Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Unidad 2: Actividad de la Unidad

Asignatura: Framework y Programación Web

Docente: Víctor Cofré Farías

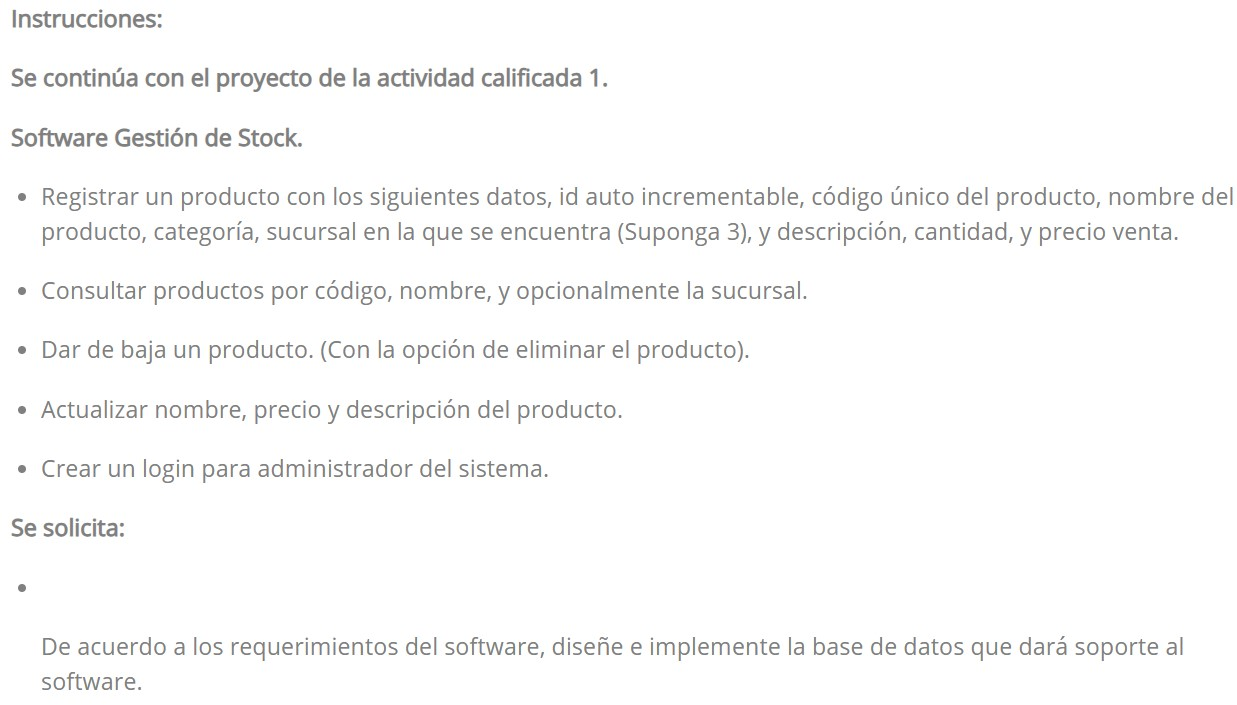
Alumnos: Mauricio Alejandro Oxman Oliva

Eliú Martínez Pincheira

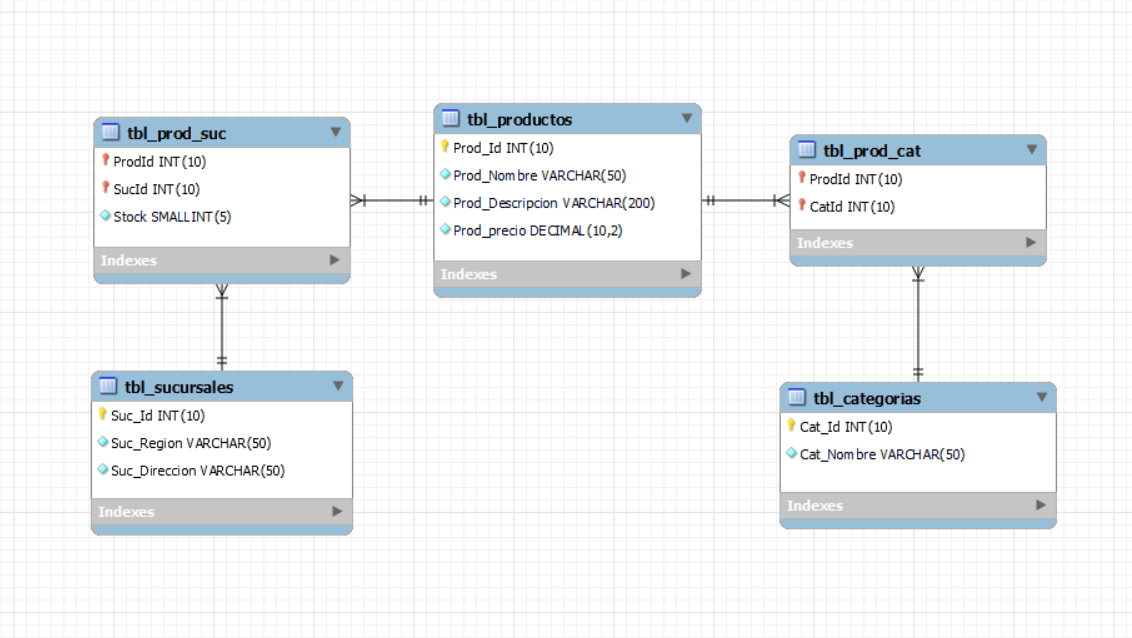
Sección y Grupo: Sección 50

Fecha de Entrega: 20/11/2023

Caso a Resolver



1. **Modelo de Base de Datos de Proyecto**



Explicación breve de modelo de base de datos

Como podemos ver, el modelo creado consta de 5 tablas, la cual, incluye las 3 tablas principales para el proyecto: **sucursales, productos y categorías**.

