

Documentación
técnica

PROYECTO FINAL

TRELLON'T

**CFGs DE
DESARROLLO DE
APLICACIONES WEB**

Odalys Tatiana Manzanares Romero
IES LUIS VIVES

TABLA DE CONTENIDO

- Definiciones y especificación de requerimientos..... 2
 - Definición general del proyecto..... 2
 - Funcionamiento 2
- Especificaciones de procedimientos 3
 - De desarrollo..... 3
- Arquitectura del sistema..... 3
 - Base de datos..... 3
 - Backend..... 4
 - Frontend..... 4

DEFINICIONES Y ESPECIFICACIÓN DE REQUERIMIENTOS

DEFINICIÓN GENERAL DEL PROYECTO

La idea general del proyecto es un gestor de proyectos y tareas. Una aplicación para colaborar en la gestión de proyectos con diferentes usuarios y organizar de manera sencilla dichos proyectos con sus tareas y participantes.

Está dirigido a cualquier tipo de usuarios, siendo posible usarse con fines de organización de equipos dentro de una empresa o para tus tareas personales.

FUNCIONAMIENTO

La aplicación abre con una pantalla de inicio de sesión o registro, ya que dentro de la aplicación tendrás tus tareas. Puesto que esto es de carácter personal será obligatorio iniciar sesión para poder acceder a los datos de esta.

Una vez dentro de la aplicación tendrás acceso a la lista de tus proyectos, tanto los proyectos que has creado como a los que eres participante.

Podrás convertirte en propietario de todos los proyectos que crees. Esto te dará ciertos permisos que solo tú, como propietario, tendrás acceso. Podrás borrar y modificar estos proyectos. También podrás añadir a otros usuarios a participar en tu proyecto poniendo su email.

Si eres participante tendrás acceso a los proyectos, pero no podrás ni borrarlo ni editarlo. Tampoco podrás añadir a más participantes.

En caso de no tener ningún proyecto todavía, podrás añadirlo con facilidad convirtiéndote en el propietario de ese nuevo proyecto. Solo es necesario un título de proyecto y, si quieres, una breve descripción.

Una vez dentro del proyecto, tanto si eres el propietario como si eres un participante, podrás añadir columnas, tareas y poner comentarios. También tendrás acceso al chat del proyecto, al cual pueden participar todos los participantes de dicho proyecto.

Las columnas son la manera que tiene la aplicación de organizar las tareas que añadas. Podrás ir moviendo las tareas de una columna a otra siempre que quieras.

Un ejemplo de uso sería tener tres columnas: *Pendiente de hacer*, *En proceso*, *Finalizado*. De esta manera podrías organizar la lista de tareas en esas tres categorías conforme vaya cambiando el estado de tu tarea.

Las tareas son pequeños bloques que podrás ampliar para ver sus detalles y añadir comentarios. Las tareas tendrán un crear, que será quien haya creado dicha tarea, y un responsable que podrás elegir o dejar en blanco. Si no añades un responsable de la tarea se te asignará a ti mismo como responsable de esta. El contenido de las tareas no es modificable.

Dentro del detalle de las tareas podrás ver los diferentes comentarios que se han ido añadiendo a esta. Podrás ver quien y cuando han añadido el nuevo comentario. También podrás añadir más si quieres, y si eres el creador de dicho comentario podrás editarlo siempre y cuando la edición lo hagas antes de dos minutos.

ESPECIFICACIONES DE PROCEDIMIENTOS

DE DESARROLLO

Para desarrollar esta aplicación han sido necesarias herramientas como Eclipse, Visual Studio Code, Workbench...

Desarrollada completamente en Java v16.0.1 para el backend, ejecutado en un servidor apache Tomcat v10.0.13, MySQL v8.0.28 para la Base de Datos y JavaScript vES6, HTML 5, CSS 3, Bootstrap v5.0.2 utilizado en frontend. La comunicación entre backend y frontend ocurre a través de AJAX desde JavaScript y el formato de datos, en la mayoría de casos, he usado JSON.

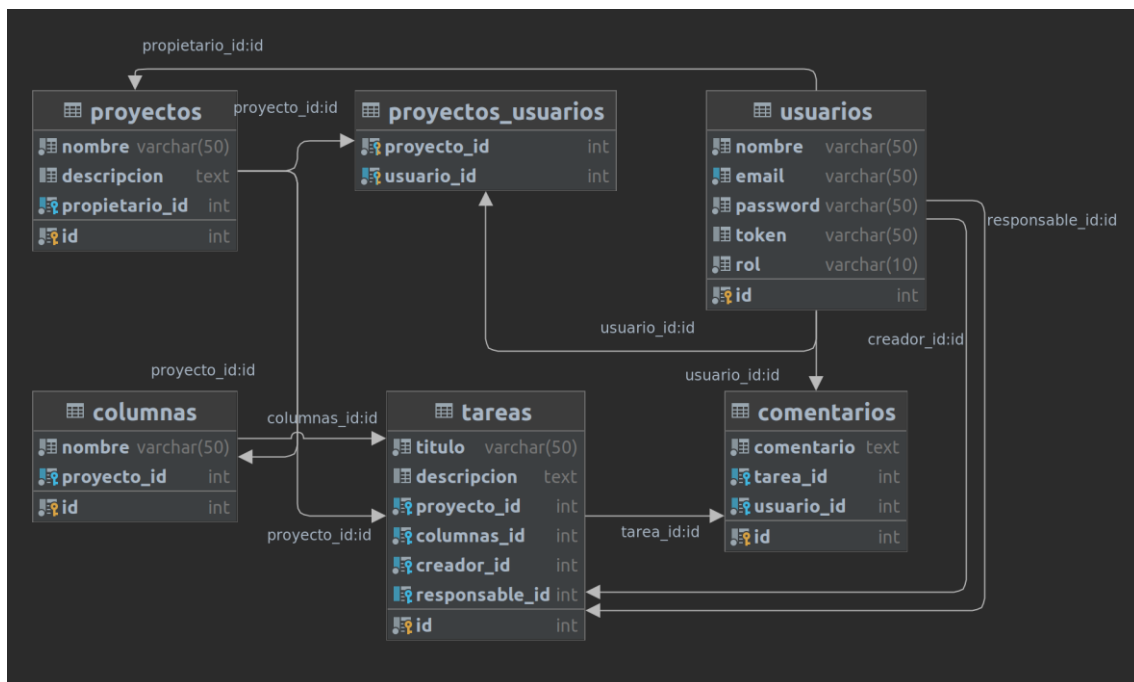
Debido a que JavaScript se usa en la vES6 esta aplicación no funcionara para navegadores antiguos. Por ejemplo, Internet Explorer.

