**PROYECTO FIN DE GRADO**

ServiRep



|  |  |
| --- | --- |
|  | I.E.S Jacarandá  Desarrollo de Aplicaciones Web  Raúl Naranjo Abril |

ÍNDICE

1. Presentación del proyecto

1.1. Introducción

1.2. Objetivos

1.3. Tecnologías y herramientas empleadas

2. Diseño del sistema.

2.1. Arquitectura del sistema.

2.2. Requisitos del sistema.

2.3. Definición de interfaces de usuario.

2.4. Mock ups.

2.5. Planificación temporal.

2.6. Especificación del plan de pruebas.

3. Desarrollo del sistema.

3.1. Descripción del entorno de desarrollo.

3.2. Desarrollo.

3.3. Documentación.

4. Despliegue e implantación.

4.1. Implantación del sistema y pruebas.

4.1.1. Instalación de servidores y aplicaciones.

4.1.2. Pruebas de implantación.

4.2. Formación.

4.2.1. Manual de usuario.

5. Mantenimiento.

6. Conclusiones.

7. Referencias.

**1. Presentación del proyecto:**

* 1. **Introducción:**

ServiRep es un proyecto que busca dar soporte a una empresa ficticia de nombre homónimo dedicada al mantenimiento de todo tipo de infraestructuras, electrónica etc. Trabajando tanto con empresas como con particulares a través de una aplicación web que gestiona de forma eficaz la comunicación entre el cliente, los diferentes trabajadores de la empresa y la empresa en sí a través de un sistema de reporte de incidencias, que deberán pasar por diferentes estados u etapas hasta su resolución, permitiendo la gestión de las mismas entre los diferentes roles dentro de la empresa (Operarios, gestores y administrador), facilitando el feedback entre cliente y empresa por los servicios prestados, implementando un sistema de puntuaciones entre los empleados para incentivar la productividad. A su vez, también incorpora un foro donde podrán participar todos los usuarios registrados en la web para debatir cuestiones.

* 1. **Objetivos**

Este proyecto pretende crear una aplicación web en la cual se gestione de forma adecuada y de fácil aprendizaje toda la curva temporal desde que un usuario reporta una incidencia hasta que dicha incidencia es calificada como resuelta.

Los sistemas de gestión de incidencias son necesarios en una gran cantidad de negocios para distribuir de forma adecuada la carga de trabajo entre sus trabajadores y facilitar la comunicación de los clientes, esta aplicación, sería integrada en la web de una empresa a medida de sus necesidades.

A continuación se describen los objetivos del proyecto en detalle:

* Proporcionar una aplicación web accesible para todos, tratando de que todo su contenido sea *responsive* para poder acceder desde diferentes dispositivos.
* Crear un sistema de reporte de incidencias sencillo e intuitivo para los usuarios de nuestra plataforma, dejando claro la descripción de los campos necesarios.
* Crear un sistema de monitoreo del estado de la incidencia a los clientes.
* Obtener el feedback del cliente, en estos tiempos donde la competencia en todos los ámbitos está llevada al extremo, todo negocio necesita estar en constante evolución y adaptarse a las necesidades de los consumidores.
* Permitir la asignación y el monitoreo de incidencias por parte de los gestores, permitiendo realizar cambios en el estado de la misma y la asignación a los técnicos.
* Permitir el fácil reporte del estado de sus tareas a los técnicos.
* Permitir la gestión de toda la plataforma al usuario administrador, pudiendo gestionar la mayoría de elementos de la plataforma, incluido la modificación de usuarios y control del foro.
* Permitir la interacción entre los usuarios de nuestra plataforma a través de un foro, creando una comunidad en torno a nuestra marca.
  1. **Tecnologías y herramientas empleadas:**

En esta sección se encontrarán explicadas todas las tecnologías y herramientas utilizadas en el desarrollo de este proyecto.

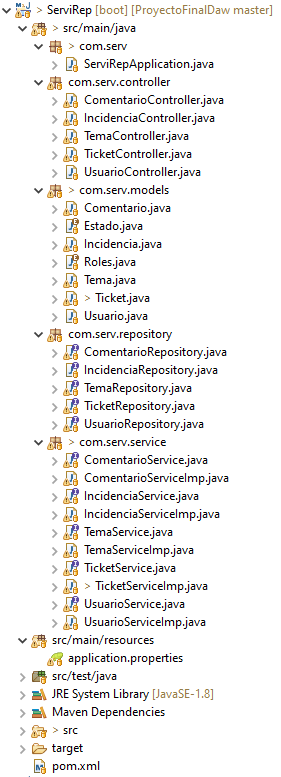
* Base de datos MySql: MySQL es un sistema de gestión de bases de datos relacional, probablemente el más popular del mundo, he utilizado esta base de datos por su fácil manejo y por la familiaridad que tengo con él.
* DBeaver: Sistema gestor de base de datos que facilita su gestión.
* EclipseEE: El IDE elegido para el desarrollo del Back End de nuestra aplicación en lenguaje Java.
* Visual Studio Code: IDE elegido para el desarrollo de la parte Front End de nuestra aplicación.
* Angular: Framework para aplicaciones web desarrollado en TypeScript elegido para desarrollar la parte Front de nuestra aplicación.
* HTML5: Lenguaje de marcas utilizado en la creación de nuestra aplicación.
* Java: Es el lenguaje de programación utilizado en el desarrollo de nuestro Back End.
* Spring boot: es un framework para el desarrollo de aplicaciones y contenedor de inversión de control, de código abierto para la plataforma Java.
* JPA: Es un framework del lenguaje de programación Java que maneja datos relacionales en aplicaciones usando la Plataforma Java.
* Hibernate: Es una herramienta de mapeo objeto-relacional para la plataforma Java que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de la aplicación.
* Bootstrap: Es un framework que proporciona un conjunto de hojas de estilo que proveen definiciones básicas de estilo para todos los componentes de HTML.
* CSS3: Es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado.
* Docker: Automatiza el despliegue de aplicaciones dentro de contenedores de software, proporcionando una capa adicional de abstracción y automatización de virtualización de aplicaciones en múltiples sistemas operativos.​
* Jenkins: Es un servidor de automatización de tareas.
* Wildfly: Servidor de aplicaciones Java EE.
* Diagrama de Gantt: Un diagrama que contiene la planificación del proyecto dividida en hitos y con unos plazos aplicados para cada hito, es un elemento esencial para establecer la línea cronológica de desarrollo del proyecto y para la correcta planificación del mismo.
* Mockups: Se trata de los bocetos de las vistas de la aplicación web que guiarán la parte estética de su desarrollo.
* Git: Sistema de control de versiones.

