

## Proyecto Final

### Curso de Laravel

A continuación, se presentan todas las condiciones del proyecto final de la materia de Programación web con el *framework* de Laravel, la idea de este proyecto es poner en práctica todos los conceptos vistos en clase. Los temas son:

- MVC usando el *framework* de Laravel.
- Vistas y *Layouts* usando el *React* con la librería [Laravel Breeze](#).
- Creación de migraciones de base de datos.
- Creación de *factories* y archivos semilla de bases de datos (*Seeder*).
- Autenticación de usuarios con la librería [Laravel Breeze](#).
- Envío de correo electrónico.
- Uso de enrutamiento. Uso del *Model Binding*.
- Uso de controladores de tipo recurso para *CRUD*.
- Uso de controladores de tipo recurso API para *CRUD*.
- Uso del ORM de *eloquent* para conexión a la base de datos y el enlace con los modelos del programa.
- Relaciones entre modelos de *Eloquent*.
- Carga de imágenes como representación gráfica de los productos.
- Uso de *Slug* en los productos.

El proyecto consiste en crear una plataforma web para venta en línea de productos, esta plataforma de E-Commerce tendrá que tener dos puntos de entrada, la que ve el usuario ( "/" ruta raíz) y un módulo administrativo en donde se alimente la información ("/login" o "/home"). Cada estudiante podrá elegir un modelo de negocio diferente, por lo que tendrá que ofrecer diferentes productos con diferentes atributos en su plataforma web final. Las rutas mínimas del proyecto son:

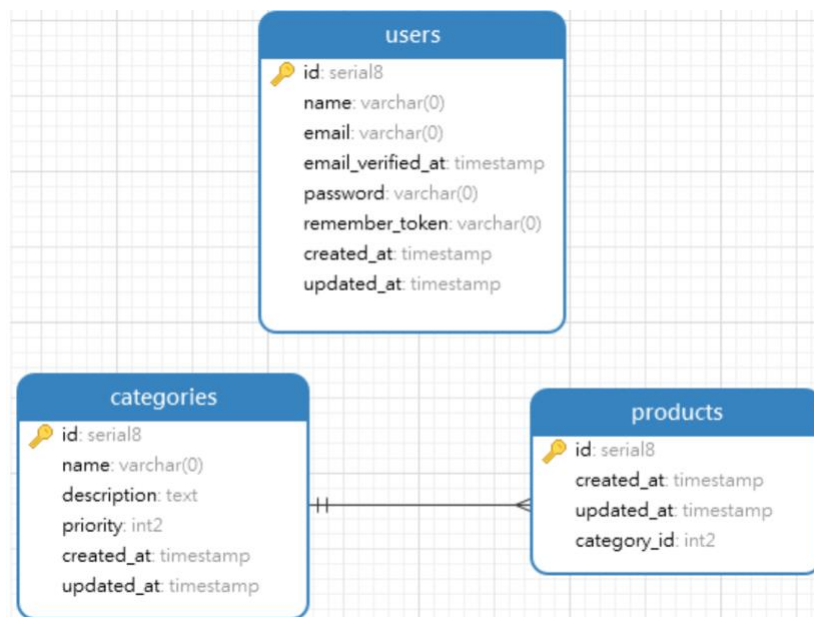
- Vista de productos disponibles en la página ("/"). En esta ruta se deben mostrar las categorías creadas junto a la totalidad de los productos que se tienen disponibles en la plataforma. La tabla de productos debe quedar con al menos 5 campos con 4 tipos de dato diferentes.
- Vista de un sólo producto ("/products/{slug}"). En esta ruta se debe visualizar el detalle de un sólo producto, mostrar todas sus características.
- Vista de login ("/login"). Permite la autenticación de un usuario, se recomienda el uso de la interfaz que ya trae por defecto el *framework* implementado.
- Vista de categorías ("/categories"). Los productos tendrán una categoría (Ver modelo entidad relación) de la cual se debe permitir realizar el *CRUD*. Esta categoría se debe ver en la página principal del usuario.
- Vista de productos ("/products"). En esta sección se debe permitir el *CRUD* de productos, estos serán los que se muestran en la página principal.
- *CRUD* vía API de categorías ("/api/categories") con autenticación de usuario usando Token. En los métodos para obtener una o todas las categorías se debe mostrar el arreglo de productos que tiene relacionados.

- CRUD vía API de productos (“**/api/products**”) con autenticación de usuario usando Token. En los métodos para obtener uno o todos los productos se debe mostrar la categoría relacionada del producto.

Las condiciones generales del proyecto se presentan a continuación:

- Se debe usar el *framework* de Laravel en su versión 11.x o superior para desarrollar la aplicación web. Se deben crear las migraciones de base de datos y los datos semilla para los tres modelos presentados.
- Se debe usar un *framework* de *frontend* como *Bootstrap*, *materialize* o *tailwind* para facilitar la implementación del diseño del aplicativo. El aplicativo debe ser *responsive design* (Que sea compatible con diferentes tamaños de pantalla).
- Se deben crear los controladores de tipo recurso para los modelos “**users**”, “**categories**” y “**products**” además de crear métodos adicionales que pueda necesitar y enlazarlos con las rutas correspondientes. Para todos los CRUD (Ya sean gráficos o tipo API) se deben realizar validaciones de los datos obtenidos validando que se encuentren en el formato correcto y no se debe perder información en caso de no pasar la validación.
- Para el ingreso al módulo administrativo es indispensable tener autenticación de usuario, todas las rutas de este módulo deben estar protegidas y la creación de usuarios debe ser únicamente cuando una persona está *logueada*.
- Se debe definir mínimo en un punto del aplicativo el envío de correos electrónicos para notificar sobre alguna acción. Se recomienda el uso de *mailtrap* para validar que los correos electrónicos estén saliendo correctamente.
- Las rutas de *products* y *categories* deben tener como parámetro el campo *slug* en lugar del campo *id* tanto para las interfaces gráficas como para las rutas de tipo API.

A continuación, se presenta el modelo entidad relación general de la aplicación:



Como se puede observar, la tabla *products* sólo cuenta con los datos básicos de Laravel y con la llave foránea de la tabla “*categories*”, esto es debido a que cada estudiante debe tener diferentes campos definidos para el modelo de negocio elegido.