantalla nueva tarea</i>	55

<i>Ilustración 31 - Modal tarea creada con éxito</i>	<i>56</i>
<i>Ilustración 32 - Tabla recursos humanos</i>	<i>57</i>
<i>Ilustración 33 - Tabla recursos de desarrollo</i>	<i>57</i>
<i>Ilustración 34 - Tabla presupuesto total.....</i>	<i>58</i>

Diseño e implementación de una Aplicación Web Full Stack para la organización y gestión de equipos.

03 de Mayo de 2022

1 Introducción

Este proyecto ha sido creado con la idea de facilitar la labor que debe desempeñar un jefe de equipo o similar dentro de un equipo de trabajo. Se quiere ayudar en la difícil tarea que es organizar y gestionar un equipo, los cuales muchas veces no rinden al máximo debido a la escasa organización. El objetivo es que la persona capacitada para liderar un grupo pueda apoyarse en nuestra herramienta.

El líder del equipo podrá administrar su equipo en pocos pasos, creando miembros y asignándoles la tarea que él vea oportuna. Además, podrá comprobar en poco tiempo cuales son las tareas que están desarrollando cada uno de sus trabajadores y realizar análisis de cómo evoluciona el equipo.

Para poder dar una solución a esta problemática, se desarrollará una aplicación full stack que dotará a los administradores de un apoyo para organización y gestión de su equipo. Se contará con una interfaz de usuario muy intuitiva, que se sustentará bajo el apoyo de unos servicios que se comunican con la base de datos.

En este documento, se describe en primer lugar un marco tecnológico, en el cual se hará contexto a las tecnologías usadas en el proyecto. Posteriormente se definirán las especificaciones y restricciones de diseño. A continuación, se describir la solución propuesta. Por último, mostraremos los resultados, con las pruebas realizadas, obteniendo las conclusiones del proyecto

2 Marco tecnológico

2.1 Angular 10+

Angular es un framework¹ de diseño de aplicaciones y plataforma de desarrollo para crear aplicaciones de una sola página eficientes y sofisticadas. Ha sido desarrollado por Google, para crear aplicaciones web interactivas y amistosas para el usuario, es decir, con interfaz gráfica sencilla, reactiva, dinámica e instantánea. Es un framework de los más utilizados en el mercado y más popular en la actualidad en la industria para desarrollar este tipo de aplicaciones. En realidad, se programa sobre TypeScript, pero finalmente se transpila hacia JavaScript. Transpilar es un término relativamente nuevo, viene del inglés "transpiler", fruto de la unión de las palabras "translate" y "compiler". Realmente es más correcto decir que TypeScript transpila, puesto que lo cierto es que no hay una compilación, sino es una traducción de un código fuente en otro código fuente en un lenguaje diferente [1].

2.2 Node

Node.js es un entorno de ejecución de un solo hilo, de código abierto y multiplataforma para crear aplicaciones de red y del lado del servidor rápidas y escalables. Se ejecuta en el motor de ejecución de JavaScript V8, y utiliza una arquitectura de E/S basada en eventos y sin bloqueos, lo que la hace eficiente y adecuada para aplicaciones en tiempo real. Está programado en C, C++ y JavaScript [2].

Node.js utiliza la arquitectura «Single Threaded Event Loop» para manejar múltiples clientes al mismo tiempo. Para entender en qué se diferencia de otros tiempos de ejecución, tenemos que entender cómo se manejan los clientes concurrentes multihilo en lenguajes como Java.

¹ Un framework es una herramienta de desarrollo web que, por lo general, se define como una aplicación o conjunto de módulos que permiten el desarrollo ágil de aplicaciones mediante la aportación de librerías y/o funcionalidades ya creadas.

En un modelo de solicitud-respuesta multihilo, varios clientes envían una solicitud y el servidor procesa cada una de ellas antes de devolver la respuesta. Sin embargo, se utilizan múltiples hilos para procesar las llamadas concurrentes. Estos hilos se definen en un pool de hilos, y cada vez que llega una petición, se asigna un hilo individual para manejarla [3].

2.3 TypeScript

TypeScript se trata de un superconjunto de Javascript, es decir, amplía Javascript con una nueva sintaxis que añade, entre otras cosas, el tipado estático opcional (una vez que la variable se declara de un tipo, no se puede cambiar el tipo más adelante), genéricos, decoradores y elementos de POO (programación orientada a objetos) como interfaces o property accessors.

TypeScript compila código JavaScript que se ejecuta en cualquier navegador, host, sistema operativo o motor de JavaScript que admita ECMAScript 3. Cuando hablamos de ECMAScript, nos referimos al estándar que ya va por la versión ES6 y determina cómo emplear el lenguaje Javascript, que permite a los fabricantes de software desarrollar las herramientas adecuada para interpretarlo correctamente.

TypeScript trata de resolver la mayoría de los problemas con JavaScript centrándose en mejorar la experiencia y la productividad de los desarrolladores. Permite utilizar técnicas como el tipado estático opcional y/o la encapsulación para generar un código mucho más mantenible y escalable que con JavaScript tradicional, sin perder el carácter dinámico del mismo.

Además, estos dos últimos años la popularidad de TypeScript se ha disparado y se perfila como uno de los lenguajes de programación con más futuro [4].

2.4 Atom

Atom es un moderno editor de código y texto que además es hackeable, es decir, que puede ser personalizado. Ha sido creado por los usuarios de GitHub. Se caracteriza

porque puede instalarse en cualquier sistema operativo, incluye plugins² para descargar o crear los tuyos propios y tiene un entorno amigable, ya que el estilo de base por defecto es muy claro [5].

2.5 Bootstrap

Bootstrap es un framework CSS, un lenguaje informático que especifica cómo se presentan los documentos a los usuarios: cómo se diseñan, compaginan, etc. de código abierto que favorece el desarrollo web de un modo más sencillo y rápido. Incluye plantillas de diseño basadas en HTML y CSS con la que es posible modificar tipografías, formularios, botones, tablas, navegaciones, menús desplegables, etc. También existe la posibilidad de utilizar extensiones de Javascript adicionales. Fue desarrollado inicialmente por Twitter en 2011 y permite crear interfaces de usuario limpias y compatibles con todo tipo de dispositivos. Entre las ventajas que tiene Bootstrap es que favorece el design responsive, el cual se utiliza para mejorar la experiencia de los usuarios en el sitio web y en consecuencia el posicionamiento [6].

2.6 Eclipse

Eclipse es un entorno de desarrollo software multi-lenguaje construido alrededor de un workspace³ al que pueden incluirse un gran número de plugins que proporcionan funcionalidades concretas relacionadas con lenguajes específicos o con la interacción con otras herramientas implicadas en el desarrollo de una aplicación. Pese a ser un entorno multi-lenguaje, está desarrollado en Java, siendo el desarrollo en este lenguaje su aplicación principal [7].

Entre las principales características de Eclipse se encuentran:

- Herramienta de código abierto.

² Los plugins son pequeños programas complementarios que amplían las funciones de aplicaciones web y programas de escritorio.

³ Un workspace es un tipo de directorio que usa Eclipse para almacenar proyectos.

- Salvo el núcleo de la aplicación, todas las funcionalidades de Eclipse están desarrolladas como plugins.
- La funcionalidad del paquete básico puede ampliarse mediante la descarga de otros plugins, y además es posible desarrollar plugins personalizados.
- Proporciona una plataforma RCP para el desarrollo de aplicaciones de propósito general.
- Permite el desarrollo de aplicaciones para el servidor y servicios web.
- Entre sus plugins se encuentran funcionalidades para la interacción con repositorios⁴ de código compartido.

2.7 SpringBoot

Según el sitio oficial del proyecto, se puede definir Spring Boot de la siguiente manera: «Es una solución para crear aplicaciones basadas en Spring de una manera rápida, autónoma y con características deseables para producción».

Al momento, resulta de mucho interés contar con herramientas que apoyen al desarrollo rápido de aplicaciones. Y ésta es la motivación principal para presentar esta primera entrada sobre el tema [8].

Entre las principales características de Spring Boot se encuentran:

- Contenedores Java embebidos: Tomcat o Jetty. Es un programa capaz de recibir peticiones de páginas web y redireccionar estas peticiones a un objeto Servlet. Un Servlet es un objeto java que pertenece a una clase que extiende `javax.servlet.http.HttpServlet`. Esto no es del todo exacto ya que existen diferentes tipos de Servlets pero con diferencia `HttpServlet` es el más usado.
- Soporte para la automatización con Maven y Gradle (sistema de automatización de construcción de código de software).
- Configuración sugerida para iniciar rápidamente con un proyecto (Starters).
- Configura automáticamente Spring, cuando sea posible.
- Características listas para producción: métricas, seguridad, verificación del estatus, externalización de configuración, etc.
- No genera código y no requiere configuración XML.

⁴ Un repositorio de software es un lugar de almacenamiento del cual pueden ser recuperados e instalados los paquetes de software en un ordenador.

2.8 Api Rest

Una API de REST, o API de RESTful, es una interfaz de programación de aplicaciones (API o API web) que se ajusta a los límites de la arquitectura REST y permite la interacción con los servicios web de RESTful. El informático Roy Fielding es el creador de la transferencia de estado representacional (REST).

Las API son conjuntos de definiciones y protocolos que se utilizan para diseñar e integrar el software de las aplicaciones. Suele considerarse como el contrato entre el proveedor de información y el usuario, donde se establece el contenido que se necesita por parte del consumidor (la llamada) y el que requiere el productor (la respuesta) [9].

En otras palabras, las API nos permiten interactuar con una computadora o un sistema para obtener datos o ejecutar una función, de manera que el sistema comprenda la solicitud y la cumpla.

2.9 REST

REST no es un protocolo ni un estándar, sino más bien un conjunto de límites de arquitectura. Los desarrolladores de las API pueden implementarlo de distintas maneras.

Cuando el cliente envía una solicitud a través de una API de RESTful, esta transfiere una representación del estado del recurso requerido a quien lo haya solicitado o al extremo. La información se entrega por medio de HTTP en uno de estos formatos: JSON (JavaScript Object Notation), HTML, XLT, Python, PHP o texto sin formato. JSON es el lenguaje de programación más popular, ya que tanto las máquinas como las personas lo pueden comprender y no depende de ningún lenguaje, a pesar de que su nombre indique lo contrario [10].

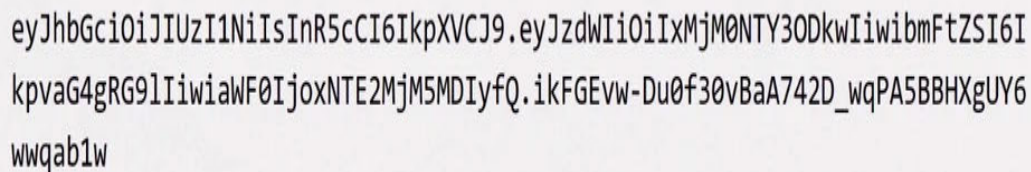
2.10 JSON WEB TOKEN (JWT)

JWT (JSON Web Token) es un estándar que está dentro del documento RFC 7519, donde se definen las especificaciones de objetos JSON, el cual, es un formato de texto sencillo para el intercambio de datos.

En el mismo se define un mecanismo para poder propagar entre dos partes, y de forma segura, la identidad de un determinado usuario, además con una serie de claims o privilegios.

Estos privilegios están codificados en objetos de tipo JSON, que se incrustan dentro del payload o cuerpo de un mensaje que va firmado digitalmente [11].

