

Guía para Clonar y Ejecutar el Proyecto Laravel

Requisitos previos:

- **Git:** Para clonar el repositorio y manejar el control de versiones.
- **PHP:** Laravel requiere PHP 8.1 o superior.
- **Composer:** Para gestionar las dependencias de PHP.
- **Node.js y NPM:** Para manejar los assets front-end de Laravel.

1. Clonar el Repositorio

1. Primero, deben clonar el repositorio del proyecto en su máquina local. Para ello, abre una terminal o consola y ejecuta el siguiente comando:

git clone https://github.com/roier-code202/Proyecto-biblio_web.git

Navega al directorio del proyecto:

```
cd Proyecto-biblio_web
```

2. Configuración de Dependencias

2.1 Instalar Dependencias de PHP

1. Asegúrate de tener **Composer** instalado. Si no lo tienes, [descárgalo desde aquí](#).
2. Una vez que Composer esté instalado, ejecuta el siguiente comando dentro del directorio del proyecto para instalar las dependencias de Laravel:

```
composer install
```

2.2 Instalar Dependencias de Front-End

1. Asegúrate de tener **Node.js** y **NPM** instalados. Si no los tienes, [descárgalos desde aquí](#).
2. Después de instalar Node.js y NPM, ejecuta el siguiente comando para instalar las dependencias de front-end:

```
npm install
```

. Configuración del Archivo .env

1. Dentro del directorio del proyecto, encontrarás un archivo llamado. env.example. Cópialo y renómbralo como. env:

```
cp .env.example .env
```

Abre el archivo .env con un editor de texto y actualiza las variables relacionadas con la base de datos y otros servicios, como el correo, si es necesario:

- **DB_CONNECTION:** MySQL o el que uses.
- **DB_HOST:** Normalmente 127.0.0.1.
- **DB_PORT:** Normalmente 3306 para MySQL.
- **DB_DATABASE:** El nombre de la base de datos que usarás.
- **DB_USERNAME** y **DB_PASSWORD:** Las credenciales de acceso a tu base de datos.
- **4. Generar la Clave de la Aplicación**
- Laravel requiere una clave de aplicación que se puede generar con el siguiente comando:

```
php artisan key:generate
```

5. Migraciones de Base de Datos

1. Si el proyecto usa migraciones para configurar la base de datos, ejecuta las migraciones con el siguiente comando:

```
php artisan migrate
```

Si el proyecto tiene datos de ejemplo o seeders, ejecuta el siguiente comando para llenarlos en la base de datos:

```
php artisan db:seed
```

6. Compilar los Assets

Laravel usa Vite para gestionar los assets front-end. Para compilar los archivos de frontend (CSS, JS, etc.), ejecuta el siguiente comando:

```
npm run dev
```

Este comando compilará los archivos y los dejará disponibles para el proyecto. Puedes usar el siguiente comando para producción:

```
npm run build
```

7. Ejecutar el Servidor de Desarrollo

Una vez que las dependencias estén instaladas y los assets compilados, puedes iniciar el servidor de desarrollo de Laravel con el siguiente comando:

```
php artisan serve
```

Esto debería iniciar un servidor en `http://127.0.0.1:8000`. Abre esta URL en tu navegador para ver el proyecto en acción.

8. Verificar la Ejecución

1. En tu navegador, ve a la siguiente URL:

<http://127.0.0.1:8000>

1. Esto debería mostrar la página principal de la aplicación.

9. Realizar Cambios y Subir a GitHub

Si tus compañeros desean hacer cambios en el proyecto y subirlos a GitHub, pueden seguir estos pasos:

1. **Hacer cambios en el código.**
2. **Agregar los cambios a Git:**

```
git add .
```

Hacer commit de los cambios:

```
git commit -m "Descripción de los cambios"
```

Subir los cambios al repositorio remoto:

```
git push origin main
```

10. Resolución de Problemas Comunes

- **Error: No se puede acceder al servidor:** Asegúrate de que `php artisan serve` esté corriendo en el terminal.
- **Problemas con la base de datos:** Asegúrate de haber configurado correctamente el archivo `.env` y de que la base de datos esté corriendo.
- **Problemas con la compilación de assets:** Si tienes problemas al ejecutar `npm run dev`, asegúrate de tener Node.js y las dependencias correctamente instaladas.