**2. Diseño del sistema:**

**2.1 Arquitectura del sistema:**

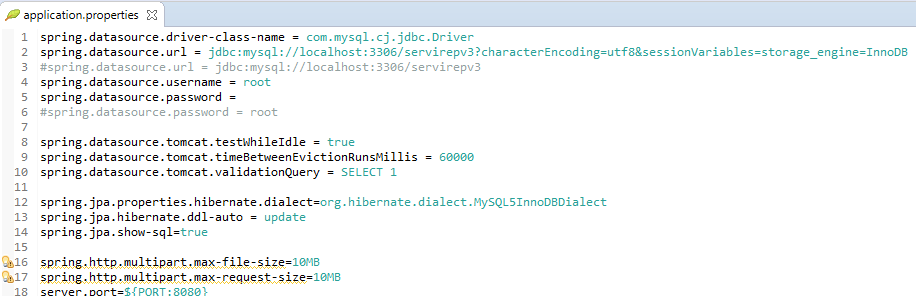
Nuestro proyecto se encuentra dividido esencialmente en dos partes, el Back End y el Front End.

El Back End está desarrollado con java 1.8, SpringBoot y JPA y será el encargado de alojar nuestros servicios con las diferentes funcionalidades que interactuarán con la base de datos, a continuación veremos una imagen con la estructura de nuestro proyecto:



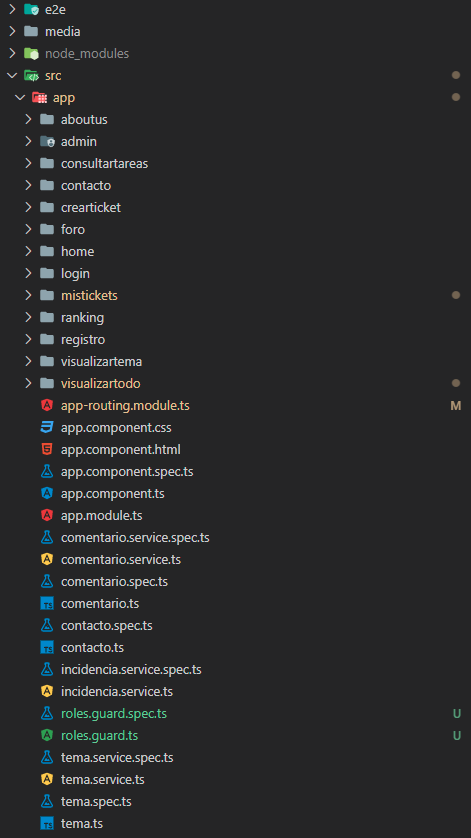
Como podemos observar en la imagen, encontramos una estructura de carpetas divididas entre controladores, servicios, repositorio y entidades:

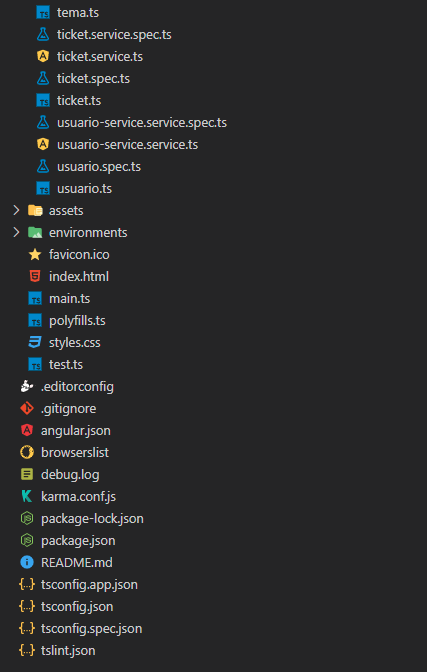
* src/main/java
  + com.serv.controller: Contiene los controladores para la recepción de peticiones del front.
  + com.serv.models: Contiene las clases con las anotaciones JPA con las cuales nos genera toda la estructura de la base de datos de forma automática.
  + com.serv.repository: Contiene las consultas a la base de datos.
  + com.serv.service: Contiene los servicios que dan la funcionalidad a la aplicación.
* src/main/resources:
  + application.properties: Archivo que contiene la configuración para el acceso con la base de datos, que veremos a continuación.



Para la parte del Front End he utilizado Angular 8 junto con Node v12.13.0 junto con el editor de texto Visual Studio Code y un puñado de plugins que hacen la vida mucho más cómoda a la hora de desarrollar.

Como veremos en la siguiente imagen, la estructura del Back End es la típica estructura que te genera un proyecto Angular en la que se distinguen componentes, servicios y clases, además de diversos archivos de configuración, archivos css etc.

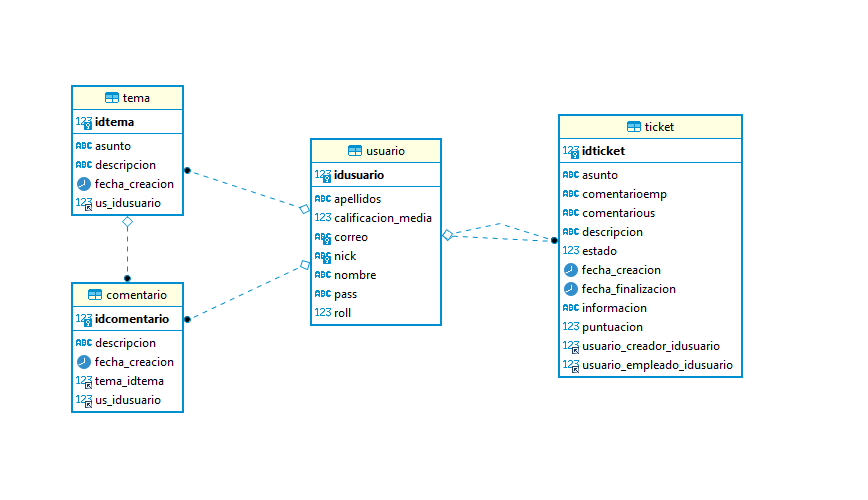




Entre los elementos podemos destacar las siguientes:

* src/app: Donde se encuentras los componentes.
* clases: Contienen los objetos definidos como en la bbdd.
* guards: Usados para capar el acceso por url a los diferentes roles.
* services: Donde se encuentran los métodos que realizan peticiones al Back End.