En la práctica, se trata de una cadena de texto o token que tiene tres partes codificadas en Base64, cada una de ellas separadas por un punto, que llevan el contenido de un mensaje como la que vemos en la imagen siguiente:



```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MzIyMDIyLmV5LWVw-Du0f30vBaA742D_wqPA5BBHXgUY6wwqab1w
```

Podemos utilizar un debugger online para decodificar ese token y visualizar su contenido. Si accedemos al mismo y pegamos dentro el texto completo, se nos mostrará lo que contiene:

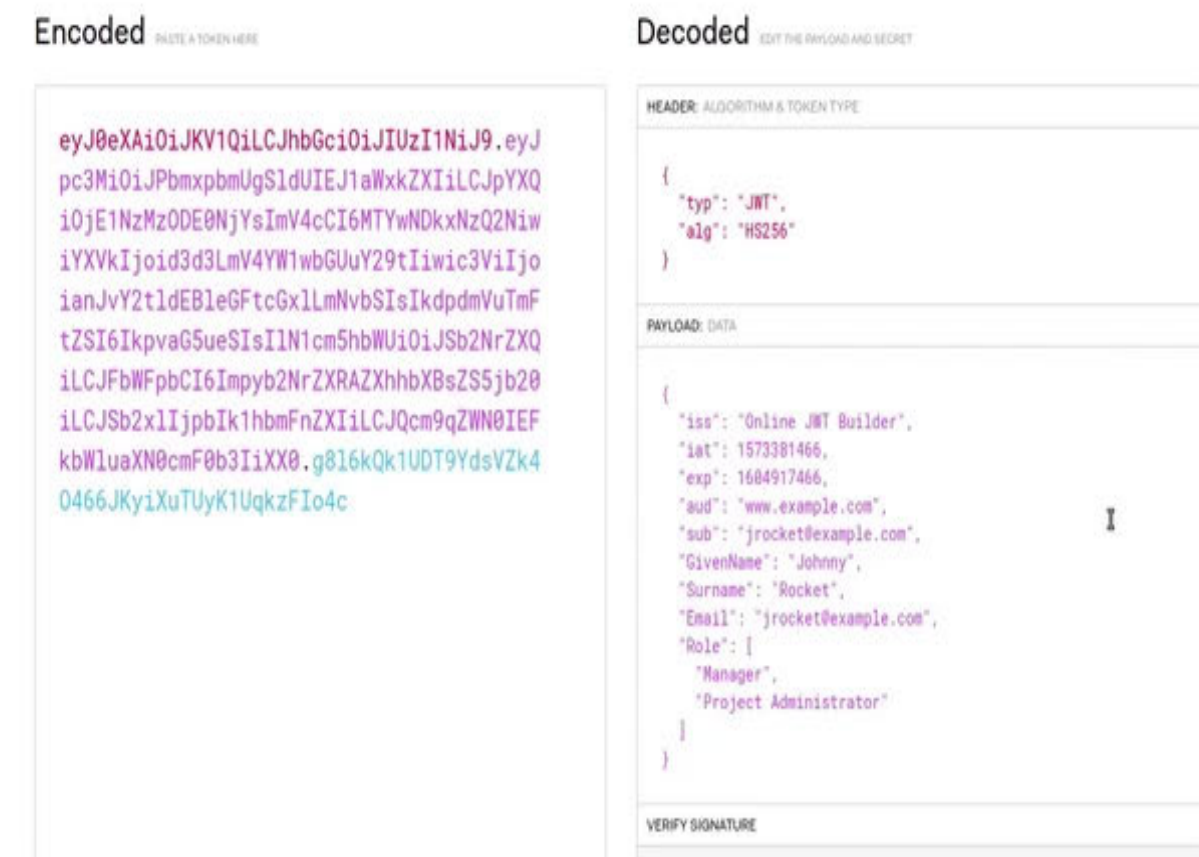


Ilustración 1 - Token JWT

Podemos ver el contenido del token sin necesidad de saber la clave con la cual se ha generado, aunque no podremos validarlo sin la misma.

Como hemos dicho, un token se divide en tres partes:

- **Header:** encabezado dónde se indica, al menos, el algoritmo y el tipo de token, que en el caso del ejemplo anterior era el algoritmo HS256 y un token JWT.
- **Payload:** donde aparecen los datos de usuario y privilegios, así como toda la información que queramos añadir, todos los datos que creamos convenientes.
- **Signature:** una firma que nos permite verificar si el token es válido, y aquí es donde radica el quid de la cuestión, ya que si estamos tratando de hacer una comunicación segura entre partes y hemos visto que podemos coger cualquier token y ver su contenido con una herramienta sencilla, ¿dónde reside entonces la potencia de todo esto?

La firma se construye de tal forma que vamos a poder verificar que el remitente es quien dice ser, y que el mensaje no se ha modificado por el camino [12].

Se construye como el HMACSHA256, que son las siglas de Hash-Based Message Authentication Code (Código de Autenticación de Mensajes), cifrado además con el algoritmo SHA de 256 bits. Se aplica esa función a:

- Codificación en Base64 de header.
- Codificación en Base64 de payload.
- Un secreto, establecido por la aplicación.

De esta forma, si alguien modifica el token por el camino, por ejemplo, inyectando alguna credencial o algún dato malicioso, entonces podríamos verificar que la comprobación de la firma no es correcta, por lo que no podemos confiar en el token recibido y deberíamos denegar la solicitud de recursos que nos haya realizado, ya sea para obtener datos o modificarlos.

Si lo que estamos requiriendo es que el usuario esté autenticado, deberíamos denegar esa petición, por lo que siempre que trabajemos con JWT deberíamos verificar con esa firma si el token es válido o no lo es.

3 Especificaciones y restricciones de diseño

Entre las finalidades principales de este proyecto se encuentra el objetivo principal de crear una herramienta a partir de la cual, los líderes de proyectos puedan llevar a cabo un seguimiento personalizado de su equipo de trabajo, además, cada trabajador podrá fijarse objetivos propios en la realización de cada tarea

Para la realización del desarrollo de esta aplicación web se han tenido en cuenta una serie de requisitos que se exponen a lo largo de este apartado.

- Queríamos que la herramienta fuera muy visual, intuitiva y fácil de manejar.
- Podemos acceder a ella desde cualquier parte del planeta, solo necesitamos conexión a Internet.
- La aplicación presentará tres tipos de perfiles de usuario: modo invitado, usuario y administrador.
- El perfil modo invitado solo tendrá acceso a la visualización de la tabla principal, donde se visualiza información de cada usuario y la tarea que tiene asignada.
- Usuario, es el perfil de cada miembro del equipo, tiene acceso al detalle de la tarea que tiene asignada donde podrá modificar aspectos relacionados con ella, como el tiempo empleado, observaciones o estado e la tarea
- Administrador, es el súper usuario, creado para ser asignado al jefe de equipo o de proyecto, su labor consiste desde crear y eliminar miembros de equipo, crear y asignar tareas a cada usuario y editar las mismas, para conseguir la supervisión global del equipo.

A continuación, se muestra el diagrama de casos de uso general de interacción de usuarios con el sistema.

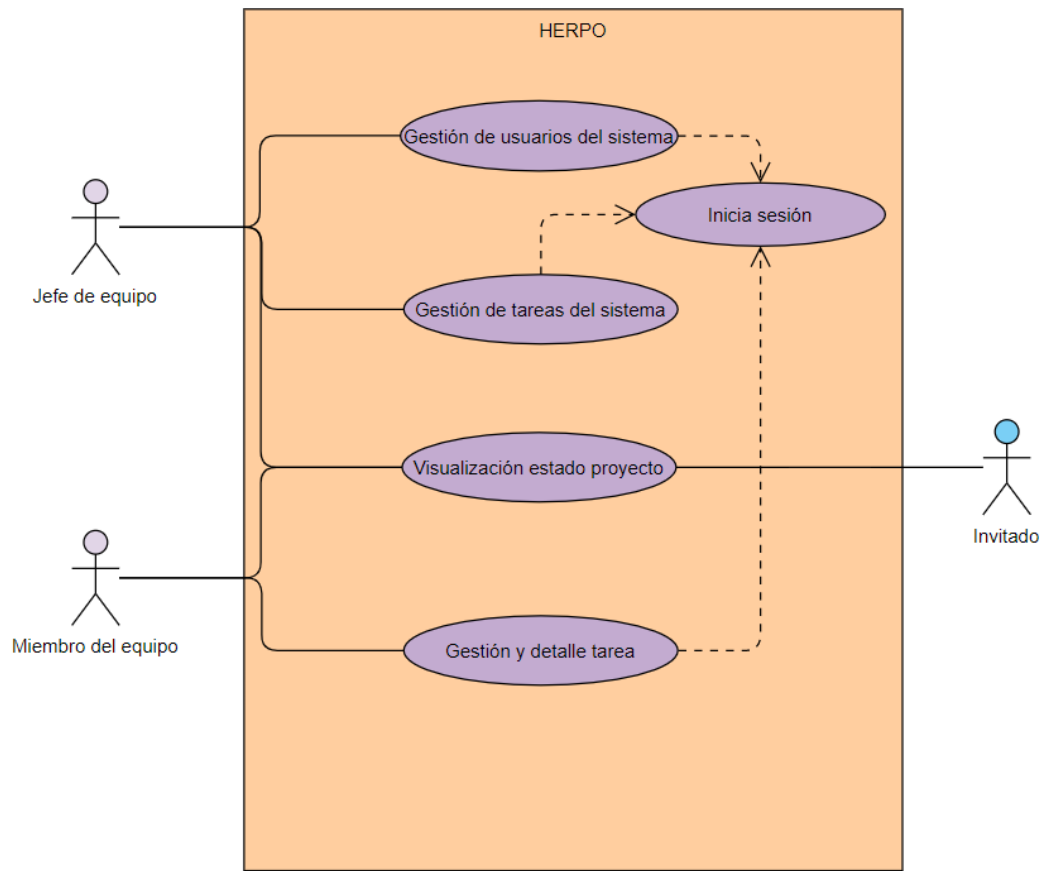


Ilustración 2 - Casos de uso

4 Descripción de la solución propuesta

La finalidad principal es crear una aplicación web, con una interfaz de usuario interactiva y amigable para facilitar la gestión a los líderes de proyecto, con método de autenticación, manejo de errores provocados por los usuarios y acceso a base de datos.

A continuación, iremos detallando como vamos a conseguir desarrollar esta aplicación.

4.1 Iniciación en Angular

Angular es un framework de Javascript para crear aplicaciones web interactivas, amigable, reactivas y dinámicas. Pero es mucho más que eso es una tecnología muy bien posicionada dentro del sector tecnológico, ya que esta desarrollado por Google. Una de las principales características de Angular es que crea aplicaciones web responsivas, es decir, que se pueden ejecutar desde cualquier dispositivo, ya sea un PC, Tablet o dispositivos móviles. Otra característica muy importante, que es la esencia de Angular es que es SPA, es decir, aplicación de una sola página. Al final, construimos una gran aplicación con muchos componentes, pero el usuario siempre va a estar interactuando dentro de la misma página del navegador, el contenido se actualiza de forma asíncrona.

Para comenzar el desarrollo, necesitamos instalar el software y herramientas necesarias. Lo primero que debemos instalar es Node.js, siempre descargaremos la última versión LTS, ya que es una versión estable.

Continuamos instalando TypeScript, aunque en principio no debería ser necesario, ya que está embebido en la descarga de Angular, pero es recomendable puesto que nos ayuda a detectar todos los errores en tiempo de escritura cuando escribamos el código en nuestro editor.

TypeScript está basado en Javascript y ha sido desarrollado como un súper conjunto de todas las evoluciones de Javascript. Es fuertemente tipado, es un tipado estático, tiene visibilidad en los atributos de las clases, podemos trabajar con clases, herencia,

interfaces, es decir, incluye todas las características importantes de la programación orientada a objetos.

Para su descarga, debemos obtener de su página web el siguiente comando: *npm install -g typescript*. Utilizamos npm, es decir, utilizamos node para instalar estos paquetes. Además, la sentencia *-g*, nos indica que lo vamos a instalar de manera global en nuestro PC, por lo que necesitamos permisos de administrador.

El siguiente paso es instalar el gestor de Angular, Angular CLI de la página oficial.

Como prerrequisito nos pide tener instalado tanto Node.js como npm. Para instalar Angular ejecutamos el comando: *npm install -g @angular/cli*.

Como editor, trabajaremos con Atom, ya que es muy potente para trabajar con Angular. Lo descargamos de <https://atom.io/>.

4.1.1 Crear nuestra aplicación con Angular

En este punto, ya podemos crear nuestra aplicación con Angular desde cero utilizando Angular CLI.

Para ello, debemos ejecutar el comando *ng new (nombre del proyecto)*. Es importante crear dentro de un directorio propio de nuestro proyecto, para tener la aplicación ordenada.

Una vez ejecutado este comando, nuestra aplicación ha sido creada y podemos levantar el servidor con el comando: *ng serve -o*. Angular incluye un servidor embebido para nuestra aplicación. Además, la instrucción *-o* nos abrirá una pestaña en nuestro navegador conectándonos al localhost. (Véase la ilustración 3).

