Actividad 6 Diseño de Aplicaciones Web

José Luis Torres Gallegos 2997315

Parte 1

Investigar lo siguiente en la consola de comandos de Artisan:

- **El comando para generar una clave de aplicación.**
- **El comando para generar un servidor de desarrollo local.**
- **El comando para crear un enlace simbólico para la carpeta Almacenamiento.**
- **El comando para crear un controlador para el modelo Producto y con qué comando crear un controlador de recursos.**
- **El comando para crear una migración para el modelo de Producto.**
- **El comando para crear una sembradora para el modelo de Producto.**
- **El comando para crear una fábrica para el modelo de Producto.**
- **El comando para ejecutar migraciones, deshacer una migración, actualizar migraciones y ejecutar migraciones.**

Explica cada punto y la utilidad de cada uno de estos comandos con tus propias palabras. Integrar la información de un documento en formato Informe.

En el desarrollo de aplicaciones web utilizando el framework Laravel, la consola de Artisan proporciona una variedad de comandos útiles para agilizar el desarrollo, la administración y el mantenimiento del proyecto. A continuación, se detallan los comandos solicitados junto con su utilidad:

Comando para generar una clave de aplicación:

php artisan key:generate

Este comando genera una nueva clave de aplicación para la aplicación Laravel. Esta clave se utiliza para encriptar datos confidenciales y garantizar la seguridad de la aplicación.

Comando para generar un servidor de desarrollo local:

php artisan serve

Este comando inicia un servidor de desarrollo local que permite ejecutar y probar la aplicación Laravel en el entorno de desarrollo. Facilita el desarrollo al proporcionar un servidor web integrado sin necesidad de configuraciones adicionales.

Comando para crear un enlace simbólico para la carpeta Almacenamiento:

php artisan storage:link

Este comando crea un enlace simbólico desde la carpeta pública de la aplicación hacia la carpeta de almacenamiento. Es útil para acceder a los archivos almacenados de manera segura mediante URL públicas.

Comando para crear un controlador para el modelo Producto y con qué comando crear un controlador de recursos:

php artisan make:controller ProductoController

Este comando crea un nuevo controlador llamado ProductoController. Si se desea generar un controlador de recursos, se puede utilizar el siguiente comando:

php artisan make:controller ProductoController --resource

El controlador generado incluirá métodos predefinidos para realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) en el modelo de Producto.

Comando para crear una migración para el modelo de Producto:

php artisan make:migration create_productos_table --create=productos

Este comando crea un nuevo archivo de migración en el directorio de migraciones. Las migraciones se utilizan para definir y modificar la estructura de la base de datos. En este caso, se está creando una migración para crear una tabla llamada productos.

Comando para crear una sembradora para el modelo de Producto:

php artisan make:seeder ProductoSeeder

Este comando crea una clase seeder llamada ProductoSeeder. Los seeders se utilizan para poblar la base de datos con datos de prueba o iniciales. En este caso, se puede utilizar para agregar datos iniciales al modelo de Producto.

Comando para crear una fábrica para el modelo de Producto:

php artisan make:factory ProductoFactory --model=Producto

Este comando crea una fábrica para el modelo de Producto. Las fábricas se utilizan para generar datos de prueba de manera automatizada. La fábrica generará instancias del modelo de Producto con datos ficticios para su uso en pruebas o desarrollo.

Comando para ejecutar migraciones, deshacer una migración, actualizar migraciones y ejecutar migraciones:

php artisan migrate

Este comando ejecuta todas las migraciones que aún no se han ejecutado en la base de datos, creando las tablas y estructuras definidas en las migraciones.

Deshacer migración:

php artisan migrate:rollback

Este comando deshace la última migración realizada, revirtiendo los cambios en la base de datos.

Actualizar migraciones y ejecutar migraciones:

php artisan migrate:refresh

Este comando deshace todas las migraciones y las vuelve a ejecutar, lo que permite actualizar la estructura de la base de datos según las últimas migraciones definidas.

Ejecutar migraciones específicas:

php artisan migrate --path=/database/migrations/nombre_del_archivo_migration.php

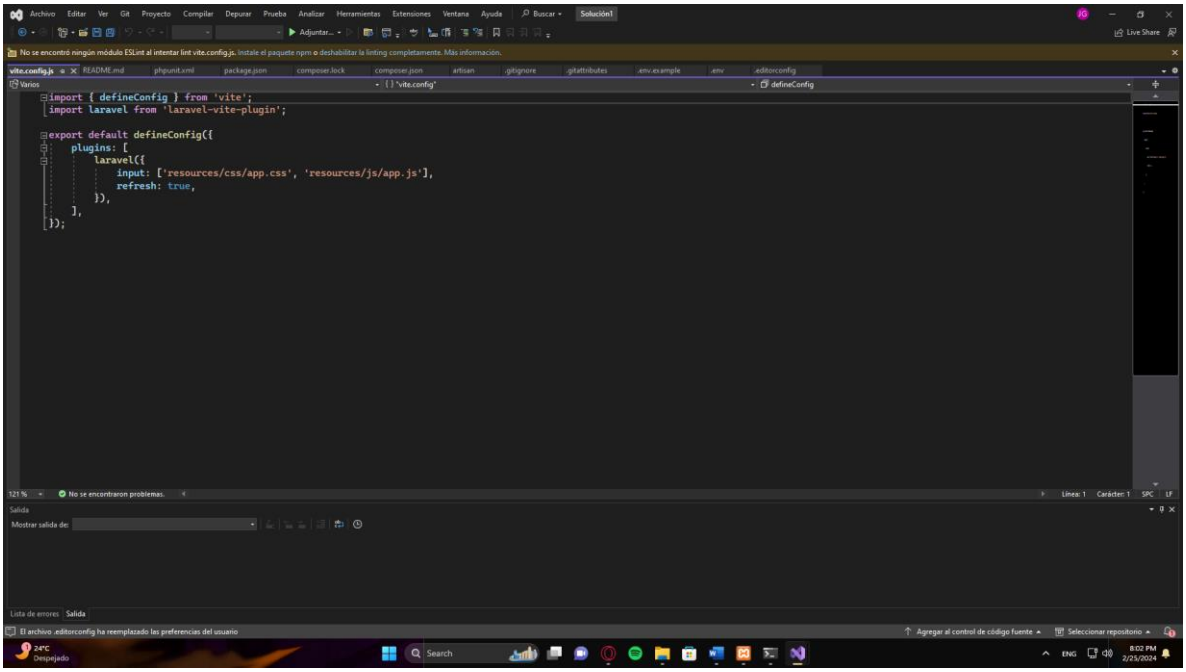
Este comando ejecuta una migración específica ubicada en una ruta determinada.