En la siguiente imagen observamos la estructura completa de la base de datos, como apunte, anteriormente era significativamente más compleja pero tras diversos problemas y tras comentarlo con mi tutora del proyecto logramos solventar esos problemas optimizando la base de datos y quedando significativamente más reducida en cuanto a entidades y relaciones.



**2.2 Requisitos del sistema:**

Existen dos tipos de requisitos:

• **Requisitos funcionales:** son aquellos que consistan en una característica requerida del sistema que expresa una capacidad de acción del mismo – una funcionalidad; generalmente expresada en una declaración en forma verbal.

• **Requisitos no funcionales**: son aquellos que consistan en una característica requerida del sistema, del proceso de desarrollo, del servicio prestado o de cualquier otro aspecto del desarrollo, que señala una restricción del mismo.

Requisitos funcionales:

* Sistema operativo: La aplicación debería funcionar sin afectar significativamente la experiencia con cualquier sistema operativo y con cualquier navegador web y dispositivos.

Requisitos funcionales:

* La gestión del sitio podrá ser llevada a cabo por un usuario sin conocimientos en programación.
* La web será atractiva a la vista y su uso será sencillo, requiriendo una corta formación en su uso los usuarios gestores/admin/operarios.

**2.3 Definición de interfaces de usuario:**

La interfaz de usuario es el medio con que el usuario puede comunicarse con una máquina, equipo, computadora o dispositivo, y comprende todos los puntos de contacto entre el usuario y el equipo.

Normalmente suelen ser fáciles de entender y fáciles de accionar, aunque en el ámbito de la informática es preferible referirse a que suelen ser "usables", "amigables e intuitivos" porque es complejo.

La interfaz de usuario es el espacio donde se producen las interacciones entre seres humanos y máquinas. El objetivo de esta interacción es permitir el funcionamiento y control más efectivo de la máquina desde la interacción con el humano.

Las interfaces básicas de usuario son aquellas que incluyen elementos como menús, ventanas, contenido gráfico, cursor, los beeps y algunos otros sonidos que la computadora hace, y en general, todos aquellos canales por los cuales se permite la comunicación entre el ser humano y la computadora.

El objetivo del diseño de una interfaz es producir una interfaz que sea fácil de usar (explicarse por sí misma), eficiente y agradable para que al operar la máquina dé el resultado deseado.

A continuación se muestran las funciones que conforman la aplicación, según el rol del usuario que se encuentre conectado. Recordemos que existen (Usuario sin identificar, usuario, operario, gestor y administrador).

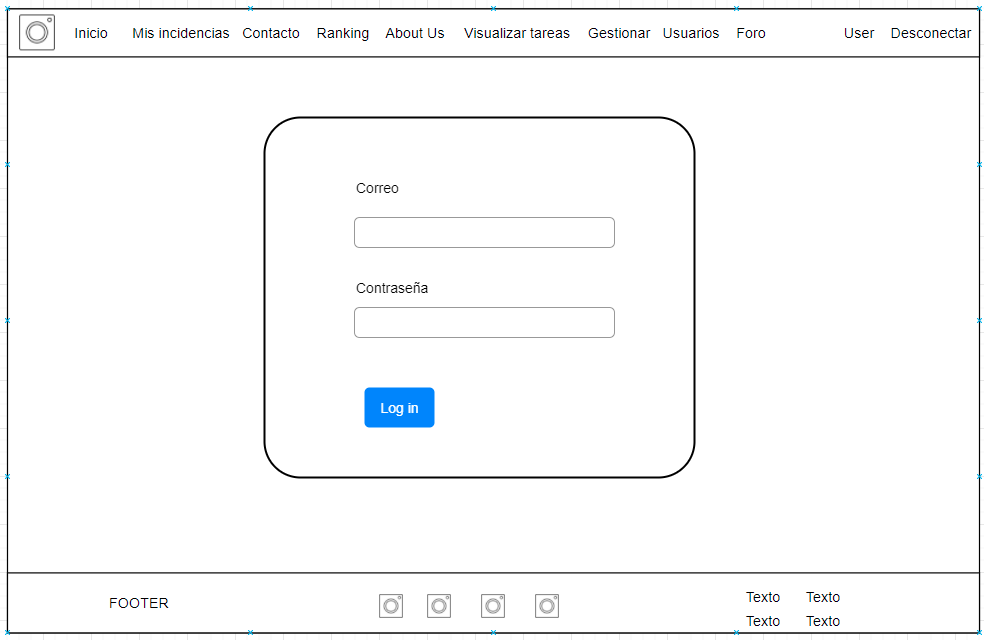
* **Home:** Es la vista principal de la web, no contiene una funcionalidad específica (Muestra contenido), se debe asegurar que todos los botones y enlaces de la misma funcionen correctamente redireccionando a las páginas correspondientes. Todos los usuarios independientemente de su rol tienen acceso a esta página.
* **About Us:** Una página meramente informativa, sin funcionalidad implementada, mostrando información relacionada con la empresa y su entorno ( Mapa, imágenes…). Todos los usuarios independientemente de su rol tienen acceso a esta página.
* **Página de contacto:** En esta página se haya un formulario que al completarse genera un ticket en la base de datos con estado de pendiente de confirmación. Todos los usuarios REGISTRADOS tienen acceso a esta página.
* **Login:** En esta página se encuentra el formulario de login, el cual habrá que completar para realizar el logeo con éxito, esta página está visible para un usuario no logeado.
* **Registro:** En esta página se encuentra el formulario de registro, el cual habrá que completar para obtener el rol de usuario registrado y tener acceso a otras secciones, esta página está visible para un usuario no logeado.
* **Mis incidencias:** Esta vista se encuentra disponible para todos los usuarios REGISTRADOS de la web, independientemente de su roll, en ella se podrá visualizar sus tickets y una vez resueltos, podrá calificar la atención recibida por el empleado y un comentario.
* **Top Empleados:** En esta vista hay un ranking con la puntuación media de los empleados, solo es visible para usuarios con rol de operario, gestor o administrador.
* **Visualizar tareas:** En esta página se muestra una tabla con las tareas que tiene asignada el empleado que la está visualizando, pudiendo modificar su estado y dejar un comentario. Esta vista está disponible para los usuarios con rol de operario y admin.
* **Gestionar tickets:** En esta página se muestra una tabla con todos los tickets, pudiendo modificar su estado y asignar un operario así como eliminarlos. Esta vista está disponible para los usuarios con rol de gestor y admin.
* **Usuarios:** En esta página se muestra una tabla con todos los usuarios, pudiendo modificar su rol así como eliminarlos. Esta vista está disponible para los usuarios con rol de y admin.
* **Foro:** Esta vista está disponible para todos los usuarios logeados en la web y en ella podrán visualizar los temas existentes, así como crear nuevos temas y eliminar los temas que les pertenecen, pueden acceder al detalle del tema con un botón el usuario admin puede borrar todo.
* **Tema:** Esta vista está disponible para todos los usuarios logeados en la web y en ella podrán visualizar la información del tema, así como ver los comentarios existentes en ese mismo tema, crear un nuevo comentario y borrar los comentarios que le pertenecen, el usuario admin puede borrar todo.