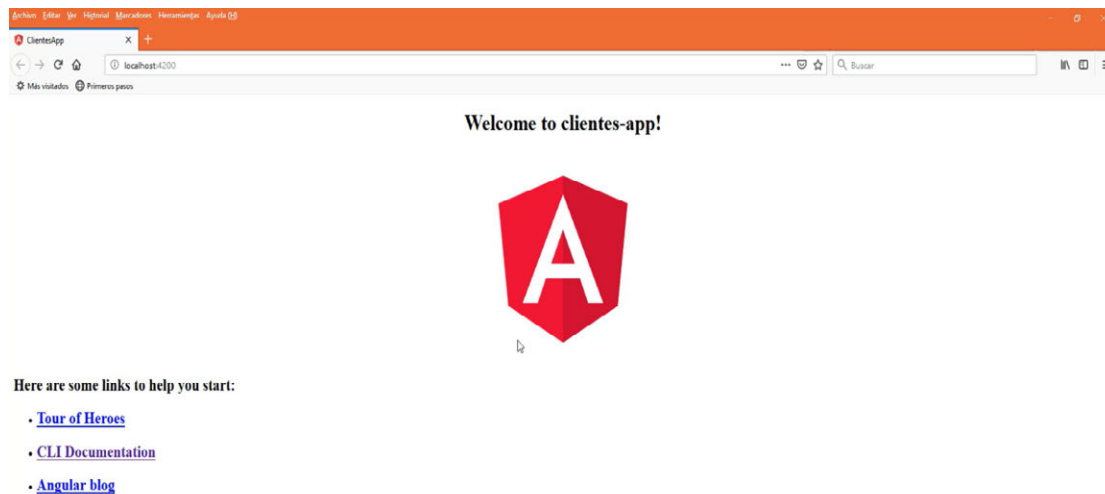


Ilustración 3 - Prototipo por defecto en entorno local

4.1.2 Integrar Bootstrap con Angular

A continuación, debemos instalar Bootstrap para conseguir una aplicación mucho más robusta y con una interfaz mucho más clara. Para ello, abrimos la consola e introducimos el comando *npm install Bootstrap*.

Para confirmar que se ha instalado correctamente abrimos el archivo *package.json* y comprobamos que se ha añadido la línea con la versión de Bootstrap.

Por otro lado, es recomendable la instalación de *jquery* y *Popper.js* para la correcta visualización de los menús, con el comando *npm install jquery Popper.js*.

Comprobamos en el fichero *package.json* que se han añadido todas las librerías. (Véase la ilustración 4).

```
"dependencies": {
  "@angular/animations": "^11.2.7",
  "@angular/cdk": "^7.0.0",
  "@angular/common": "^11.2.7",
  "@angular/core": "^11.2.7",
  "@angular/forms": "^11.2.7",
  "@angular/http": "^7.0.0",
  "@angular/material": "^7.0.0",
  "@angular/material-moment-adapter": "^7.3.7",
  "@angular/platform-browser": "^11.2.7",
  "@angular/platform-browser-dynamic": "^11.2.7",
  "@angular/router": "^11.2.7",
  "bootstrap": "^4.1.3",
  "core-js": "^2.5.7",
  "jquery": "^3.3.1",
  "moment": "^2.29.1",
  "popper.js": "^1.14.3",
  "rxjs": "^6.3.3",
  "sweetalert2": "^7.26.9",
  "zone.js": "^0.11.4"
```

Ilustración 4 - package.json

4.2 Backend: Spring API REST

Spring es una de las tecnologías más importantes del desarrollo en Java, pues permite crear aplicaciones de forma muy rápida y con una auto configuración sorprendente. Para el desarrollo completo de la parte backend necesitamos algunas herramientas que nos permitan llevar a cabo nuestro objetivo.

En primer lugar, necesitamos el motor del lenguaje Java JDK, que contiene todas las librerías, herramientas, clases, entre otros, necesarios para trabajar con el lenguaje Java. Cuando lo descargemos este paquete va a incluir la parte de desarrollo y testing, pero también la parte de ejecución (JDK + JRE). Para la descarga visitamos la página oficial de Oracle y seleccionamos la versión más actualizada del JDK en nuestro sistema operativo, en este caso Windows

(<https://www.oracle.com/java/technologies/downloads/#jdk17-windows>).

Continuamos con eclipse, herramienta fundamental para escribir nuestro código. Es el entorno de desarrollo integrado para crear nuestras aplicaciones en Java, en este caso con Spring, nuestro Backend (<https://www.eclipse.org/>). Es importante en este desarrollo, la instalación del plugin SpringTools, que nos permite crear los proyectos con Spring Boot, que nos facilitará la configuración de forma automática, administrando las librerías y dependencias. Para instalarlo, accedemos al market place de Eclipse, donde encontramos todos los plugins disponibles y buscamos SpringUtil, en este caso instalaremos la versión 4. (Véase la ilustración 5). Además, tiene embebido un servidor dentro del proyecto, lo cual nos ahorra tener que instalar un Tomcat o cualquier otro servidor.

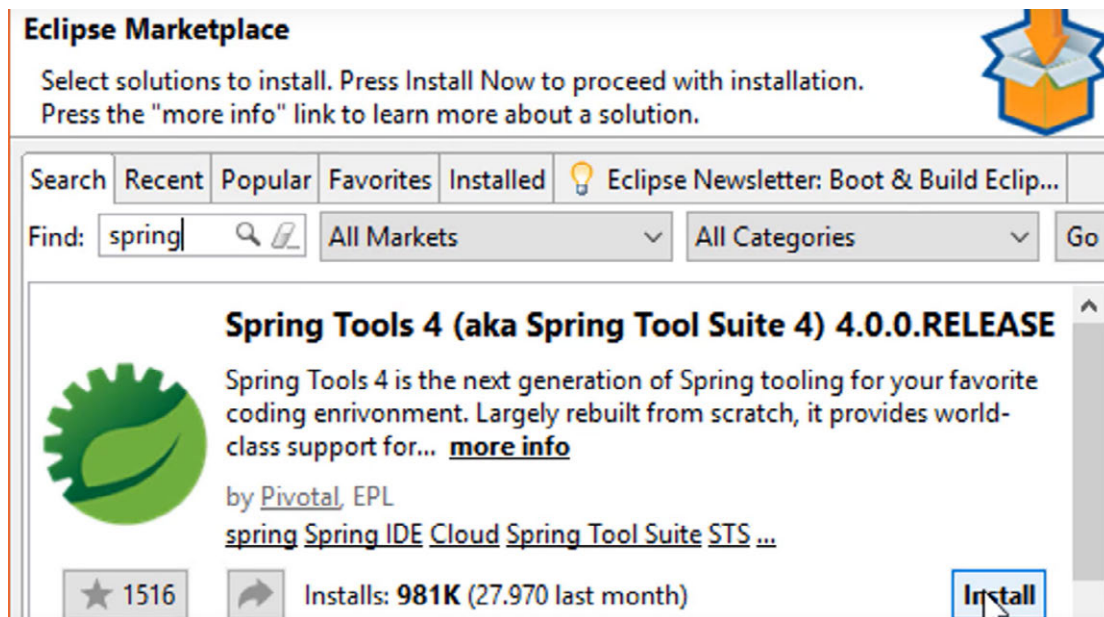


Ilustración 5 - SpringUtils

Para la generación de proyectos, utilizamos Maven, que se encarga de la gestión de dependencias, pero en este caso Maven, está configurado dentro de la instalación de Eclipse.

Como motor de base de datos utilizaremos MySQL. Para la instalación navegamos a la página oficial de MySQL (<https://dev.mysql.com/downloads/mysql/>), utilizaremos la versión community.

Para la realización de las pruebas de la API REST utilizaremos Postman, una aplicación opensource que nos permite enviar peticiones de cualquier tipo, ya sea verbo o método (GET, PUT, POST o DELETE). Nos ayuda a comprobar si nuestros servicios funcionan correctamente o si son suficientemente rápidos, con lo cual podremos optimizar nuestra API con la ayuda de esta aplicación. Para la descarga, navegamos a la página oficial de Postman (<https://www.postman.com/downloads/>)

4.2.1 Crear proyecto Spring API REST

En este momento, ya estamos preparados para crear nuestro proyecto backend. Para ello, nos apoyaremos en las herramientas que vienen preinstaladas en nuestro IDE Eclipse.

En primer lugar, crearemos un nuevo proyecto en Eclipse. (Véase la ilustración 6 y 7).

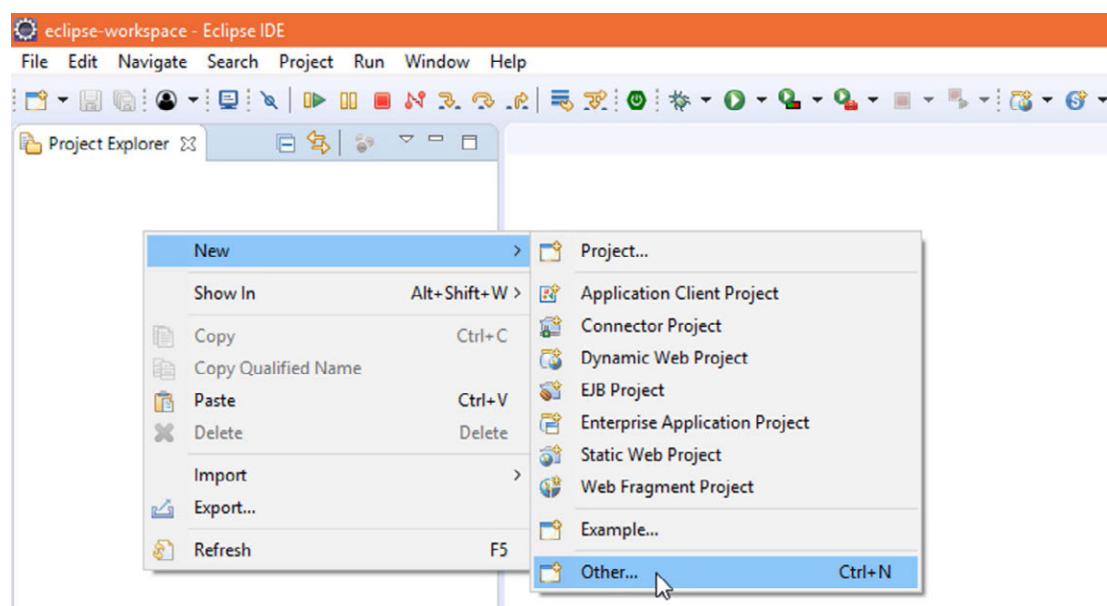


Ilustración 6 - Nuevo proyecto Spring (I)

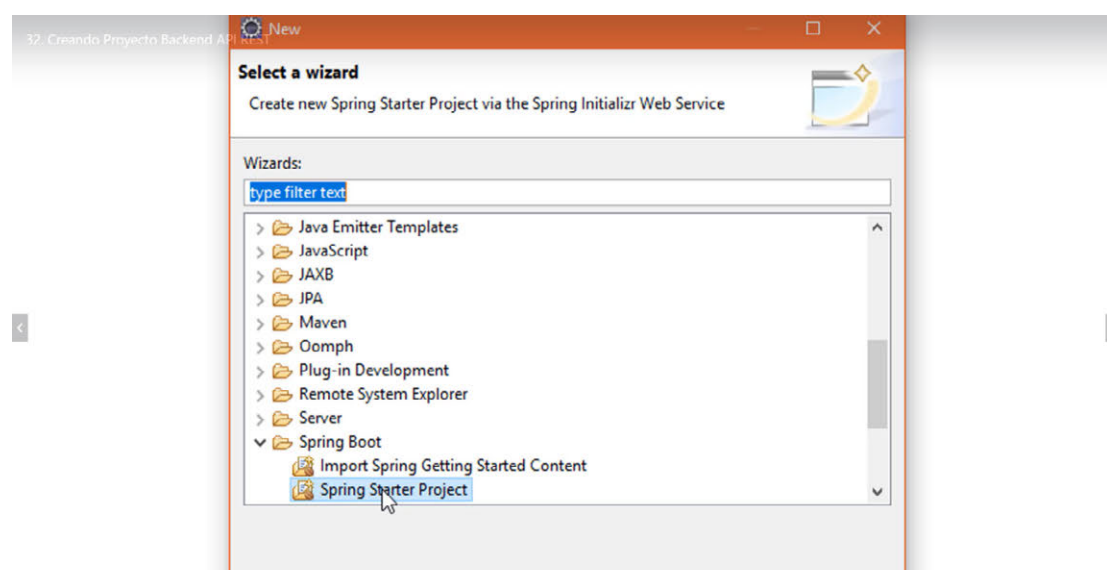


Ilustración 7 - Nuevo proyecto Spring (II)

A continuación, nos pedirá información relevante para la creación de nuestro proyecto, en este caso utilizaremos un proyecto tipo Maven, pero podríamos haber utilizado otra tecnología como es Gradle, la versión de Java será la 10 y el tipo de empaquetado será jar ya que estamos utilizando una aplicación Api Rest y Spring Boot ya tiene embebido un servidor, por lo que los despliegues nos lo permite hacer en formato jar. Si utilizamos servidores externos como Tomcat seleccionaríamos war para los despliegues.

