Logotipo, nombre de la empresa

Descripción generada automáticamente

|  |  |  |
| --- | --- | --- |
| **Nombre**:  Héctor Endy Pérez Navarro | | **Matrícula**:  03009066 |
| **Nombre del curso:**  Diseño de aplicaciones web | **Nombre del profesor**:  Prof. Cristopher Gerardo Gaytan Diaz | |
| **Módulo**: 1 | **Actividad**:Actividad 6 | |
| **Fecha**: 29-09-2024 | | |
| **Bibliografía**: | | |

**Parte 1:**

En la raíz de su proyecto "actividad-6", abrirá una terminal.

Con la ayuda de la consola de comandos de Artisan, investigarás lo siguiente:

* **El comando para generar una clave de aplicación.**

**php artisan key:generate:** Este comando genera una clave única para tu aplicación Laravel y la establece en el archivo .env. La clave es crucial para asegurar la sesión y otros datos encriptados, mejorando así la seguridad de la aplicación.

* **El comando para generar un servidor de desarrollo local.**

**php artisan serve:** Este comando inicia un servidor web de desarrollo que permite probar tu aplicación en un entorno local. De forma predeterminada, el servidor escucha en http://localhost:8000, facilitando la visualización de cambios en tiempo real.

* **El comando para crear un enlace simbólico para la carpeta Almacenamiento.**

**php artisan storage:link:** Este comando crea un enlace simbólico desde la carpeta public/storage a la carpeta storage/app/public, lo que permite acceder a archivos almacenados públicamente desde la aplicación. Es útil para servir imágenes y otros archivos de forma segura.

* **El comando para crear un modelo llamado**

**php artisan make:model NombreDelModelo:** Este comando genera un nuevo modelo de Eloquent. Por ejemplo, para el modelo Producto, usarías php artisan make:model Producto. Los modelos representan las tablas de la base de datos y son utilizados para interactuar con los datos de la aplicación.

* **El comando para crear un controlador para el modelo Producto y con qué comando crear un controlador de recursos.**

**php artisan make:controller ProductoController –resource: El** primer comando crea un controlador básico para manejar la lógica de negocio relacionada con el modelo Producto. El segundo comando crea un controlador de recursos que incluye métodos predefinidos para operaciones CRUD (Crear, Leer, Actualizar, Borrar), facilitando la gestión de recursos en la aplicación.

* **El comando para crear una migración para el modelo de Producto.**

**php artisan make:migration create\_productos\_table:** Este comando genera un archivo de migración en la carpeta database/migrations que se utiliza para definir la estructura de la tabla productos en la base de datos. Las migraciones permiten versionar la base de datos y facilitar su actualización.

* **El comando para crear una sembradora para el modelo de Producto.**

**php artisan make:seeder ProductoSeeder:** Este comando crea un archivo de sembradora que se utiliza para poblar la base de datos con datos de prueba para el modelo Producto. Las sembradoras son útiles para generar datos iniciales y para pruebas.

* **El comando para crear una fábrica para el modelo de Producto.**

**php artisan make:factory ProductoFactory:** Este comando genera un archivo de fábrica que se utiliza para crear instancias del modelo Producto con datos ficticios. Las fábricas son útiles para pruebas y desarrollo, ya que permiten generar datos de manera rápida y consistente.

* **El comando para ejecutar migraciones**

**php artisan migrate:** Este comando ejecuta todas las migraciones pendientes que aún no se han aplicado a la base de datos. Crea las tablas y columnas definidas en los archivos de migración ubicados en la carpeta database/migrations. Es esencial para aplicar la estructura de la base de datos según lo definido por los desarrolladores.

* **deshacer una migración**

**php artisan migrate:rollback:** Este comando revierte la última "lote" de migraciones que se ejecutaron. Si deseas deshacer los cambios realizados por la última migración, este comando es muy útil. Se puede utilizar para deshacer cambios no deseados y revertir la base de datos a su estado anterior.

* **actualizar migraciones**

**php artisan migrate –force:** Este comando aplica migraciones en un entorno de producción. Al usar la opción --force, le indicas a Laravel que ejecute las migraciones sin pedir confirmación. Es importante tener cuidado al usar este comando, ya que aplicará todos los cambios pendientes a la base de datos de producción.

* **ejecutar migraciones**

**php artisan migrate path=/database/migrations/2023\_01\_01\_000000\_create\_productos\_table.php:** Este comando permite ejecutar migraciones específicas al indicar la ruta del archivo de migración que deseas aplicar. Es útil cuando solo quieres aplicar cambios de una migración en particular sin afectar otras migraciones. Esto proporciona un control más granular sobre el proceso de migración.

* 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.

**Parte 2:**

Con la ayuda de la terminal de comandos de Artisan, en la raíz de tu proyecto harás lo siguiente:

* Crearás un modelo llamado Producto junto con su migración y controlador (debe ser un controlador de recursos).

php artisan make:model Producto -m

php artisan make:controller ProductoController --resource

* 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".

Texto

Descripción generada automáticamente

* Un archivo llamado blade.php. Coloca dentro del archivo una etiqueta html de tipo título y con el texto "Edición de Producto".

Texto

Descripción generada automáticamente

* Un archivo llamado blade.php. Coloca dentro del archivo una etiqueta html de tipo título y con el texto "Creación de Nuevo Producto".

Captura de pantalla de computadora

Descripción generada automáticamente

* 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.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

* 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.

[navbarro/Actividad6 (github.com)](https://github.com/navbarro/Actividad6)