**2.4 MockUps:**

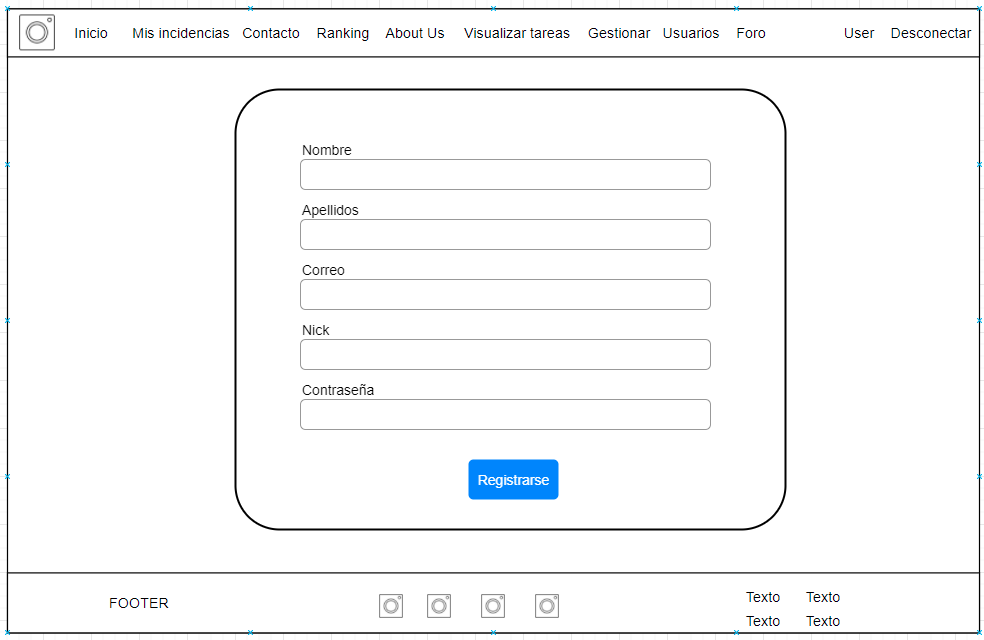
Los mockups son una herramienta muy útil que se utilizan para tener las directrices a la hora de desarrollar una vista en la web, en nuestro caso, los realizamos en la etapa de planificación del proyecto y nos han guiado en todo el proceso, aunque hemos tenido que modificar algunos aspectos en algún caso.

A continuación podemos observarlos.