Pulsando en siguiente llegaremos a la ventana de dependencias o librerías. En primer lugar, seleccionaremos la última versión de Spring Boot estable, es decir que no sea una SNAPSHOT ya que estas versiones están en desarrollo. En segunda instancia, seleccionaremos las dependencias que vamos a utilizar en nuestro proyecto.

Añadiremos la dependencia web, una de las más importante, ya que incluye, el API y anotaciones para crear nuestros controladores RESTfull, por ejemplo la anotación Controller, que nos ayudará a realizar diferentes operaciones, la dependencia JPA de SQL ya que trabajaremos con el API de persistencia de JAVA, a través de hibernate a través de anotaciones como Entity para la relación con bbdd, la dependencia MySQL, nuestro motor de bbdd, la dependencia DevTools de Core, nos configura que cualquier cambio que hagamos en el código de nuestras clases de lógica de negocio se actualice y se despliegue automáticamente en nuestro servidor, ahorrándonos mucho tiempo.

Finalizamos y nuestro proyecto ya habrá sido creado.

4.2.2 Configurar Datasource a MySQL

El Datasource hace referencia al conjunto de conexiones a base de datos (pool de conexiones) creadas por el servidor que se reservan durante un determinado tiempo al componente que lo requiera.

Para configurarlo, abrimos el fichero application.properties y añadimos las siguientes líneas.

```
spring.datasource.url=jdbc:mysql://localhost/db_springboot_backend?useSSL=false  
spring.datasource.username=root
```

```
spring.datasource.password=sasa  
spring.datasource.driver-class-name=com.mysql.jdbc.Driver  
spring.jpa.database-platform=org.hibernate.dialect.MySQL5InnoDBDialect  
spring.jpa.hibernate.ddl-auto=create-drop  
logging.level.org.hibernate.SQL=debug
```

4.2.3 Instalación MySQL

MySQL es un sistema de gestión de bases de datos relacionales (RDBMS) de código abierto respaldado por Oracle y basado en el lenguaje de consulta estructurado (SQL). MySQL funciona prácticamente en todas las plataformas, incluyendo Linux, UNIX y Windows.

En nuestro caso, procederemos a instalar MySQL 8, a través del siguiente enlace:

<https://dev.mysql.com/downloads/mysql/>.

Una vez que hayamos elegido la versión correcta para nuestro sistema operativo, abriremos el archivo de instalación y seguiremos la siguiente guía:

En primer lugar, elegimos una instalación personalizada (Custom), ya que vamos a seleccionar los productos a instalar manualmente. (Véase la ilustración 8).

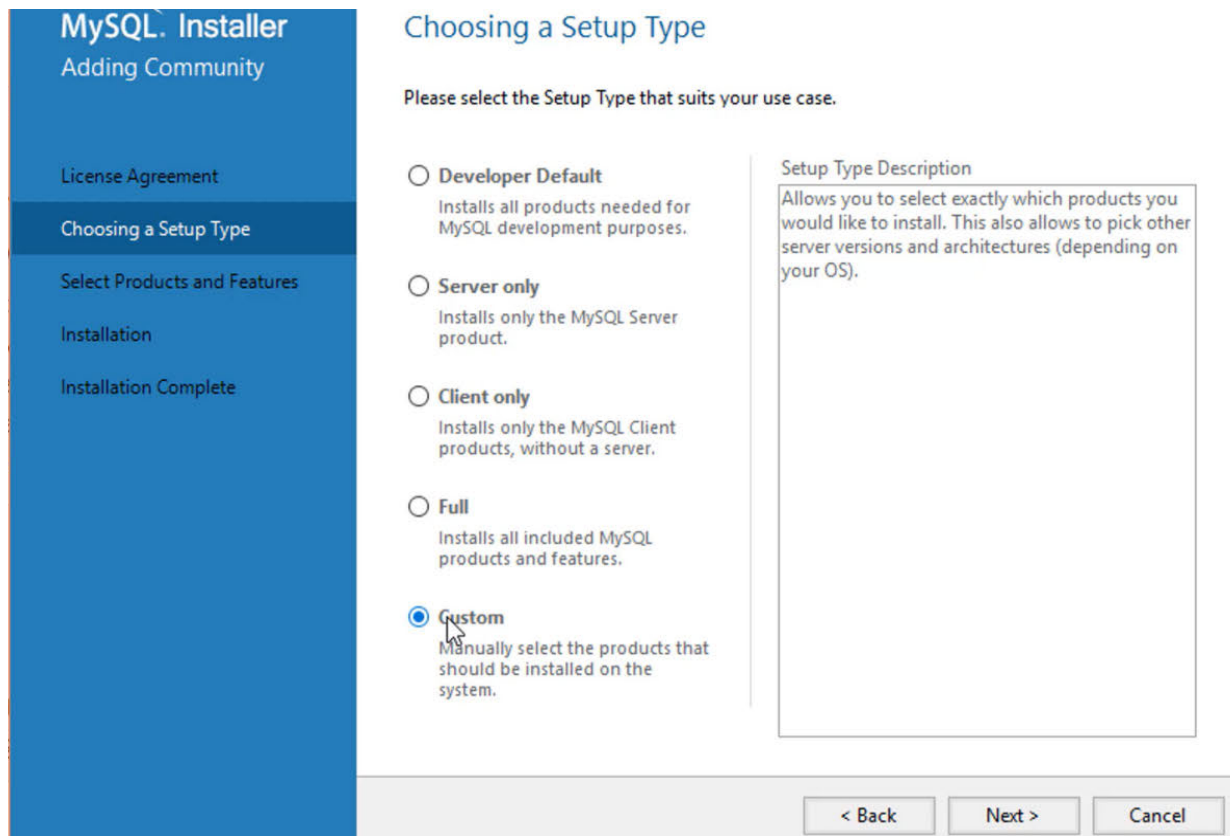


Ilustración 8 - Instalación MySQL (I)

En la siguiente pantalla, elegimos los productos ofrecidos, en nuestro caso seleccionaremos los detallados en la imagen.

El más importante es el producto MySQL Server, ya que es el servidor.

También añadimos MySQL Workbench, una herramienta con interfaz gráfica para la administración de la base de datos, ya que nos permite crear la base de datos, esquemas, tablas y nos ayudará en la gestión principal.

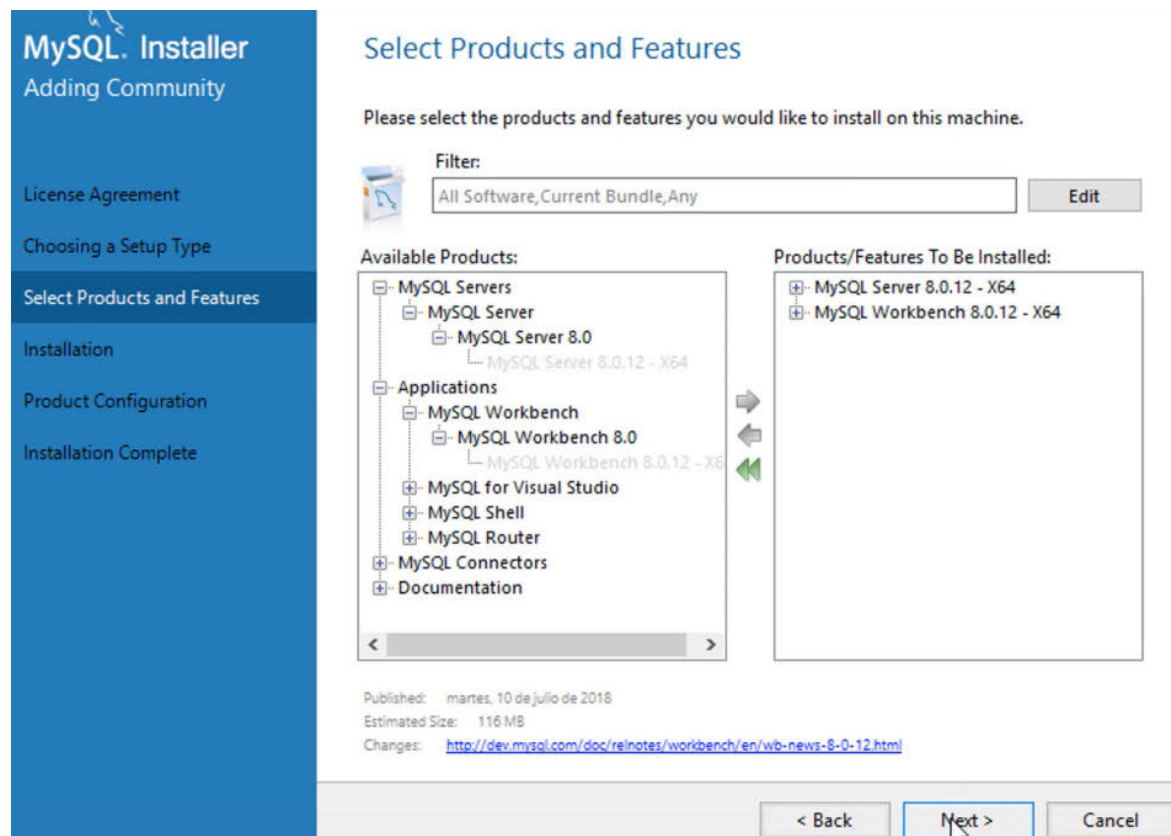


Ilustración 9 - Instalación MySQL (II)

Finalizamos la instalación pulsando en siguiente dejando todo por defecto.

Llegaremos a la nueva ventana donde nos pide insertar la cuenta y roles de la base de datos. Como hemos dicho anteriormente y configurado en el archivo application.properties, el usuario administrador será 'root' y la contraseña la hemos definido como 'sasa'.

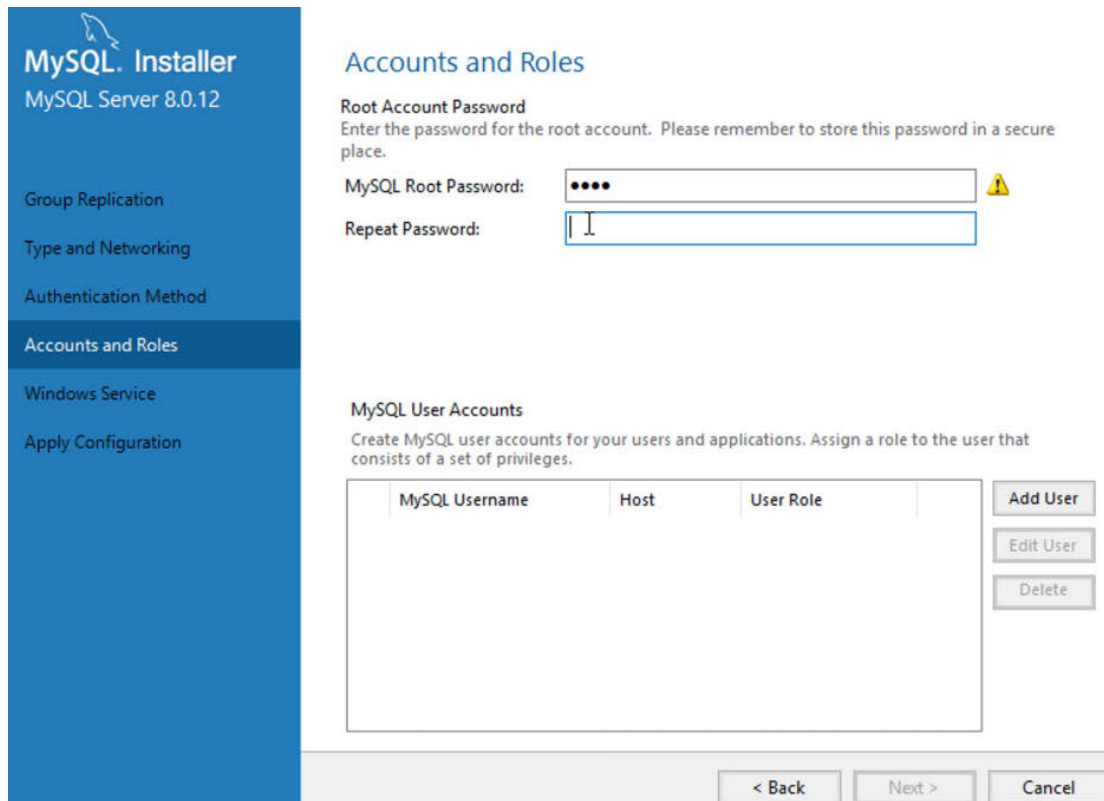


Ilustración 10 - Instalación MySQL (III)

Seguimos el flujo de la instalación y la finalizamos marcando la casilla para que arranque el programa Workbench una vez finalizada la instalación.