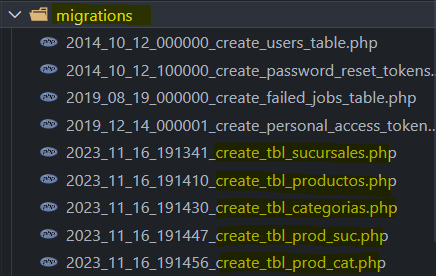
Ahora veremos sus relaciones y explicación del modelo:

En **una sucursal** puede haber **muchos productos** y **un producto** puede estar en **muchas sucursales**, por lo tanto, tenemos una relación de muchos a muchos, por lo tanto, se crea la tabla: **tbl\_prod\_suc**, en esta agregamos el campo **stock**, la que nos permitirá saber los stocks de un producto en una determina sucursal.

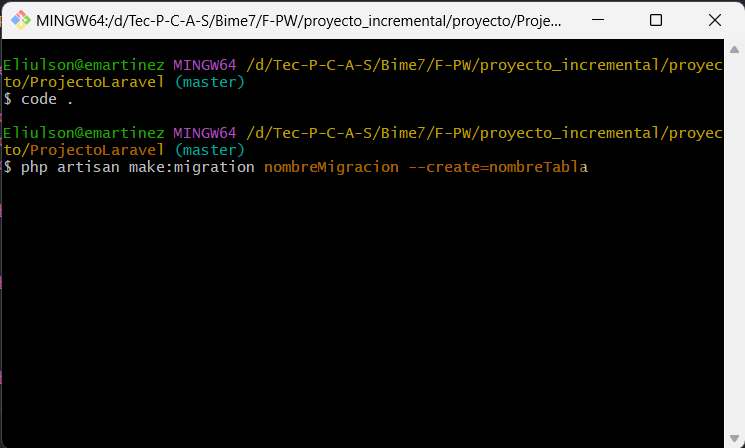
También tenemos la relación de productos y categorías, muy similar a la anterior. **Un producto** puede tener **más de una categoría** y **un tipo de categoría** puede estar en **muchos productos**. Ejemplo: el producto reloj de pared, puede pertenecer a la categoría de hogar y a la categoría de adorno, y viceversa, la categoría electrónica la pueden tener los productos tales como: celular, televisión, audífonos, etc.

Ahora que conocemos el modelo, nos adentramos a la continuación del proyecto, esto incluirá migraciones, seeders y modelos.

1. **Migraciones en el Proyecto**



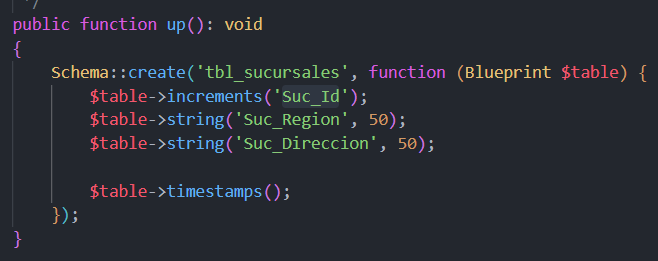
Ahora veremos como creamos cada una de las migraciones necesarias, para el modelo.



Abrimos la terminal que vamos a usar, el IDE (vs code en este caso) y nos debemos posicionar en el proyecto con marco de trabajo Laravel.

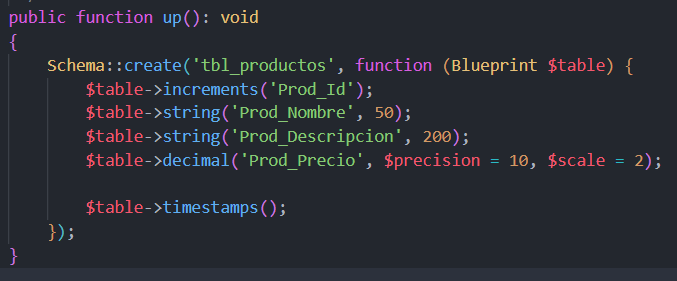
Luego ejecutar el comando mostrado en pantalla, cambiándole **nombreMigracion**, por el nombre de nuestra migración y **--create=nombreTabla.** Por el nombre de la tabla a la que se relaciona en nuestro modelo.

De esta manera creamos todas nuestras migraciones que necesitamos.



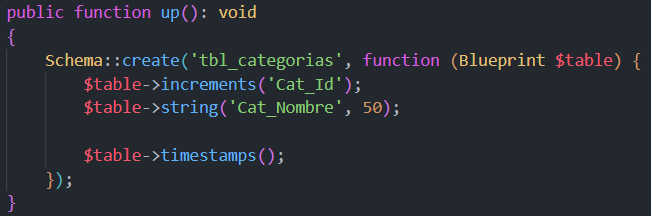
Como primera instancia, veremos nuestra migración que hace referencia a la **tbl\_sucursales**. Nos enfocaremos en el método **up()**, ya que, este nos permitirá crear las columnas y datos de nuestra tabla.