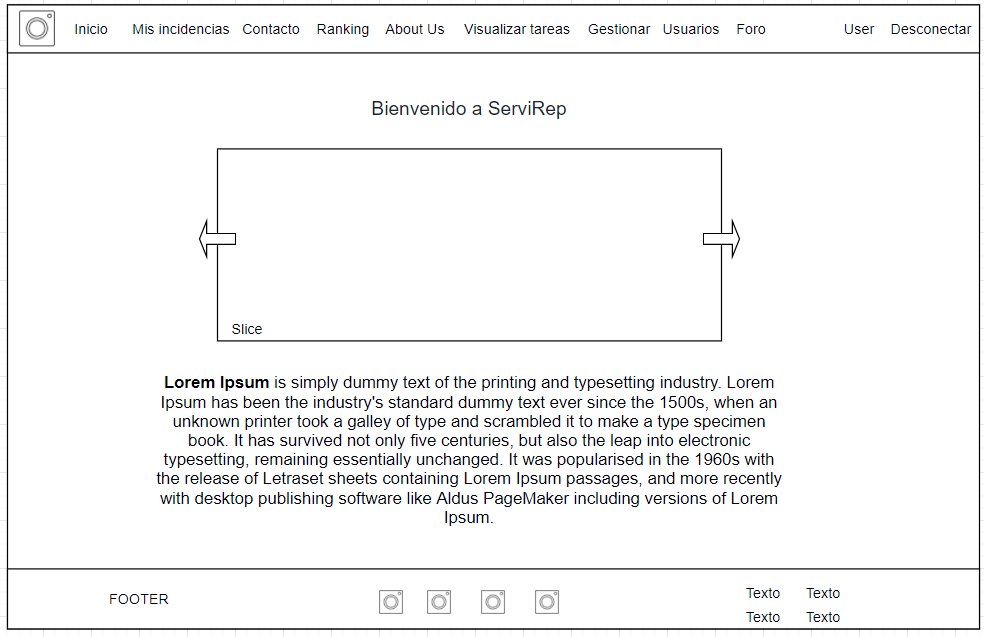
Log in



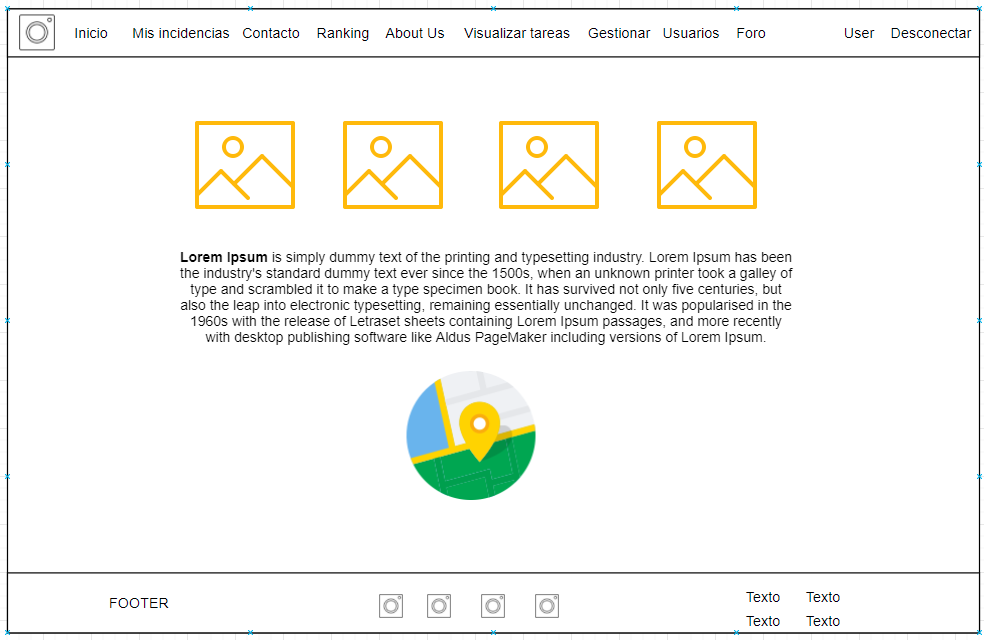
Registro



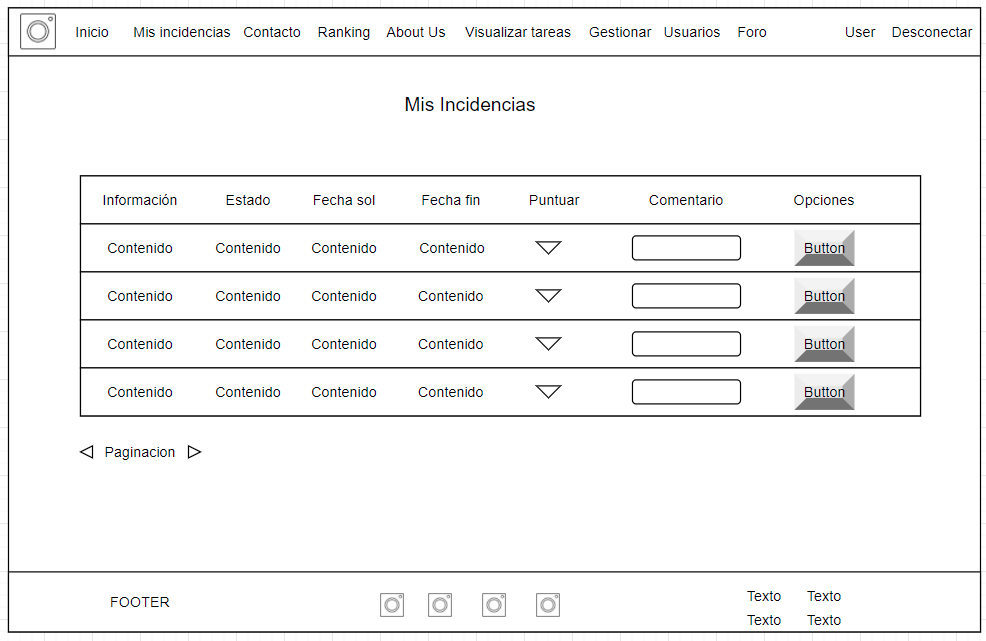
Home



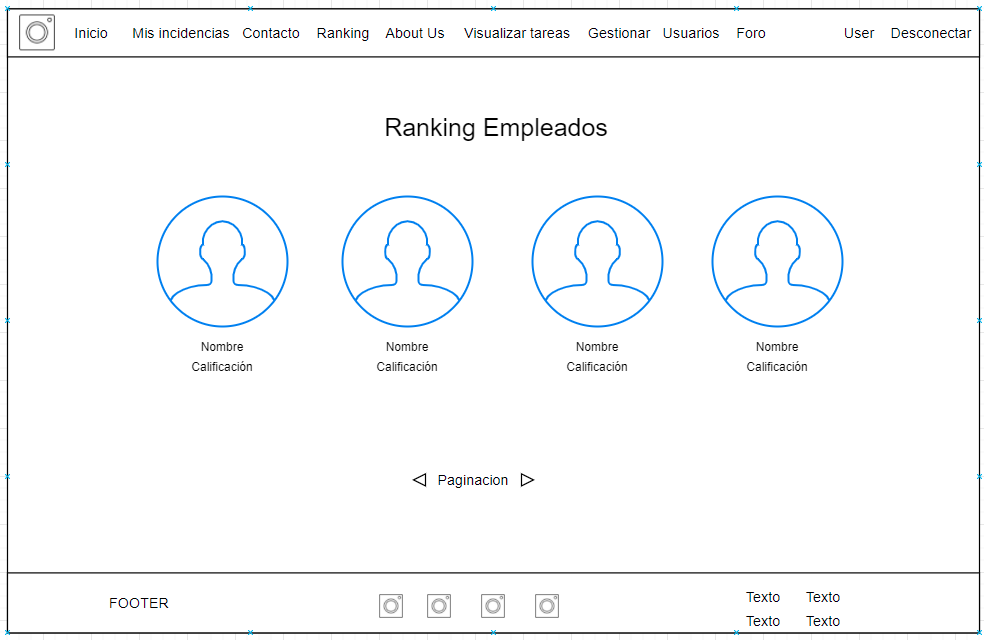
About us



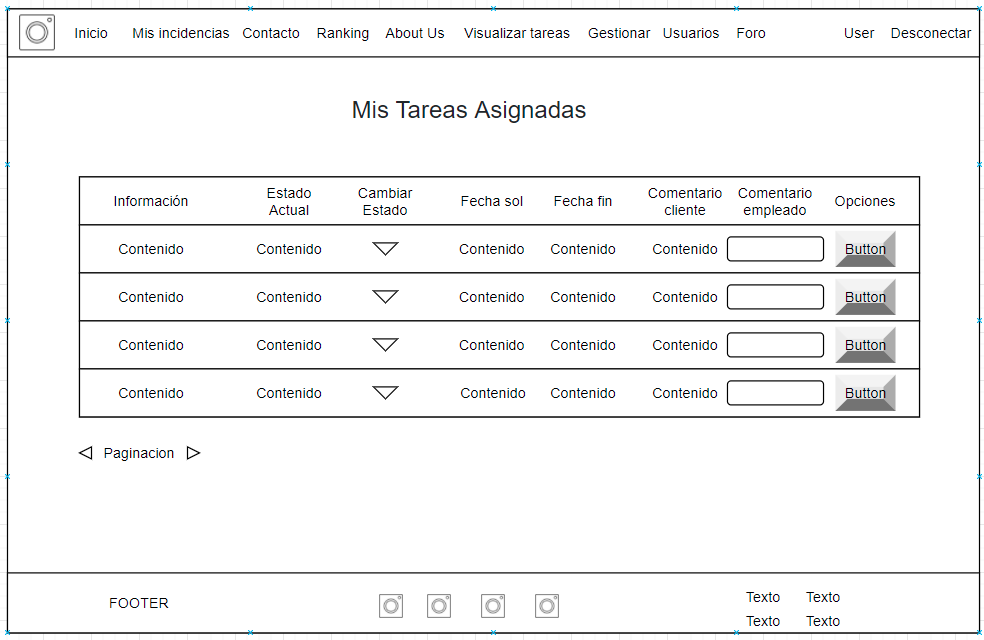
Mis incidencias



