

Practica TS1

☰ Tags

Manual Técnico Tasky

Descripción General

Este proyecto es una aplicación web que sigue el patrón de diseño MVC (Modelo-Vista-Controlador) y está desarrollado en PHP 8.2. Utiliza Docker para simplificar el despliegue y la configuración del entorno. La funcionalidad principal de la aplicación incluye la gestión de usuarios, proyectos, tareas, y reportes. El proyecto se organiza en carpetas según sus componentes principales, como controladores, modelos, servicios y vistas.

Tecnologías utilizadas:

- **PHP 8.2:** Lenguaje de programación principal.
- **MySQL:** Base de datos para almacenar la información de usuarios, proyectos, y tareas.
- **Docker y Docker Compose:** Para la configuración y despliegue del entorno de desarrollo.
- **Nginx:** Servidor web.
- **Composer:** Para la gestión de dependencias de PHP.

Estructura del Proyecto

El proyecto sigue una estructura clara y modularizada. A continuación se describe cada parte clave del sistema.

1. Carpeta `app/`

Contiene todos los archivos relacionados con la lógica de negocio, organizados en Controladores, Modelos, Servicios y Vistas.

- **Controllers/:** Aquí se encuentran los controladores que manejan las solicitudes HTTP y definen la lógica de la aplicación.
 - `BoardController.php` : Controla las acciones relacionadas con los tableros de proyectos.
 - `DashboardController.php` : Maneja la lógica del panel de control.
 - `ReportController.php` : Controlador para generar reportes.
 - `UserController.php` : Controlador para gestionar usuarios.
- **Models/:** Define las entidades del sistema y su interacción con la base de datos.
 - `Project.php` : Modelo para la entidad "Proyecto".
 - `ReportModel.php` : Modelo que gestiona la lógica de los reportes.
 - `User.php` : Modelo para la entidad "Usuario".
- **Services/:** Contiene servicios que abstraen la lógica de acceso a datos y otros servicios auxiliares.
 - `AuthService.php` : Servicio de autenticación de usuarios.
 - `Database.php` : Servicio para la conexión a la base de datos.
 - `ProjectService.php` : Servicio para la gestión de proyectos.
- **Views/:** Vistas utilizadas para mostrar la interfaz al usuario.
 - `404NotFound.php` : Vista que muestra un mensaje de error 404.
 - Varias vistas para formularios de proyectos, tareas y reportes (`create_task.php` , `dashboard.php` , `ReportTasksByUserAndProjectView.php` , etc.).

2. Archivos principales

- `composer.json` y `composer.lock` : Archivos para la gestión de dependencias del proyecto.
- `index.php` : Punto de entrada principal de la aplicación, donde se enrutan las solicitudes HTTP.

3. Carpeta `config/`

- `config.php` : Archivo de configuración donde se definen las constantes globales, como la configuración de la base de datos.

4. Carpeta `css/`

- `styles.css` : Hoja de estilos utilizada en las vistas HTML del proyecto.

5. Carpeta `docker/`

- `nginx.conf` : Archivo de configuración de Nginx utilizado en el contenedor de Docker.

6. Carpeta `docs/`

- Contiene documentación adicional, como los scripts de la base de datos (`script.sql`) y archivos de arquitectura (`practice1.architect`).

7. `docker-compose.yml`

Define los servicios utilizados en el entorno de Docker, como el servidor web Nginx, la base de datos MySQL, y PHP.

Cómo Levantar el Proyecto con Docker Compose

Este proyecto está configurado para ejecutarse de manera sencilla utilizando `docker-compose`. Para levantar el proyecto en tu entorno local, sigue estos pasos:

Requisitos previos

- Tener instalados Docker y Docker Compose.

Pasos para levantar el entorno:

1. Ejecuta Docker Compose para iniciar los contenedores:

```
docker-compose up -d
```

Este comando levantará todos los servicios definidos en el archivo `docker-compose.yml`, que incluyen:

- **PHP:** Para procesar la aplicación.

- **MySQL:** Base de datos.
 - **Nginx:** Servidor web para servir la aplicación.
2. Accede a la aplicación en tu navegador:
Visita `http://localhost:8000` para ver la aplicación funcionando.
 3. (Opcional) Para ver los logs de los servicios:

```
docker-compose logs -f
```

Migraciones de Base de Datos

Para cargar las tablas en la base de datos, asegúrate de ejecutar el script de SQL que se encuentra en `docs/script.sql`:

1. Conéctate a la base de datos del contenedor MySQL:

```
docker exec -it nombre_contenedor_mysql bash
```

2. Importa el script SQL:

```
mysql -u root -p < /path/to/docs/script.sql
```

Reportes en el Proyecto

Se generan varios reportes a través del `ReportController.php`, utilizando las vistas correspondientes en la carpeta `Views/`. Los principales reportes incluyen:

1. **Reporte de tareas por usuario y proyecto:** Usa la vista

`ReportTasksByUserAndProjectView.php`.

- Controlador: `getTaskCountByUserAndProject()`.

2. **Reporte de cantidad de tareas por estado:** Usa la vista

`ReportTaskCountByStatusView.php`.

- Controlador: `getTaskCountByStatus()`.

3. Reporte de tiempo gastado por proyecto: Usa la vista

`ReportTotalTimeByProjectView.php` .

- Controlador: `getTotalTimeByProject()` .

4. Historial de cambios de estado de tareas: Usa la vista

`ReportTaskStatusHistoryView.php` .

- Controlador: `getTaskStatusHistory()` .