Nuestra tabla sucursal consta de 3 columnas, y la timestamps por defecto. **Increments()**, lo que hará es: **crear un id auto-incrementable de dato unsignedInteger**. Luego tenemos los campos strings y con su longitud como segundo parámetro.

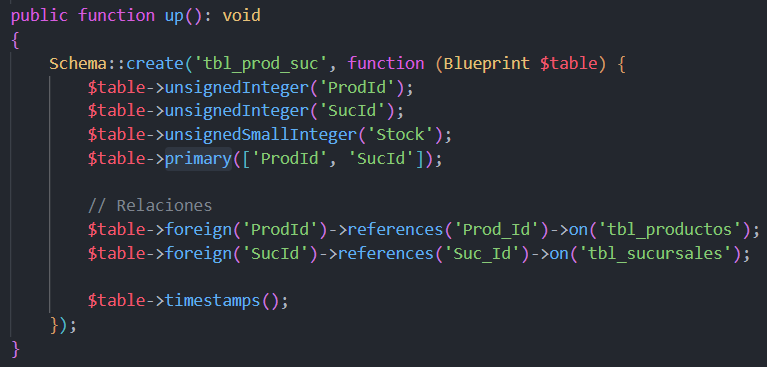


Tenemos la migración de la **tbl\_productos**, tenemos 4 columnas y la timestamps.

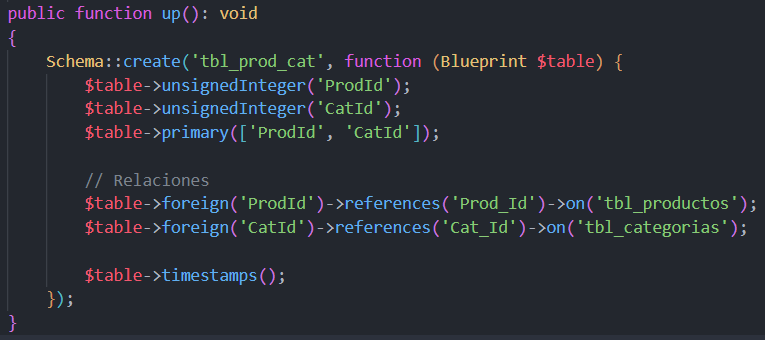
Una de ellas sería nuestro id **‘increments()’**, 2 columnas strings con sus longitudes, y la columna decimal, que incluye 2 parámetros, el primero es el total de dígitos numéricos y el segundo es el total de decimales que tendrá.



Tenemos la migración de la **tbl\_categorias**, tenemos 2 columnas y la timestamps. Tenemos el id **‘increments()’** y una columna string con su longitud.



Tenemos la migración de la **tbl\_prod\_suc**, incluye 3 columnas y la timestamps. Esta tabla refleja una relación de muchos a muchos, por lo que, contendrá 2 id, que hacen referencia a las tablas involucradas. Primero se debe crear los campos que serán las llaves foráneas, estas deben ser del mismo tipo de dato que las llaves primarias **‘unsignedInteger’**, también un campo **‘unsignedSmallInteger’** para el Stock de los productos en una sucursal determinada. Creamos los campos foráneos, pero, estos serán **ids**, por lo tanto, se agregarán en la función **‘primary()’**. Finalmente debemos crear las relaciones foráneas que, incluye la sintaxis mostrada, primero va el dato de la llave foránea, luego, viene el dato de la llave primaria, y tercero la tabla que contendrá la llave primaria.



Tenemos la migración de la **tbl\_prod\_cat**, incluye 2 campos y la timestamps. Creamos las llaves foráneas del mismo tipo de dato que las primarias **‘unsignedInteger’**, luego, indicamos que son **ids** con **‘primary()’** y creamos las relaciones con la sintaxis mostrada.

Luego que creamos las migraciones, podemos inicializarlas, comandos básicos:

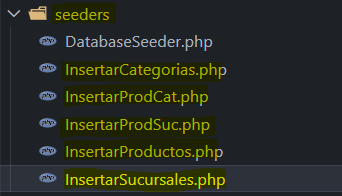
**‘php artisan migrate** o **php artisan migrate:install’**, nos sirve para iniciar la migración.

**‘php artisan migrate:restart’**, realiza una limpieza de las migraciones.

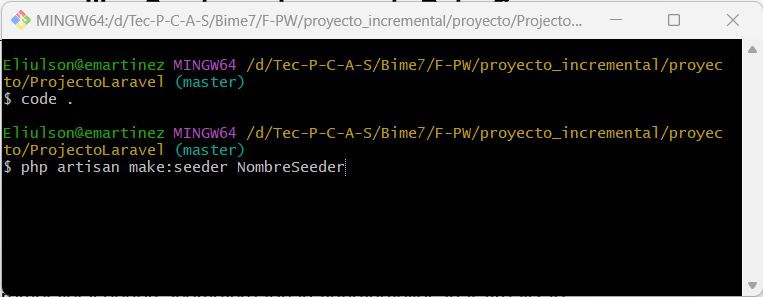
**‘php artisan migrate:refresh’**, realiza una limpieza y luego inicia las migraciones.