Parte 2

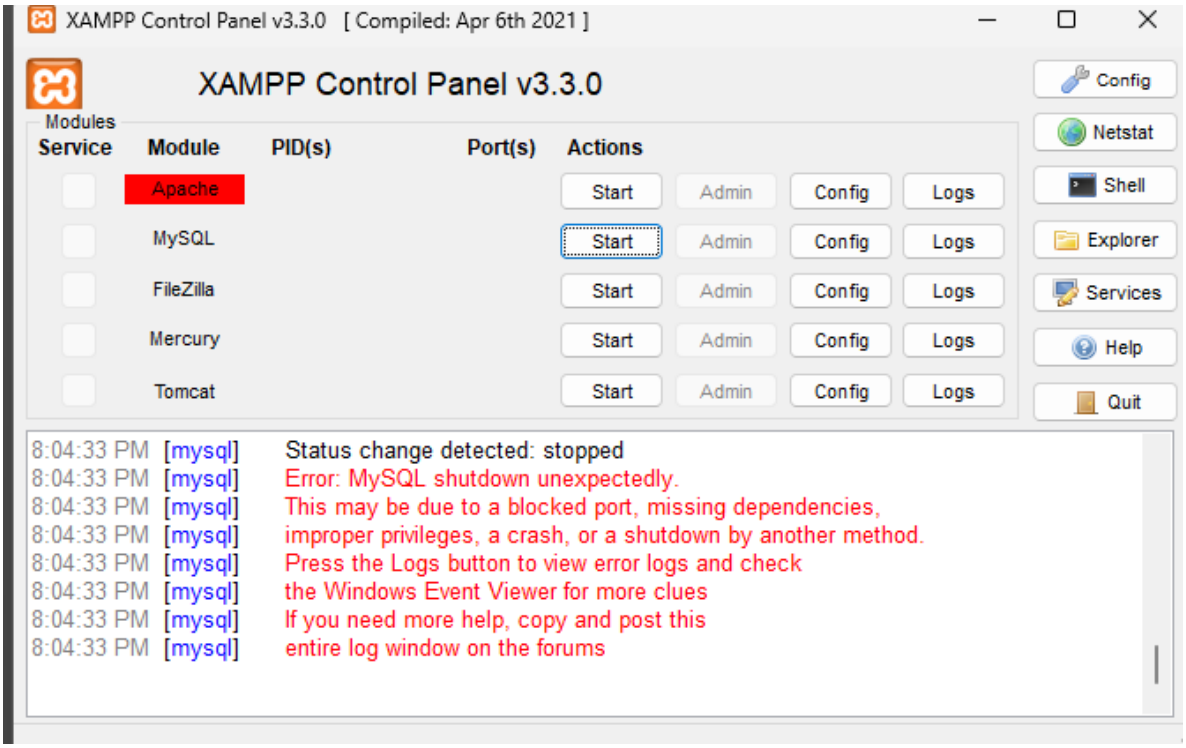
Investigar lo siguiente en la consola de comandos de Artisan:

- Crearás un modelo llamado Producto junto con su migración y controlador (debe ser un controlador de recursos).
- Cree una carpeta dentro de recursos/vistas llamada productos, dentro de esa carpeta cree los siguientes archivos:
- Un archivo llamado blade.php. Coloque dentro del archivo una etiqueta de título html con el texto "Vista del producto".
- Un archivo llamado blade.php. Coloca dentro del archivo una etiqueta html de tipo título y con el texto "Edición de Producto".
- Un archivo llamado blade.php. Coloca dentro del archivo una etiqueta html de tipo título y con el texto "Creación de Nuevo Producto".
- Inicializa un repositorio Git en tu proyecto y súbelo a GitHub, crea tu confirmación con la etiqueta "Productos base".
- Modifica tu archivo README.md colocando solo el título de la actividad.
- Copie la URL de su repositorio de la actividad e intégrela en el documento de informe que desarrolló en la primera parte de la actividad.

Tuve problemas con la instalación de Laravel



No me aparece como en el video la terminal, no se si sea porque uso el Visual Studio Community



No se muy bien a que se deban estos errores pero de igual manera le escribo aquí como se haría la segunda parte:

Estos comandos crean la estructura de carpetas y archivos para las vistas de productos.

Modificar archivos de vistas:

<title>Vista del producto</title>

Para edit.blade.php:

<h1>Edición de Producto</h1>

Para create.blade.php:

<h1>Creación de Nuevo Producto</h1>

Inicializar repositorio Git, hacer confirmación y subir a GitHub:

git init

git add .

git commit -m "Productos base"

git remote add origin <URL_del_repositorio_de_GitHub>

git push -u origin master