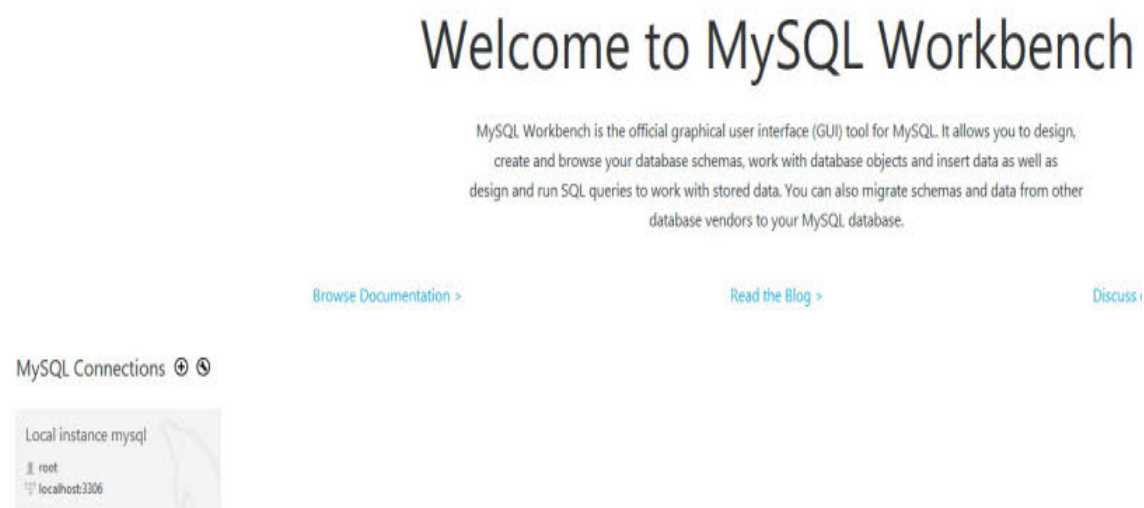


Ilustración 11 - Ventana inicio Workbench (I)

Hacemos click sobre la conexión MySQL que hemos creado en el proceso de instalación.

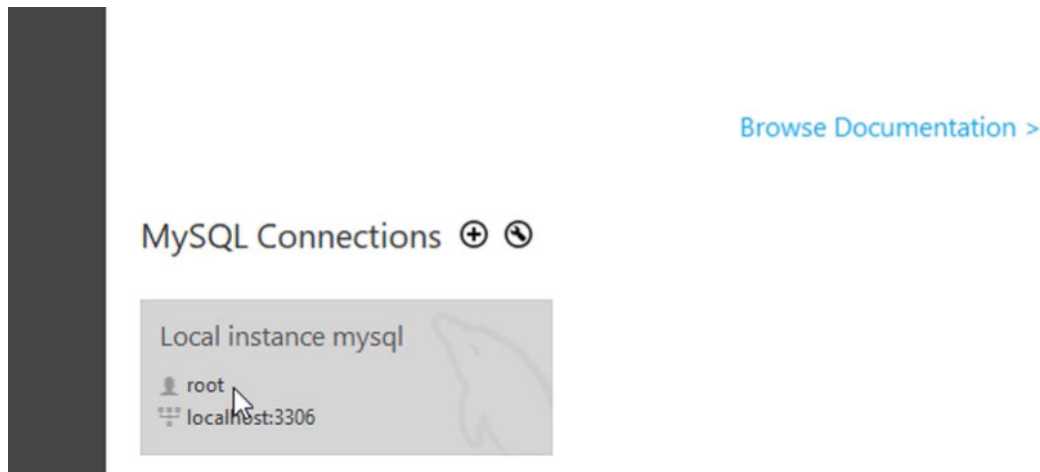


Ilustración 12 - Ventana inicio Workbench (II)

A continuación, nos pedirá que introduzcamos la clave del usuario root y accederemos a la administración de nuestra base de datos.

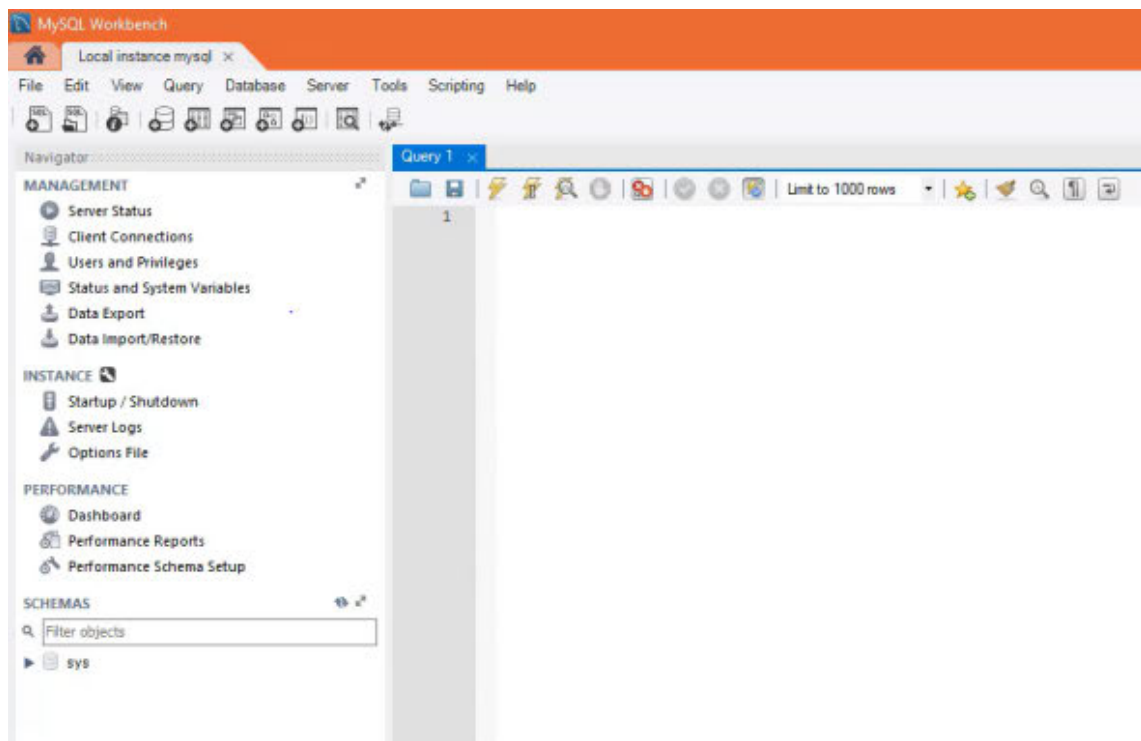


Ilustración 13 - Instancia MySQL

En esta pantalla podemos realizar todas las gestiones de nuestra base de datos, ver su estado, iniciarla, apagarla, obtener métricas, leer el log, etc.

4.2.4 Creación de la base de datos

Una vez configurada la base de datos en el fichero `application.properties`, en la cual definimos la url de la base de datos y su nombre en la propiedad `'spring.datasource.url=jdbc:mysql://localhost/db_springboot_backend'`.

Abrimos el terminal e introducimos el comando `'mysql -u root -p'`, 'u' de usuario y 'p' de password. Nos pregunta por el password que en nuestro caso es 'sasa', para conectarnos.

Una vez conectados a MySQL nos pedirá el nombre de nuestra base de datos, a la que vamos a llamar `'db_springboot_backend'`. Verificamos que se haya creado con el comando `'show databases'` y deberá aparecer nuestra base de datos en la tabla.

4.3 Diagrama de clases de la aplicación Spring

A continuación, se muestran los diagramas de clases de esta plataforma que resultan más importantes para la comprensión de la aplicación. Los diagramas de clases completos se encuentran en la carpeta `uml` del proyecto incluido en el CD adjunto a este documento.

4.3.1 Diagrama de clases del paquete Entity

En la paquetería Entity nos encontramos las siguientes clases que son clases simples, donde definimos los objetos de Tarea, Usuario, Cliente y Role:



Ilustración 14 - Diagrama de clases paquete Entity

4.3.2 Diagrama de clases del paquete Services

En la paquetería Services encontramos las clases definidas para la lógica de negocio:

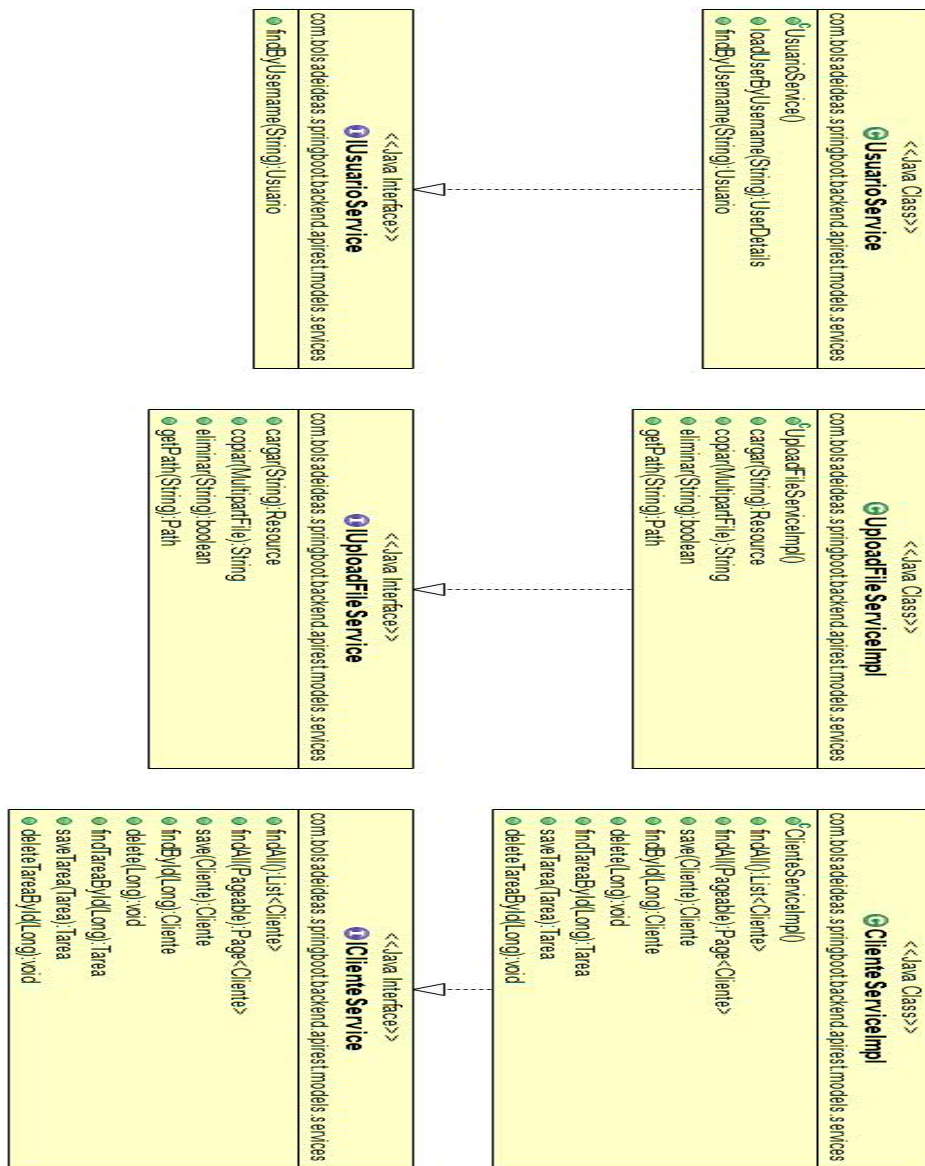


Ilustración 15 - Diagrama de clases paquete Services

4.4 Modelo de datos

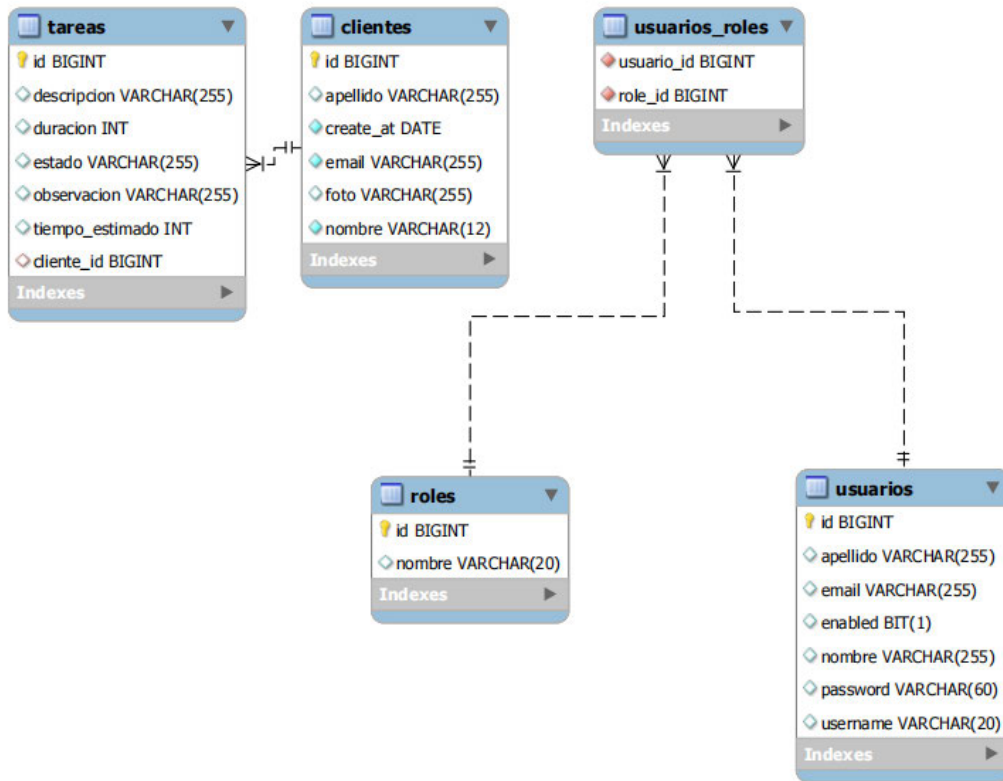


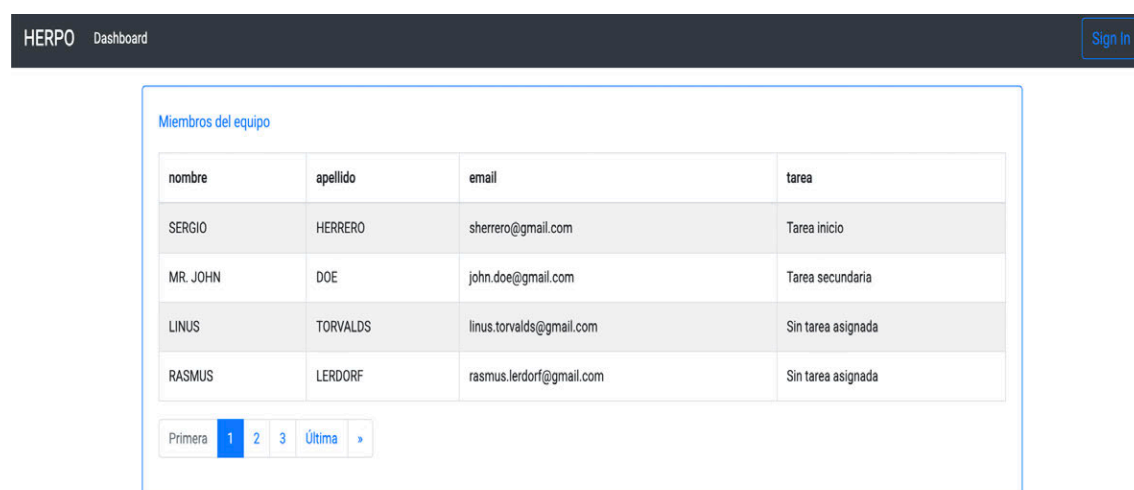
Ilustración 16 - Modelo de datos

5 Resultados

Como ya hemos comentado anteriormente, se ha dividido el acceso a recursos en 3 niveles de permisos: modo invitado, modo usuario y modo administrador.

5.1 Acceso modo invitado

Este modo de acceso es aquel el cual no necesita autenticación y cualquier actor puede visualizar. La primera vista que nos encontramos al acceder a la aplicación es la siguiente:



The screenshot shows the HERPO Dashboard interface. At the top left, it says 'HERPO Dashboard' and at the top right, there is a 'Sign In' button. The main content area is titled 'Miembros del equipo' and contains a table with the following data:

nombre	apellido	email	tarea
SERGIO	HERRERO	sherrero@gmail.com	Tarea inicio
MR. JOHN	DOE	john.doe@gmail.com	Tarea secundaria
LINUS	TORVALDS	linus.torvalds@gmail.com	Sin tarea asignada
RASMUS	LERDORF	rasmus.lerdorf@gmail.com	Sin tarea asignada

Below the table is a pagination control with buttons for 'Primera', '1', '2', '3', 'Última', and a right arrow.

Ilustración 17 - Vista modo invitado

En este modo solo podemos visualizar la tabla general de miembros del equipo con los datos más relevantes de cada uno de ellos. Podemos navegar por cada página, pero no podremos interactuar con ningún elemento de la tabla. El paginador es interactivo y se actualiza dependiendo de la página donde nos encontremos. Además, siempre se mostrará un botón para viajar a la primera y última página y otro botón de siguiente página. Todo ello hace que nuestra interfaz sea muy manejable y usable para cualquier tipo de usuario.

Pulsando en el botón 'Sign in' podremos autenticarnos y la pantalla que mostramos aparece a continuación, donde encontraremos un espacio para el introducir el

‘Username’ o nombre de usuario, un campo para ‘Password’ o contraseña y el botón de login.



Por favor Sign In!

Username

Password

Login

Ilustración 18 - Pantalla de inicio de sesión

5.2 Acceso modo usuario

Una vez que hemos iniciado sesión con un usuario, nos aparece un modal en la pantalla dándonos la bienvenida si el usuario y contraseña es correcto o un modal informándonos que el acceso no está autorizado.

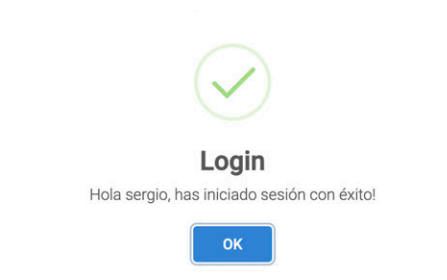


Ilustración 19 - Modal acceso usuario correcto

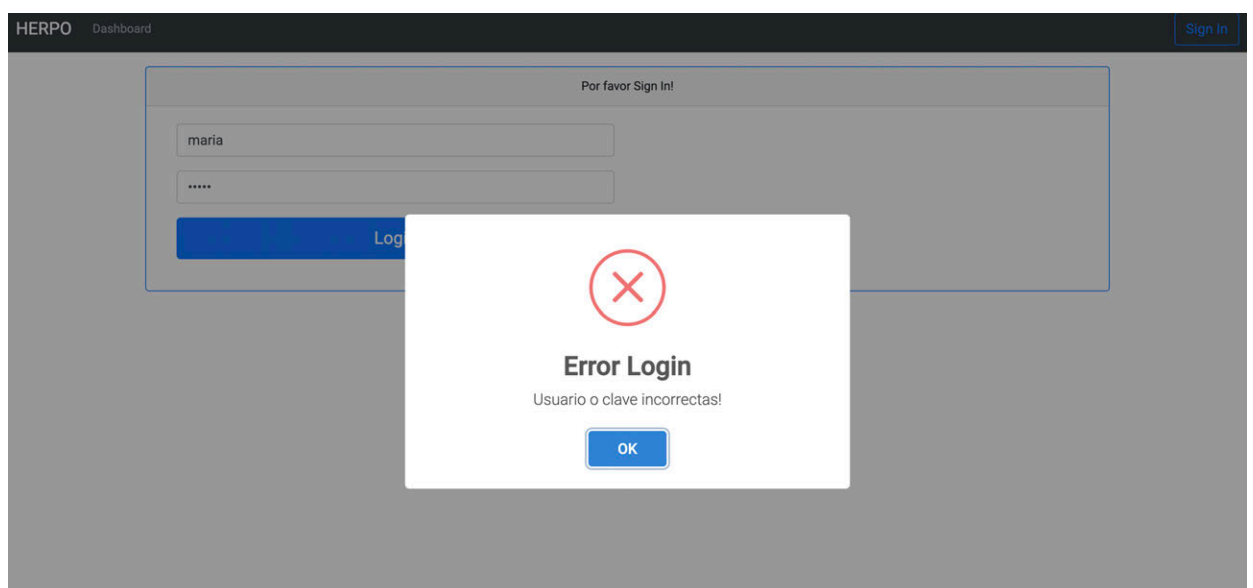
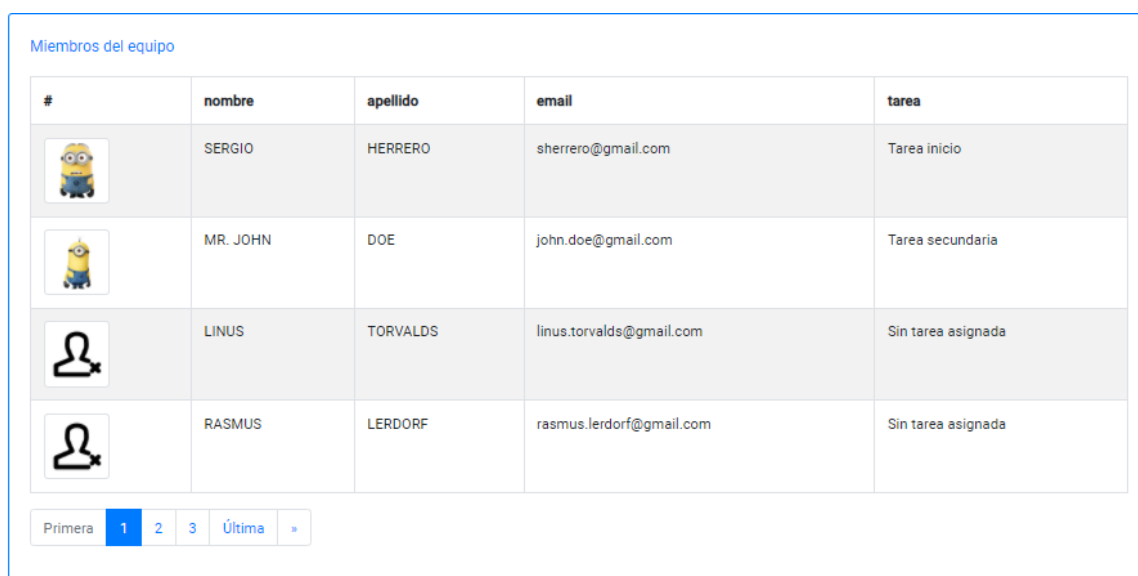






Ilustración 20 - Modal acceso usuario erróneo

Cómo se puede observar, se ha diseñado la aplicación siempre pensando en la fácil usabilidad y orientada a ser lo más informativa posible, ya que, además de mostrar mensajes informativos de manera recurrente, este tipo de modales se resaltan sobre la pantalla de fondo, añadiéndole un tono oscuro a la pantalla trasera para resaltar el modal y dotarle de gran relevancia.