Al final del proyecto, veremos el proceso completo configurando la base de datos, para luego, realizar las migraciones y el ingreso de datos con los seeders.

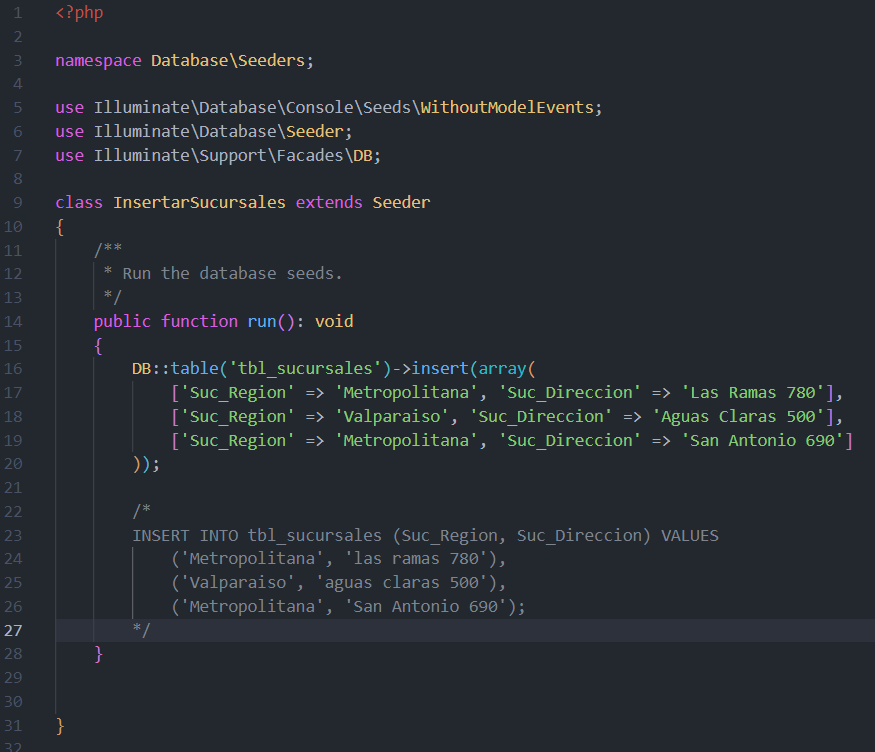
1. **Seeders e Ingreso de Datos**



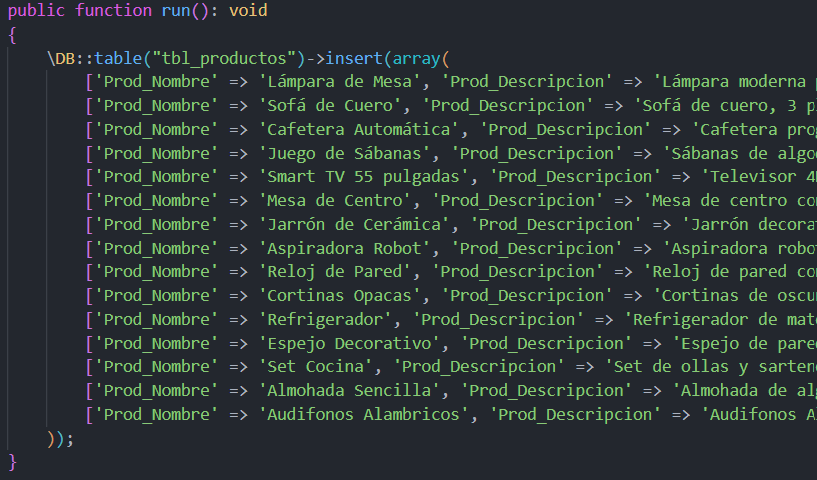
Ahora veremos los seeders, como crear cada uno de ellos, para realizar inserciones a nuestra base de datos.



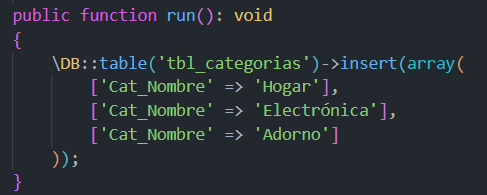
Para crear un seeder, se debe realizar la sintaxis mostrada, en la terminal que estemos utilizando y en el directorio de nuestro proyecto Laravel.



Tenemos el seeder **InsertarSucursales**, en este caso usamos una librería, que nos provee Laravel **‘use Illuminate\Support\Facades\DB’**, Con este podremos ingresar a la base de datos a través de la función: **‘run()’**, la sintaxis para conectarse es la mostrada, donde se indica que nos conectamos a la **DB,** a la tabla sucursales (en este caso), para realizar una inserción de datos, que a la vez se ingresa un arreglo, ya que, un registro de base de datos normalmente contiene más de un dato, luego se ingresa el nombre del campo de la tabla y luego el valor.