Ranking Empleados



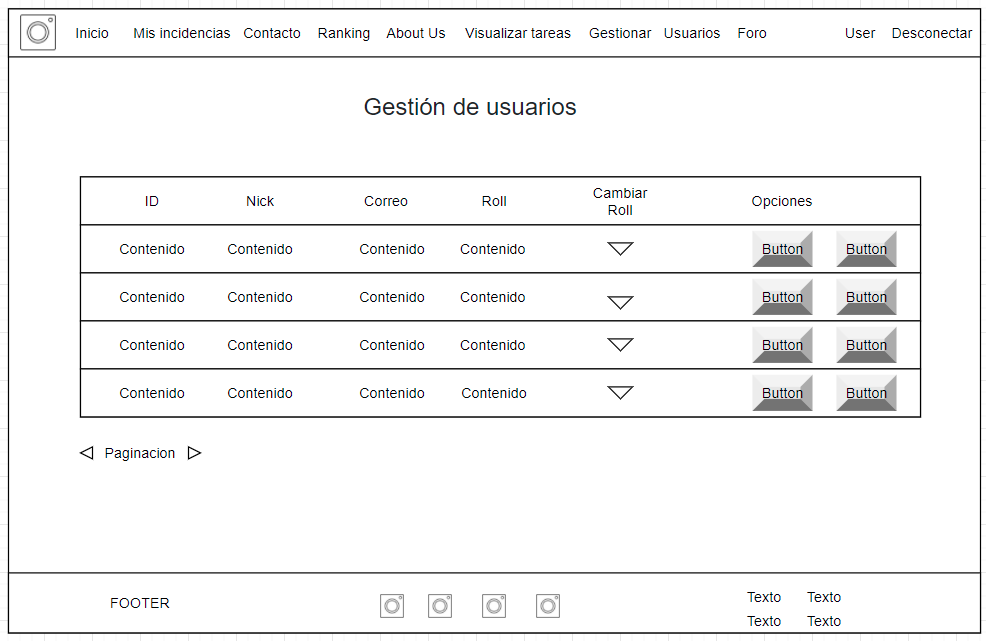
Visualizar Tareas



Gestión de incidencias



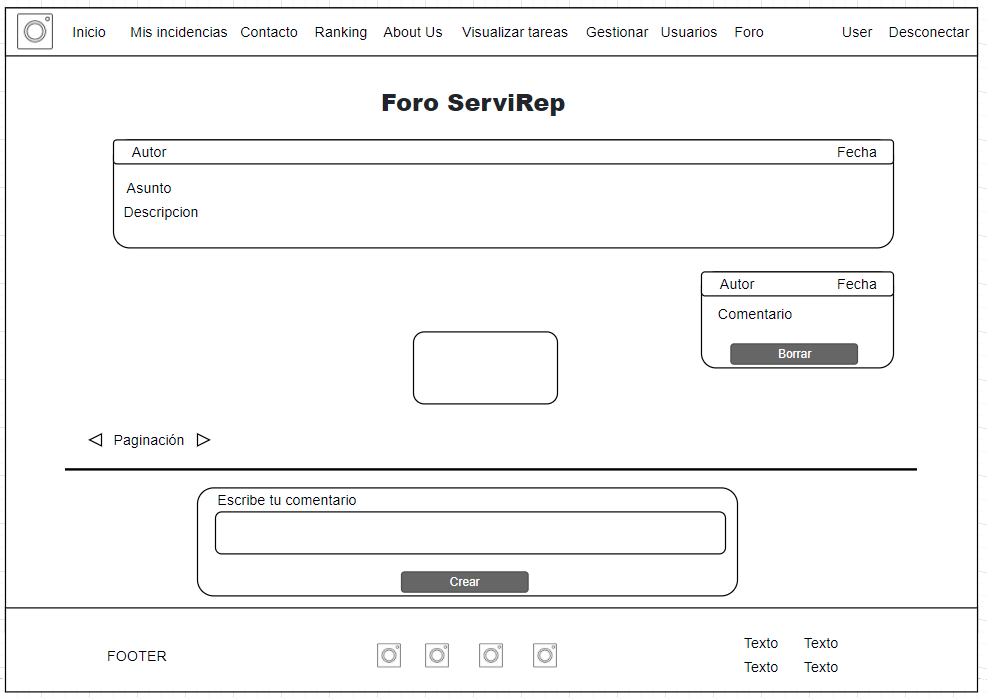
Usuarios



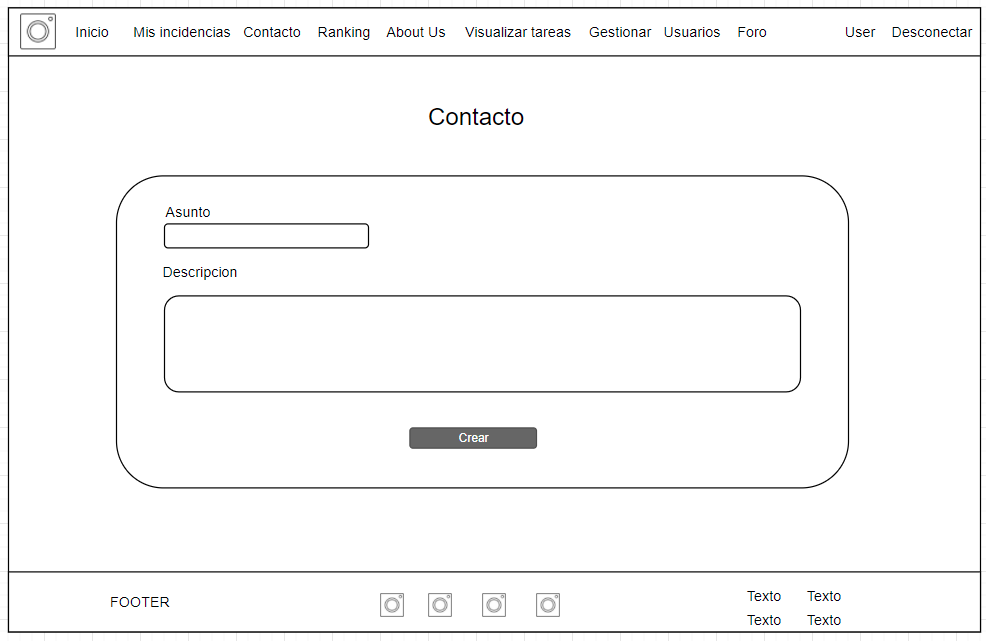
Foro



Tema



Contacto



**2.5 Planificación temporal:**

La Planificación Temporal