Una vez iniciada sesión de manera satisfactoria con un usuario, se verá la siguiente pantalla:



#	nombre	apellido	email	tarea
	SERGIO	HERRERO	sherrero@gmail.com	Tarea inicio
	MR. JOHN	DOE	john.doe@gmail.com	Tarea secundaria
	LINUS	TORVALDS	linus.torvalds@gmail.com	Sin tarea asignada
	RASMUS	LERDORF	rasmus.lerdorf@gmail.com	Sin tarea asignada

Primera 1 2 3 Última »

Ilustración 21 - Pantalla principal modo usuario

En esta pantalla se observa la lista de miembros del equipo con una breve descripción de cada uno de ellos y la tarea que le ha sido asignada.

Se puede apreciar que algunos usuarios tienen avatar. Los que aún no han configurado su avatar tendrán un avatar por defecto, que es el que tiene el usuario Linus Torvalds, por ejemplo.

Si se pulsa sobre el avatar de uno de los usuarios, se accederá a una pantalla dónde hay una descripción más detallada del mismo:

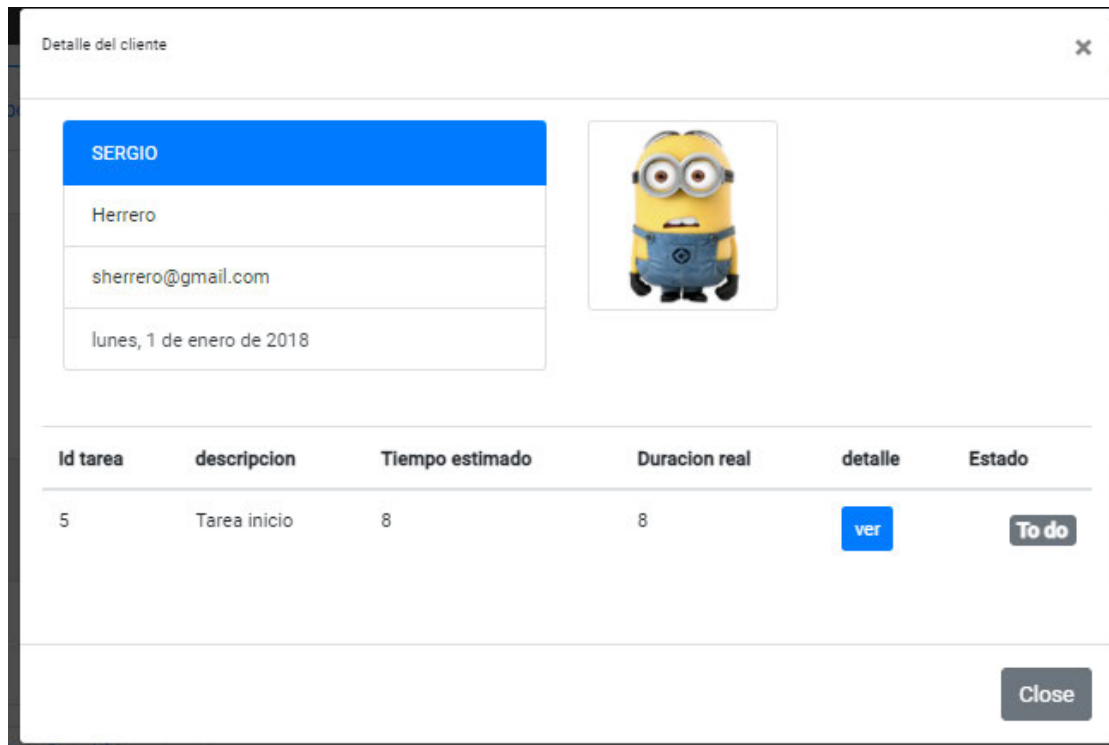


Ilustración 22 - Modal detalle usuario (modo usuario)

En este modal, se observa el nombre del usuario, su correo electrónico, fecha de creación del usuario, una tabla con la tarea que tiene asignada y un detalle sobre ella.

Además, podemos ver en esta tabla lo siguiente:

- Id de la tarea.
- Descripción de la tarea.
- Tiempo estimado por el jefe de equipo para la realización de la tarea.
- La duración real de la misma que indica el tiempo que le ha llevado al miembro del equipo a realizar esta tarea.
- El detalle dónde encontramos un botón 'ver' que nos hará navegar a otra pantalla que se explicará más adelante.
- El estado de la tarea que puede ser:
 - o TO DO: tarea pendiente de realizar.
 - o EN PROCESO: tarea que se está realizando en ese momento.
 - o HECHO: tarea terminada.

Al pulsar sobre el botón ver, se navegará a esta pantalla:

Tarea: Tarea inicio

« volver

Datos del miembro

Sergio

Herrero

Datos de la tarea

Id: 1

Descripción: Tarea inicio

Duración (horas):

8

Estado de la tarea:

To do

Tiempo estimado de la tarea (horas): 8

Observaciones

Añade algún comentario

Estado de la tarea: To do

Guardar

Ilustración 23 - Pantalla detalle tarea

En esta página, se observa una descripción más detallada de la tarea asignada a un miembro del equipo, pudiendo modificar la duración de la tarea, el estado y añadir unas observaciones si fuesen necesarias.





5.3 Acceso modo administrador

Este modo de acceso es el que tiene los privilegios de superusuario. En este caso, el administrador es el jefe de equipo, el cual va a poder administrar la aplicación por completo.

En primer lugar, el jefe de equipo va a visualizar una tabla mucho más detallada, tiene el poder de crear, eliminar y editar nuevos usuarios y nuevas tareas.

Miembros del equipo

[Crear miembro](#)

#	nombre	apellido	email	tarea	Asignar tarea	editar	eliminar
	SERGIO	HERRERO	sherrero@gmail.com	Tarea inicio	Asignar tarea	editar	eliminar
	MR. JOHN	DOE	john.doe@gmail.com	Tarea secundaria	Asignar tarea	editar	eliminar
	LINUS	TORVALDS	linus.torvalds@gmail.com	Sin tarea asignada	Asignar tarea	editar	eliminar
	RASMUS	LERDORF	rasmus.lerdorf@gmail.com	Sin tarea asignada	Asignar tarea	editar	eliminar

Primera [1](#) [2](#) [3](#) [Última](#) »

Ilustración 24 - Pantalla principal modo administrador


Pulsando sobre el botón ‘Crear Miembro’, se navegará a la siguiente pantalla:

Crear Cliente

Nombre

Apellido

Email

Fecha 

[Crear](#) [Volver](#)

Ilustración 25 - Pantalla creación nuevo miembro

En esta pantalla se definen los datos de un nuevo miembro en el equipo con un pequeño formulario. Una vez que se ha creado el usuario, se lanzará un modal en la pantalla:

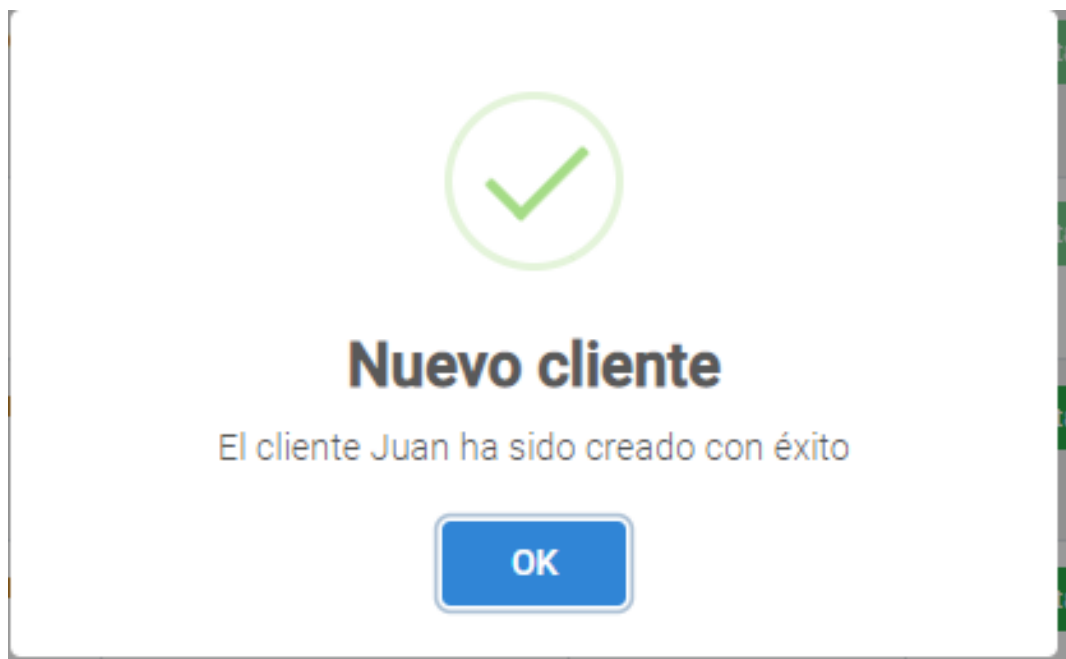



Ilustración 26 - Modal nuevo miembro creado con éxito

Tras esto, la tabla de miembros del equipo se actualizará automáticamente, y se añadirá el nuevo miembro:

Miembros del equipo

Crear miembro

#	nombre	apellido	email	tarea	Asignar tarea	editar	eliminar
	JUAN	GÓMEZ	juangomez@gmail.com	Sin tarea asignada	Asignar tarea	editar	eliminar

« Primera 1 2 3 4 Última

Ilustración 27 - Nuevo miembro en la tabla

Terminada la operativa de creación de nuevo usuario, se podrá editar y añadir un nuevo avatar pulsando sobre su imagen:

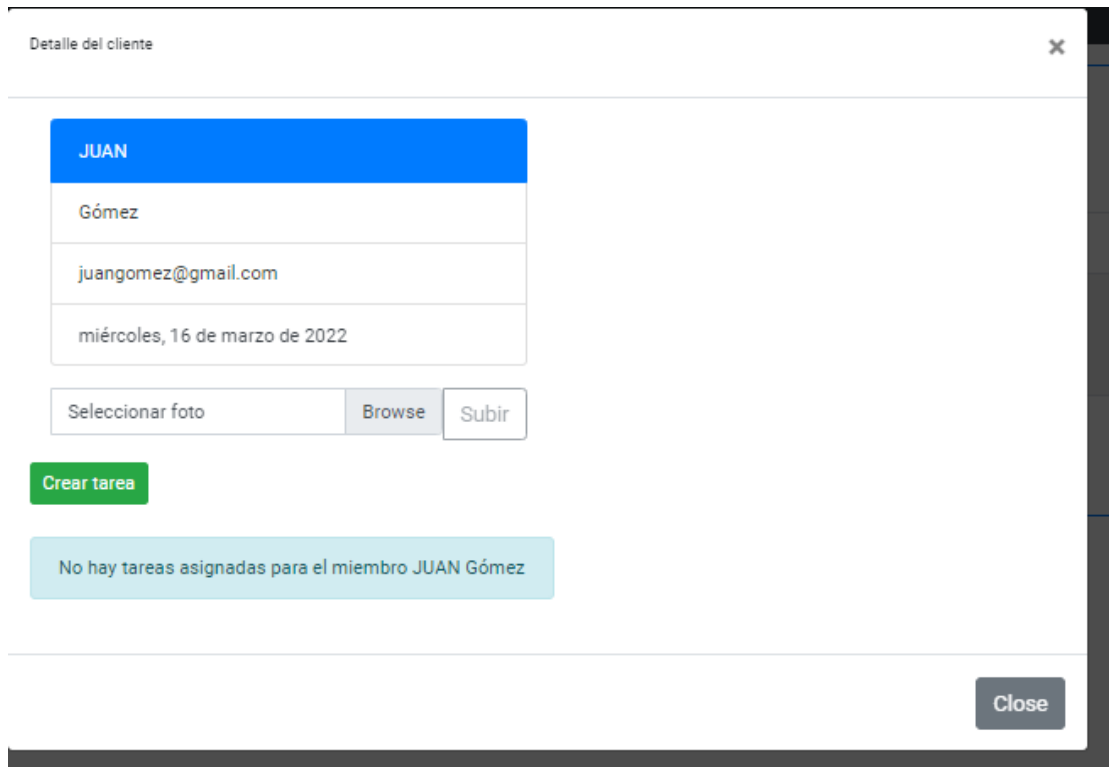


Ilustración 28 - Modal detalle miembro

En este modal, se muestra información del usuario, pero en este caso, podemos subir una imagen para nuestro avatar, se pulsa en 'browse' y se abrirá el explorador de carpetas de nuestro sistema operativo, seleccionamos la imagen que deseemos y se pulsará sobre 'subir':