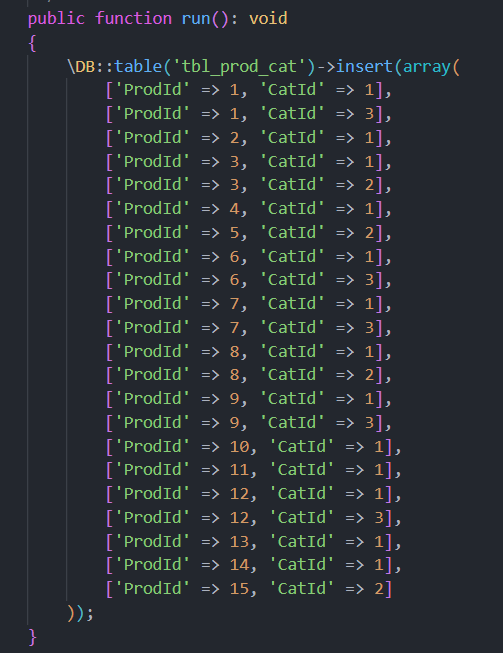
Tenemos el seeder **InsertarProductos**, esta vez no utilizamos la librería de Laravel y nos enfocamos en la función **‘run()’**, debido a que, dentro de esta función nos conectamos a la base de datos y realizamos las inserciones. La única diferencia para conectarnos esta vez a la base de datos es que, debemos agregar el signo **‘\’**. Aquí nos conectamos a la tabla productos y realizamos las inserciones de sus campos.



Tenemos el seeder **InsertarCategorias**, nos conectamos e insertamos datos a la tabla: **tbl\_categorias**.



Tenemos el seeder **InsertarProdSuc**, nos conectamos e insertamos datos a la tabla: **tbl\_prod\_suc**.

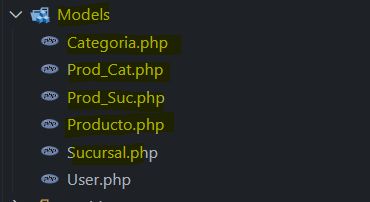


Tenemos el seeder **InsertarProdCat**, nos conectamos e insertamos datos a la tabla: **tbl\_prod\_cat**.

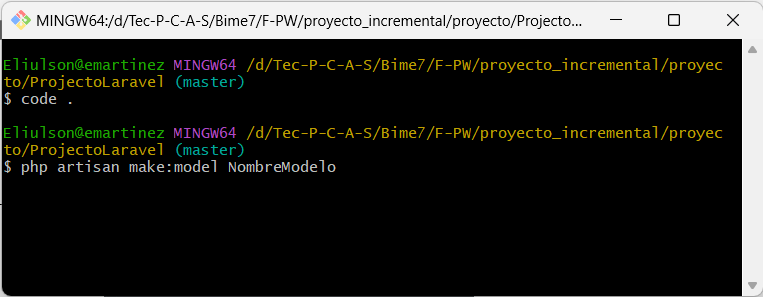
Finalmente tenemos los comandos para realizar las inserciones que los veremos al final:

**php artisan db:seed --class=NombreSeeder**, este comando nos permitirá realizar la inserción a la base de datos desde un seeder.

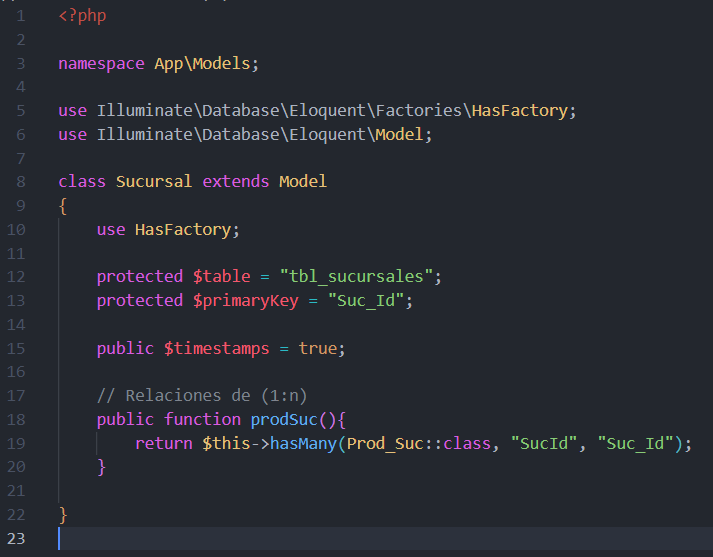
1. **Modelos del Proyecto**



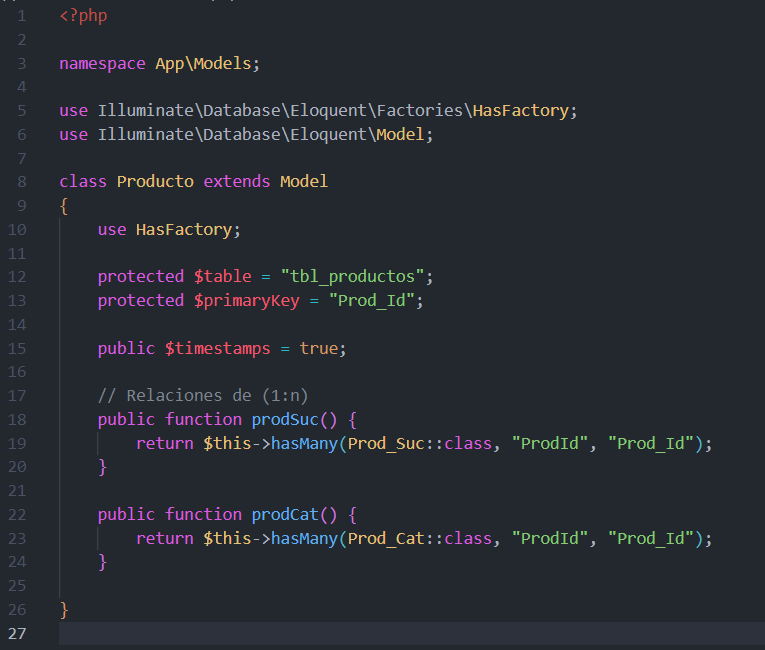
A continuación, veremos los modelos creados para el proyecto.