es la identificación de tareas, asignación de tiempos y recursos a dichas tareas y planificación de la secuencia de ejecución de forma que el tiempo de desarrollo del proyecto sea mínimo.

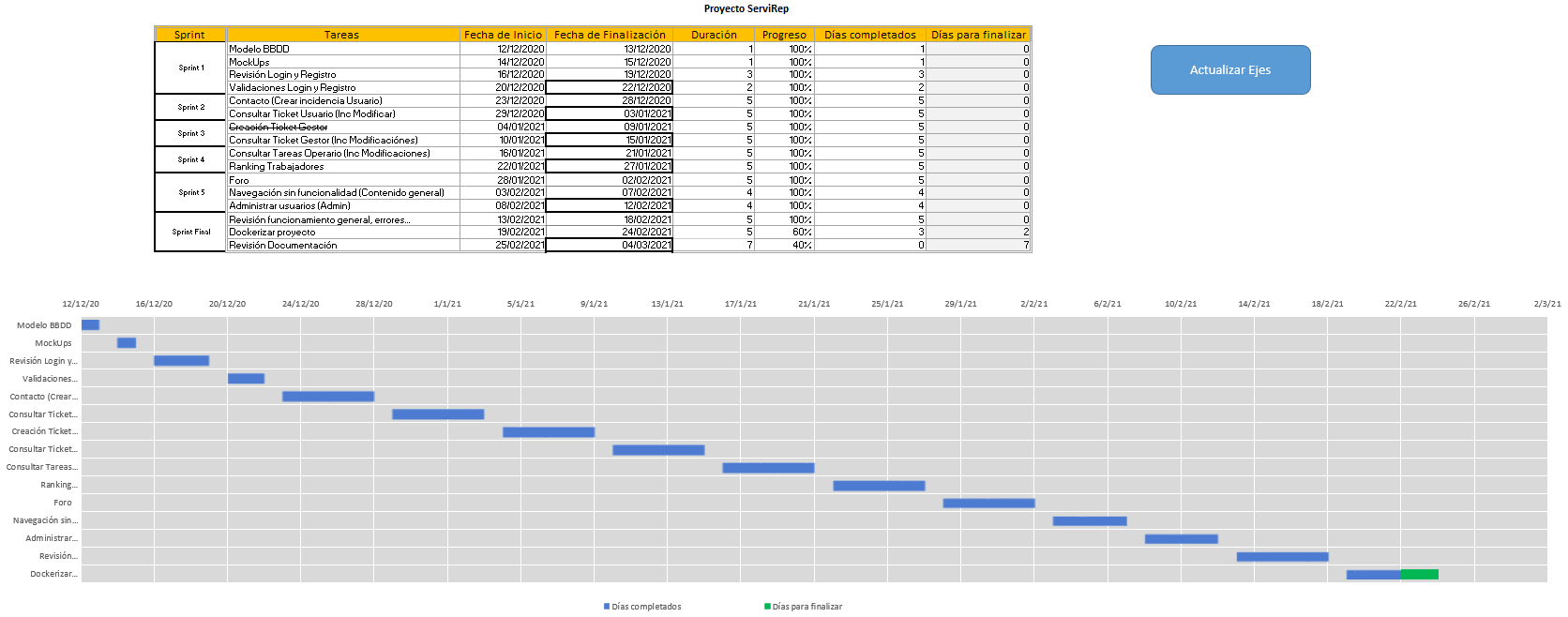
Principios de la planificación temporal:

* **Compartimentación**: descomposición del proyecto en un número manejable de tareas.
* **Interdependencia**: Se deben determinar las dependencias de cada tarea.
* **Asignación de tiempo**: A cada tarea se le debe asignar un cierto número de unidades de trabajo, una fecha de inicio y otra de finalización.
* **Validación del esfuerzo**: A medida que se realiza la asignación de tiempo, el gestor del proyecto se tiene que asegurar de que los técnicos necesarios estarán disponible en cada momento.
* **Responsabilidades definidas**: Cada tarea que se programe debe asignarse a un miembro específico del proyecto.
* **Resultados definidos**: El resultado de cada tarea, normalmente un producto, deberá estar definido. Los productos se combinan generalmente en entregas.
* **Hitos definidos**: Todas las tareas grupos de tareas deben asociarse con algún hito del proyecto. Se considera un hito cuando se ha revisado la calidad de uno o más productos y se han aceptado.

La planificación temporal se ha llevado a cabo a través de un diagrama de Gantt que es una herramienta gráfica cuyo objetivo es exponer el tiempo de dedicación previsto para diferentes tareas o actividades a lo largo de un tiempo total determinado.

Las tareas se han dividido en hitos con una fecha de entrega que más o menos he intentado cumplir, con algunos ligeros retrasos en alguna ocasión por problemas durante el desarrollo.

A continuación adjunto imagen del mismo:



**2.6 Especificación del plan de pruebas:**

El plan de pruebas es un documento formal que define los objetivos de la prueba de un sistema, establece y coordina una estrategia de trabajo, y provee del marco adecuado para elaborar una planificación paso a paso de las actividades de prueba.

En nuestro caso no ha existido un plan de pruebas predeterminado, sino que he ido realizando pruebas funcionales a medida que he ido desarrollando la aplicación.

Estas pruebas están compuestas por:

* Log in:
  + Introducir usuario incorrecto, contraseña incorrecta y comprobar que no realiza el inicio de sesión con una ventana emergente “Correo o contraseña incorrectos”.
  + Introducir usuario incorrecto, contraseña correcta y comprobar que no realiza el inicio de sesión con una ventana emergente “Correo o contraseña incorrectos”.
  + Introducir usuario correcto, contraseña incorrecta y comprobar que no realiza el inicio de sesión con una ventana emergente “Correo o contraseña incorrectos”.
  + Introducir usuario correcto, contraseña correcta y comprobar que realiza el inicio de sesión con una ventana emergente “Logeado con éxito”.
* Comprobar que al desconectarme se limpia la sessionStorage y nos envía a la página home.
* Mis incidencias:
  + Introduzco un comentario pero no una calificación teniendo esta disponible y observo que sale una ventana emergente reflejando dicho error.
  + Introduzco una calificación pero no un comentario y observo que sale una ventana emergente reflejando dicho error.
  + Introduzco comentario y calificación correctamente y compruebo que me sale un pop pup reflejando este mensaje de éxito.
  + Introduzco un comentario cuando no puedo ver la puntuación y compruebo que me sale la ventana emergente de éxito.
* Contacto:
  + Introduzco diferentes datos en el formulario y compruebo que me salen las validaciones creadas anteriormente, en el caso de cumplir con los criterios de validación puedo pulsar crear y se crea correctamente saltando un pop pup indicando el éxito en la operación.
* Top empleados:
  + Verifico al entrar en la página que los datos cargan correctamente.
* Gestionar tickets:
  + Introduzco un estado pero no asigno un operario y observo que sale una ventana emergente reflejando dicho error.
  + Introduzco un operario pero no estado y observo que sale una ventana emergente reflejando dicho error.
  + Introduzco los datos correctamente y al salvar compruebo que me sale una ventana emergente indicando el éxito.
* Visualizar tareas:
  + Introduzco un comentario pero no un estado y observo que sale una ventana emergente reflejando dicho error.
  + Introduzco un estado pero no un comentario y observo que sale una ventana emergente reflejando dicho error.
  + Introduzco los datos correctamente y al salvar compruebo que me sale una ventana emergente indicando el éxito.
* Gestión de usuarios:
  + Compruebo que al salvar un usuario sin especificar su rol me salta un pop up indicándome dicho error.
  + Compruebo que al seleccionar un rol y pulsar en salvar al modificar un usuario me sale un pop up de éxito.
* Foro:
  + Compruebo que los post mostrados salen ordenados por fecha de publicación.
  + Compruebo con el perfil de administrador que al pulsar el botón de borrar el post desaparece de la lista y de la bbdd.
  + Compruebo que al pulsar el botón eliminar con el usuario creador del post el post desaparece de la lista y de la bbdd.
  + Compruebo que el botón eliminar post solo es visible para el usuario administrador y para el creador del post.
  + Compruebo que puedo ver el detalle del tema pulsando sobre el botón visitar.
  + Compruebo que el formulario de creación de tema cumpla con las validaciones y al pulsar en publicar salgan los correspondientes mensajes de error/éxito.
* Tema:
  + Compruebo que los comentarios mostrados salen ordenados por fecha de publicación.
  + Compruebo con el perfil de administrador que al pulsar el botón de borrar el comentario desaparece de la lista y de la bbdd.
  + Compruebo que al pulsar el botón eliminar con el usuario creador del comentario el comentario desaparece de la lista y de la bbdd.
  + Compruebo que el botón eliminar comentario solo es visible para el usuario administrador y para el creador del comentario.
  + Compruebo que el formulario de creación de comentario cumpla con las validaciones y al pulsar en publicar salgan los correspondientes mensajes de error/éxito.

**3. Desarrollo del sistema.**

**3.1. Descripción del entorno de desarrollo.**

Un entorno de desarrollo es un conjunto de procedimientos y herramientas que se utilizan para desarrollar un código fuente o programa. Este término se utiliza a veces como sinónimo de entorno de desarrollo integrado (IDE), que es la herramienta de desarrollo de software utilizado para escribir, generar, probar y depurar un programa. También proporcionan a los desarrolladores una interfaz de usuario común (UI) para desarrollar y depurar en diferentes modos.