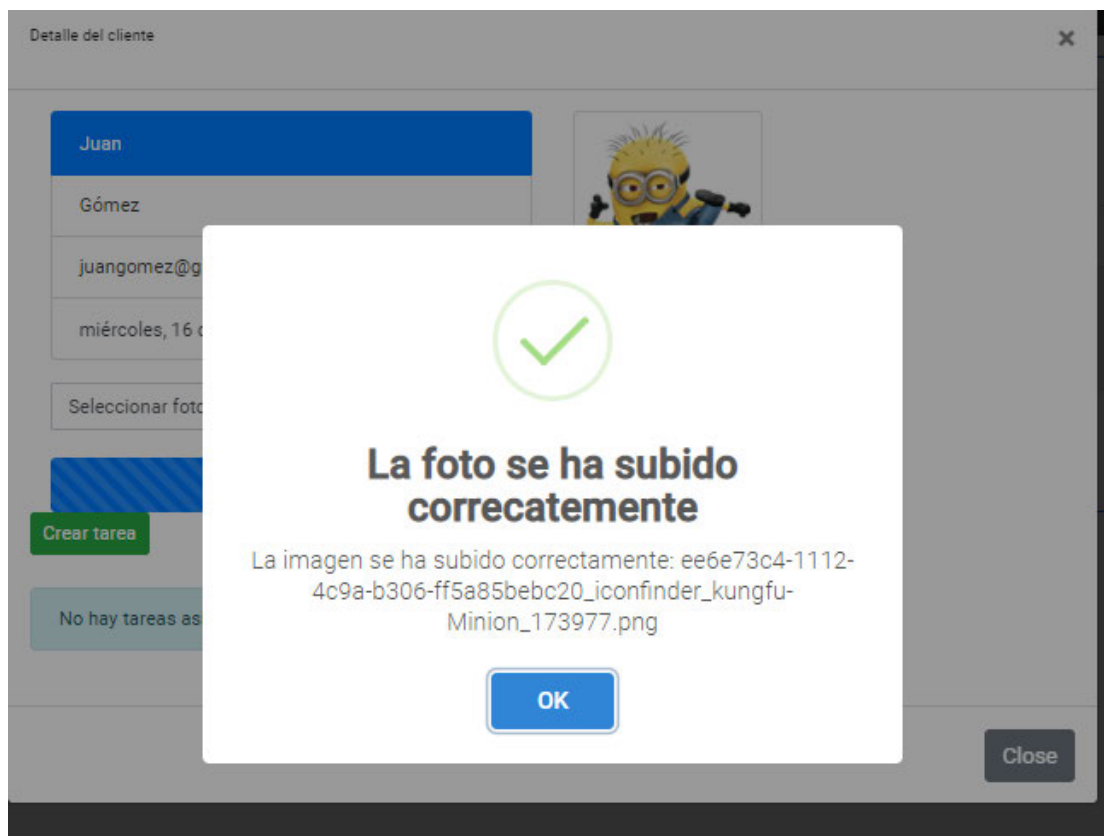


Ilustración 29 - Modal imagen subida

Volviendo a la pantalla de detalle del miembro, se podrá crear una nueva tarea, que será asignada directamente a ese nuevo miembro del equipo, al pulsar sobre el botón 'Crear tarea', se abrirá una nueva pantalla:

A screenshot of a web application form titled 'Crear una tarea' (Create a task). The form has a '« volver' (Back) button at the top left. It contains four input fields: 'Cliente' with the value 'Juan Gómez', 'Descripción' with 'Tarea Nueva', 'Observación' with 'Detalles de la tarea', and 'Tiempo estimado' with '2'. A 'Crear tarea' (Create task) button is located at the bottom left of the form.

Ilustración 30 - Pantalla nueva tarea

En esta pantalla, el jefe de equipo podrá mediante el formulario dar información sobre la nueva tarea que se debe realizar. Como siempre, cuando la tarea se ha creado correctamente, se abre un modal de confirmación:

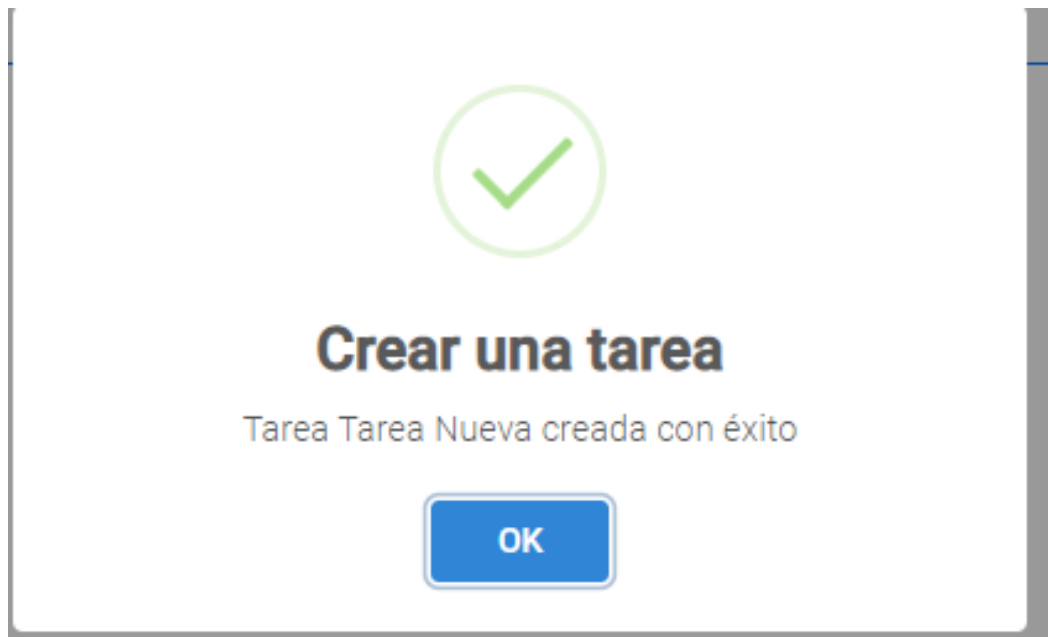


Ilustración 31 - Modal tarea creada con éxito

Una vez creada, el usuario puede ver su nueva tarea asignada y poder trabajar sobre ella como hemos comentado anteriormente.

6 Presupuesto

En esta sección del documento se recoge el presupuesto del proyecto que se ha llevado a cabo de acuerdo con todo lo especificado anteriormente.

6.1 Recursos humanos

La ilustración muestra el presupuesto del proyecto para los recursos humanos.

Categoría	Número de personas	Horas	Precio unitario	Precio total
Analista programador	1	250	14,00€	3.500,00€

Ilustración 32 - Tabla recursos humanos

6.2 Recursos de desarrollo

Descripción	Unidades	Precio unitario	Amortización	Importe amortizado	Importe total
PC de desarrollo	1	800,00€	10%	80,00€	80,00€

Ilustración 33 - Tabla recursos de desarrollo

Se ha añadido una columna para la amortización, la cual se ha calculado teniendo en cuenta la vida útil de los recursos, que suele ser de aproximadamente cuatro años.

Como el proyecto ha durado un semestre, la amortización ha de ser aproximadamente de un 10%.

6.3 Resumen del presupuesto

Nombre de la categoría	Importe
Recursos humanos	3.500,00€
Recursos de desarrollo	80,00€
Total	3.580,00€
IVA (21%)	751,80€
Total con IVA	4.331,80€

Ilustración 34 - Tabla presupuesto total

El precio del proyecto es de 4.331,80€ con el IVA incluido.

7 Conclusiones

El propósito de este capítulo es proporcionar una visión general del Proyecto de Fin de Grado, simplificando el problema propuesto, evaluando la solución implementada y proporcionar las conclusiones extraídas del resultando final, atendiendo a las dificultades encontradas durante el desarrollo del proyecto y los logros alcanzados. Finalmente, se proponen unas líneas de trabajo futuras que permiten darle continuidad al proyecto para dotarle de una funcionalidad completa.

7.1 Conclusiones principales

El objetivo principal de este proyecto fin de grado es proporcionar una herramienta para la organización y gestión de equipos de trabajo, destinada a los jefes de equipo que tienen la necesidad de que todos sus trabajadores puedan mejorar su rendimiento.

La solución propuesta para llevar a cabo este objetivo, es la creación de una aplicación web en la que se desarrolla una programa para la gestión de tareas. Como entorno de desarrollo se ha definido una aplicación Java con Spring en la parte del servidor y Angular 10 para la definición de la vista.

El resultado final, como se describe en el apartado ‘5 Resultados’ es una aplicación web desarrollada para la organización de equipos en un proyecto laboral determinado o incluso para cualquier tipo de gestión de equipos, en la cual, podremos añadir miembros de trabajo y asignarles tareas. Lo que llega a hacer realmente interesante este producto, es que tiene la capacidad de incentivar las labores de cada miembro, pudiendo marcarle tiempo de trabajo a cada tarea.

Por lo tanto, se concluye que se han conseguido todos los objetivos establecidos como punto de partida en este proyecto añadiendo un grado de valor funcional gracias al tratamiento de las necesidades de usuario y la corrección de los errores cometidos durante el desarrollo.

La aplicación se encuentra en producción y se puede visitar en la siguiente URL:
<https://tareas-app-angular.web.app/clientes>

7.2 Trabajos futuros

Este proyecto cumple con los requisitos establecidos en las fases de definición del objetivo y análisis de requisitos y su viabilidad, no obstante, se ofrecen a continuación unas vías de trabajo futuras que incrementarían el valor laboral de la aplicación.

En primer lugar, se propone como primera línea de trabajo futuro añadir nuevas funcionalidades que mejoren la experiencia del usuario, como añadir líneas de trabajo organizadas en pequeños tramos de tiempo, para después poder sacar estadísticas que ayuden a mejorar la productividad. Estas estadísticas, podrán visualizarse mediante la descarga de un fichero Excel que muestre las tareas realizadas por cada miembro en el periodo de tiempo que se estime oportuno.

Por último, para mejorar la realización de tareas y la resolución de estas, se propone añadir un chat en tiempo real para la comunicación entre miembros del equipo.

8 Referencias

- [1] «Angular,» [En línea]. Available: <https://angular.io/docs>. [Último acceso: 01 05 2022].
- [2] «Nodejs,» [En línea]. Available: <https://nodejs.org/es/docs/>. [Último acceso: 01 05 2022].
- [3] «Kinsta,» [En línea]. Available: <https://kinsta.com/es/base-de-conocimiento>. [Último acceso: 01 05 2022].
- [4] «Softwarecrafters,» [En línea]. Available: <https://softwarecrafters.io/typescript/typescript-tutorial-javascript-introduccion>. [Último acceso: 01 05 2022].
- [5] «Atom,» [En línea]. Available: <https://atom.io/docs>. [Último acceso: 01 05 2022].
- [6] «Bootstrap,» [En línea]. Available: <https://getbootstrap.com/docs/5.1/getting-started/introduction/>. [Último acceso: 01 05 2022].
- [7] «Eclipse,» [En línea]. Available: <https://www.eclipse.org/documentation/>. [Último acceso: 01 05 2022].
- [8] «SpringBoot,» [En línea]. Available: <https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/>. [Último acceso: 01 05 2022].
- [9] «Api Rest,» [En línea]. Available: <https://www.ibm.com/docs/en/inventory-visibility?topic=apis-rest-api-documentation>. [Último acceso: 01 05 2022].
- [10] «REST,» [En línea]. Available: <https://idratherbewriting.com/learnapidoc/#:~:text=%E2%80%9CREST%E2%80%9D%20stands%20for%20Representational%20State%20Transfer.&text=I%20dive%20more%20into%20the,sample%20responses%20from%20the%20endpoints..> [Último acceso: 01 05 2022].
- [11] «JWT,» [En línea]. Available: <https://jwt.io/introduction>. [Último acceso: 01 05 2022].
- [12] «Webinars,» [En línea]. Available: <https://openwebinars.net/blog/que-es-json-web-token-y-como-funciona/>. [Último acceso: 01 05 2022].