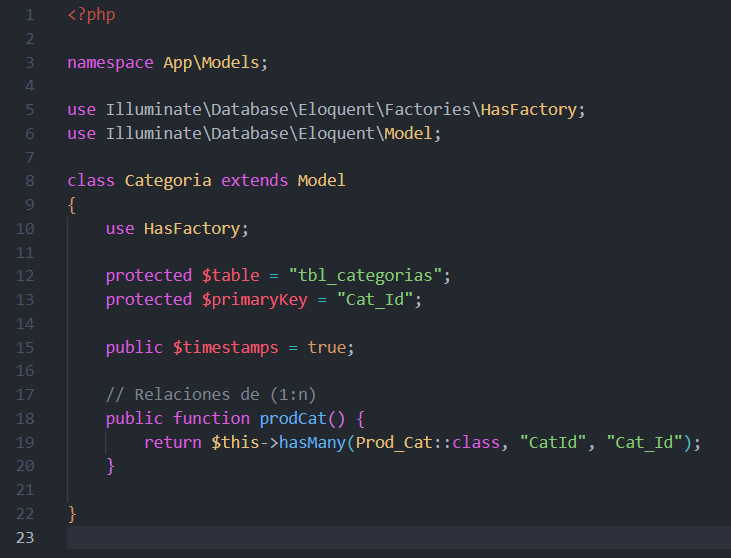
Con la sintaxis anterior, podemos crear los modelos que nos permitirán comunicarnos con la base de datos que, se utilizará para el proyecto.



Tenemos el modelo **Sucursal**, en este creamos el nombre de la tabla, su id, y la función que, hace referencia de a la relación (1:n): **una sucursal puede tener muchos productos,** relacionándose con el modelo **Prod\_Suc**.



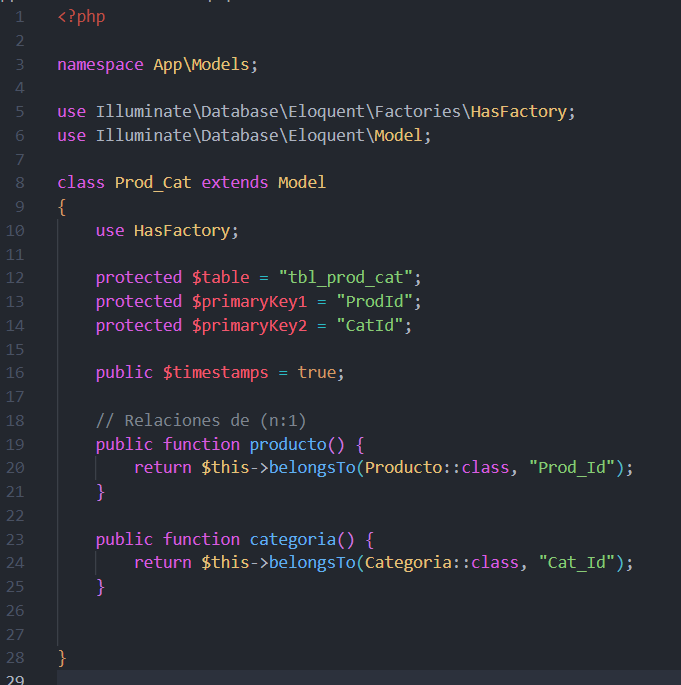
Tenemos el modelo **Producto**, creamos el nombre de la tabla, el id de la tabla, y luego creamos las funciones que hacen referencias a las relaciones que tiene nuestra tabla. Tiene la relación con sucursal, donde, **un producto** puede estar en **muchas sucursales**, es por eso que se relaciona con el modelo **Prod\_Suc**, y también, tiene la relación con categorías donde, **un producto** puede tener **más de una categoría**, es por eso que se relaciona con el modelo **Prod\_Cat**.



Tenemos el modelo **Categoria**, creamos el nombre de la tabla, el id de la tabla y la función que hace referencia a la relación de la tabla, en este caso, nuevamente tenemos una relación de (1:n) donde, **un tipo de categoría** puede estar en **muchos productos**, es por eso la relación con el modelo **Prod\_Cat**.



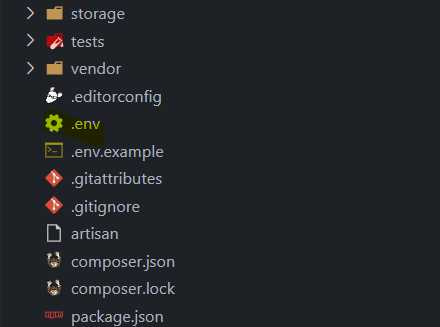
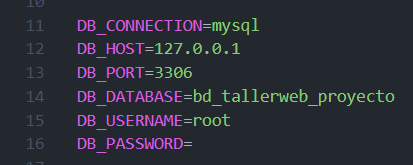
Aquí tenemos el modelo de **Prod\_Suc**, este incluye el nombre de la tabla, los ids de las tablas, ya que, es una tabla intermedia que representa una relación (n:n), por lo tanto, la clave primaria es compuesta, y luego tenemos las funciones que hacen referencias a estas relaciones. Primero (n:1), tenemos que **muchos productos** le pueden pertenecer **a una misma sucursal**, por eso la relación con el modelo **Sucursal**. Segundo (n:1), que **muchas sucursales** pueden tener **el mismo producto** por eso la relación con el modelo **Producto**.