ARQUITECTURA DEL SISTEMA

Para explicar correctamente cual es la arquitectura del sistema debemos separar las partes que lo componen, empezando por el funcionamiento de la base de datos, la organización de tablas, la estructura de carpetas y el flujo que sigue toda la aplicación para relacionar todas estas partes.

BASE DE DATOS

La aplicación usa una base de datos relacional MySQL v8.0.28. A continuación muestro un diagrama de cómo están relacionadas las tablas en esta base de datos.



Partimos de una tabla usuario, la cual almacenará, como datos importantes, el token de inicio de sesión. También tendremos una tabla de proyectos la cual guardara en ella la información de cada proyecto, su título, breve descripción de este y el propietario del proyecto.

Los propietarios tendrán ciertos permisos extra, como son, eliminar el proyecto e invitar a nuevos usuarios a dicho proyecto.

En la tabla Proyecto_Usuario se almacenarán los usuarios que PERTENECEN a un proyecto, es decir, esta tabla solo guardara a los usuarios que participen en proyectos. (No guarda los

proyectos de los cuales eres propietario). Con esta relación podré darles a los usuarios participantes acceso a los proyectos en los que este como colaborador y también, relacionándolo con la tabla de Proyecto, los proyectos que le pertenecen.

Los proyectos tendrán columnas. Las columnas, a su vez, tendrán tareas y las tareas podrán tener comentarios. Cada bloque tiene su propia tabla y se relacionan entre ellas por su ID.

BACKEND

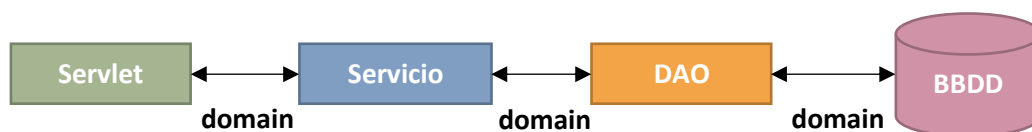
Toda la comunicación de peticiones http esta en la carpeta Servlets usando WebServlets de Java.

Estos Servlets únicamente se encargan de comprobar que los datos recibidos son correctos y generar una respuesta para el frontend.

Los Servlets usaran los Servicios para realizar las acciones que requieran. Estos Servicios se encuentran en la carpeta Servicios. Los servicios podrán llamar a otros servicios o a los DAOS para comunicarse con la base de datos.

Los DAOS usaran transacciones de MySQL para asegurarse que toda la lectura y escritura ocurre de la manera correcta. Estos DAOS ejecutaran Queries escritas en lenguaje SQL.

En común a todo lo antes mencionado, existen las clases, en la carpeta domain, las cuales son la que le dan forma a la aplicación. Siendo estos los bloques más pequeños que usara la aplicación. Por ejemplo, la clase Usuario. Sin esta clase nada podría funcionar ya es la cual transporta todos los datos.



FRONTEND

Existen diferentes páginas, cada una tendrá un archivo HTML. A su vez, cada uno de estos archivos tiene otro archivo JavaScript que se encarga de controlar todo lo que ocurre en su HTML. Desde estos archivos JS se utilizan los archivos que existen en las carpetas *conexión_servidor* y *componentes*.

En la carpeta *conexión_servidor* existen archivos que se encargan de realizar las peticiones http para el servidor (las cuales ejecutan los Servlets). En esta carpeta también hay un archivo *urls.js* que contiene las funciones necesarias para generar todas las URLS del proyecto.

En la carpeta *componentes* hay archivos que tienen funciones para generar el HTML de todos los componentes que vaya a querer reutilizar y que necesite generar de manera dinámica.

Por ejemplo, para crear un proyecto habría que entrar desde un botón del HTML de la pagina *proyectos.html*, esto ejecutara una función del archivo *paginas/proyectos.js*. Esta función llamara a la función *crearProyecto()* del archivo *conexión_servidor/proyectosApi.js*. Esta función sacará la URL necesaria para crear el proyecto del archivo *conexión_servidor/urls.js*.

Después de crear el proyecto se refrescará el HTML de la pagina para que los cambios sean visibles. En el momento de generar el HTML varias partes de este saldrían de archivos de la carpeta *componentes*. Esto está hecho para así ahorrar tiempo y código.