Para concluir tenemos el modelo **Prod\_Cat**, este incluye el nombre de la tabla, los ids de las tablas, ya que, es una tabla intermedia con relación (n:n) y las funciones que hacen referencias a estas relaciones. Primero (n:1), **un conjunto de categorías** las puede tener **un mismo producto**, por eso la relación con el modelo **Producto**. Segundo (n:1), **un conjunto de productos**, pueden compartir **una misma categoría**, por eso la relación con el modelo de **Categoria**.

1. **Prueba de Migraciones/Seeders**

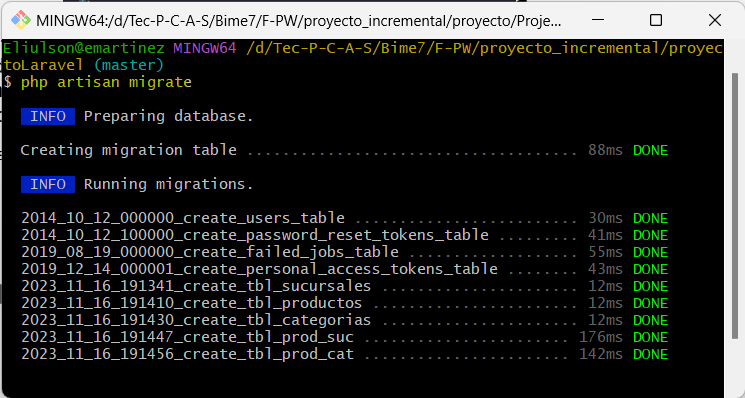
Vamos a realizar una prueba con la arquitectura que tenemos creada, para eso configuramos nuestras variables de entornos y configuraciones de la base de datos.

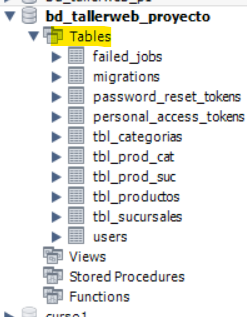
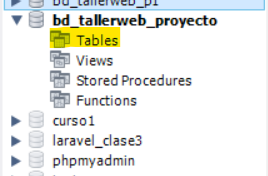
 

Para configurar las variables de entorno nos dirigimos al archivo **.env** y luego nos enfocamos en las líneas 11-16, esta será nuestra configuración inicial de las base de datos, donde, veremos la **conexión, host, puerto, base de datos, nombre de usuario y contraseña**, normalmente tenemos las conexiones por defecto, y nos enfocamos en el nombre de la base de datos, que creamos para realizar las migraciones e inserciones.

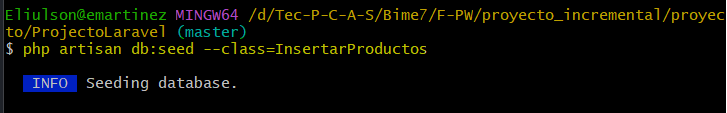


También nos dirigimos a nuestro archivo **database.php**, que se encuentra en nuestro directorio **config**, para realizar los ajustes en las propiedades**: charset, collation y engine**. De esta manera, nos encontramos en las condiciones para realizar los comandos de migraciones e inserciones a la base de datos creada.



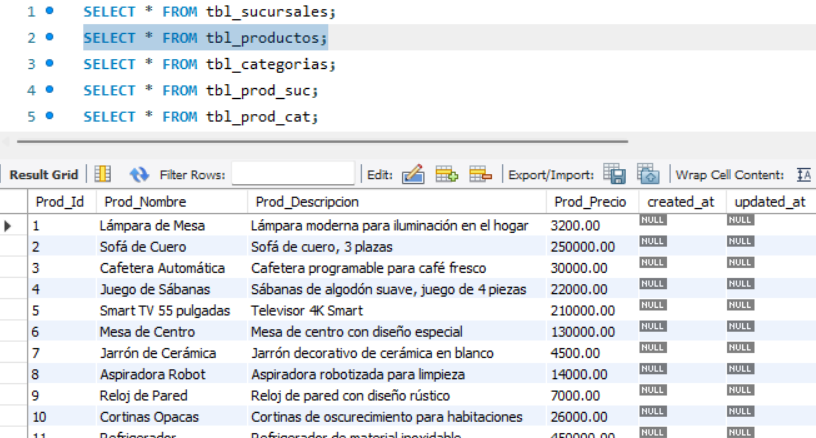
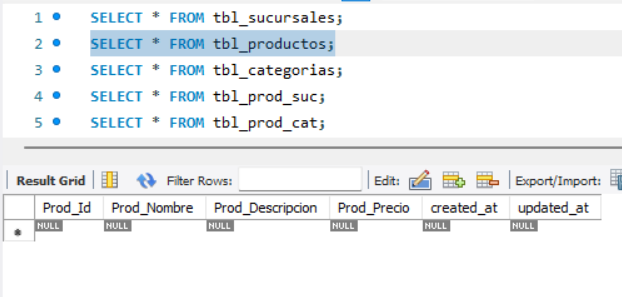


Realizamos el comando para ejecutar las migraciones y vemos nuestra base de datos antes de las migraciones sin ninguna tabla y después de las migraciones con las tablas creadas.



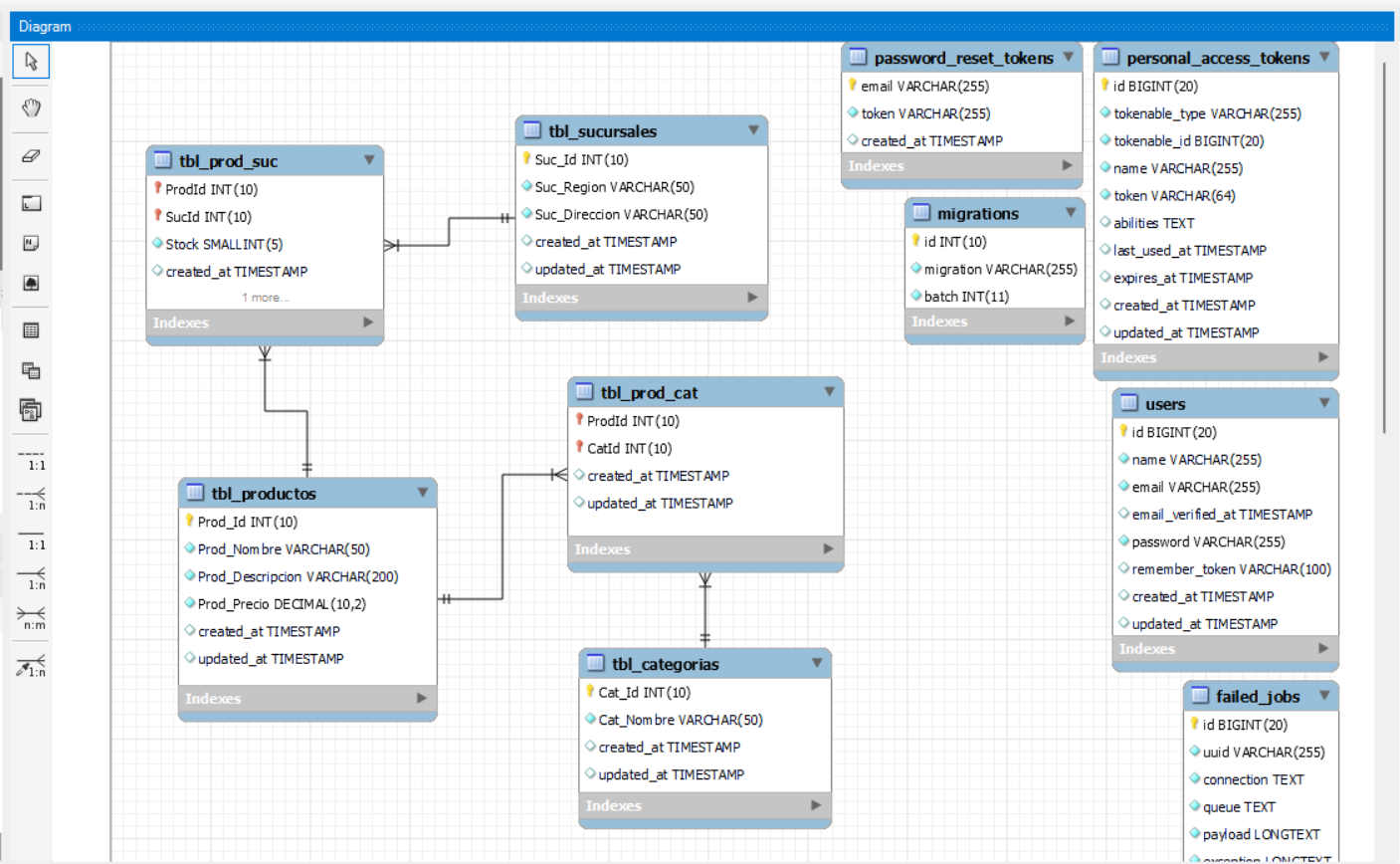
Realizamos el comando para insertar datos, a través de los seeders, un recordatorio importante, las inserciones con los seeders, al igual que la creación de las migraciones, deben ser en orden, o de otra manera, nos dará error, ya que, no podemos relacionar datos entre sucursales y productos, si no hemos realizado el ingreso de productos y sucursales antes.

Aquí tenemos un ejemplo de los registros de la base de datos antes y después del comando que nos permite hacer la inserción:



Una vez ingresamos los comandos de las migraciones y seeder, con **MySQL workbench**, podremos crear el modelo de la base de datos que será similar a la inicial de este informe.

1. **Modelo Creado con Migraciones y Seeders**



1. **Conclusiones**

Hemos podido realizar con éxito la implementación de la base de datos: desde la integración de migraciones, seeders y modelos, para posteriormente darle el uso en el proyecto final, se adjunta archivos y enlace de Git.

Enlace de Git:

https://github.com/gasdar/LaravelMyLogo