

SENA

3197815

Instructora

Javier Leonardo Pineda Uribe

Aprendiz

Juan Pablo Hernández Cadenas

Competencia

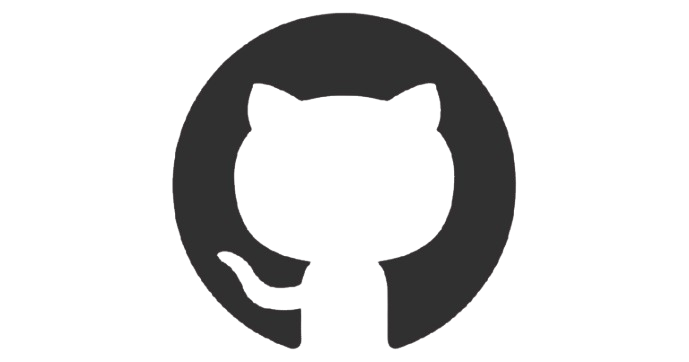
Aplicación de la calidad de software

19/10/2025

# DOCUMENTO DE ARQUITECTURA DE SOFTWARE

Proyecto "Moterium"

|  |  |  |  |
| --- | --- | --- | --- |
| Versión | Fecha | Autor | Descripción |
| 1.0 | 2025-10-19 | Juan Pablo Hernandez | Estructura base de la aplicación y definición arquitectónica inicial. |

Repositorio Git Hub   
[](https://acortar.link/q2L6Tq)

## 1. INTRODUCCIÓN Y OBJETIVO

El presente documento detalla la arquitectura de software del proyecto Tienda, un sistema de gestión y visualización de productos basado en el patrón Modelo-Vista-Controlador (MVC). El objetivo es establecer un marco técnico sólido, escalable y mantenible para el desarrollo de futuras funcionalidades de comercio electrónico.

## 2. ARQUITECTURA DEL SISTEMA

### 2.1. Patrón Arquitectónico Principal

El sistema se fundamenta en la arquitectura Modelo-Vista-Controlador (MVC), impuesta por el framework Laravel.

|  |  |  |
| --- | --- | --- |
| Componente | Definición en Laravel | Responsabilidad |
| Modelo | Clases en app/Models (Eloquent ORM) | Encapsula la lógica de negocio y la interacción con la base de datos (MySQL). Define relaciones y restricciones de datos. |
| Vista | Archivos Blade en resources/views | Capa de presentación. Contiene el HTML, la integración de Bootstrap, CSS/JS y el uso de animaciones AOS. |
| Controlador | Clases en app/Http/Controllers | Gestiona las peticiones HTTP, coordina los Modelos, ejecuta la lógica de control y selecciona la Vista a retornar. |

### 2.2. Tecnologías y Stack

|  |  |  |
| --- | --- | --- |
| Categoría | Tecnología | Rol Arquitectónico |
| Lenguaje Back-end | PHP 8.x | Lógica de servidor y core de la aplicación. |
| Framework | Laravel (Versión LTS recomendada) | Orquestación MVC, enrutamiento, ORM (Eloquent) y gestión de sesiones/autenticación. |
| Base de Datos | MySQL | Persistencia de datos (esquema relacional). |
| Frontend Base | JavaScript (Puro), CSS3 | Interactividad y estilo. |
| Framework CSS/UI | Bootstrap 5 | Diseño responsive y componentes predefinidos. |
| Animación | AOS (Animate On Scroll) | Mejora la experiencia de usuario con efectos visuales. |

## 3. VISTA DE COMPONENTES FUNCIONALES

El proyecto está segmentado en módulos clave que corresponden a áreas funcionales y a la estructura de directorios de Laravel.

### 3.1. Módulo de Peticiones HTTP (Rutas y Controladores)

|  |  |  |
| --- | --- | --- |
| Componente | Directorio | Descripción |
| Enrutamiento | routes/web.php | Define la URL de la aplicación (/, /login, /dashboard). |
| Middleware | app/Http/Middleware | Filtros de peticiones, utilizados principalmente para la Autenticación simulada y protección CSRF. |
| Controladores | app/Http/Controllers | Implementación de la lógica para el CRUD de Productos y Categorías, además del flujo de Login/Logout. |

### 3.2. Módulo de Datos (Modelos y Persistencia)

|  |  |  |
| --- | --- | --- |
| Componente | Directorio | Descripción |
| Modelos | app/Models | Clases Product y Category con sus respectivas relaciones Eloquent. |
| Migraciones | database/migrations | Control de versiones del esquema de base de datos MySQL (tablas de productos y categorías). |
| Configuración DB | .env / config/database.php | Parámetros de conexión a la instancia de MySQL. |

## 4. GESTIÓN DE LA EXPERIENCIA DE USUARIO (UX/UI)

El diseño se centra en dos áreas con estéticas diferenciadas:

### 4.1. Estética General y Branding

Paleta de Colores: La interfaz del Dashboard se adhiere a una combinación de tonalidades rojas y negras, ofreciendo un look moderno y de alto contraste.

Responsive Design: Asegurado mediante la integración y el uso de las clases de Bootstrap 5.

### 4.2. Flujo de Vistas Clave

|  |  |  |
| --- | --- | --- |
| Petición | Ruta | Nivel de Acceso / Componentes Involucrados |
| Index Principal | / (Público) | index.blade.php, ProductController (mostrar listado de productos). |
| Login | /login (Público) | login.blade.php, AuthSimuladaController (manejo de credenciales). |
| Dashboard | /dashboard (Restringido) | dashboard.blade.php, Controladores de gestión (Productos/Categorías). |

### 4.3. Assets y Front-end

Todos los recursos estáticos (CSS, JS, Imágenes) se gestionan a través del directorio public/ASSETS y se referencian mediante la función asset() de Laravel para garantizar rutas consistentes en el despliegue.

## 5. RECURSOS INICIALES Y ESQUEMA DE BASE DE DATOS

### 5.1. Recursos Iniciales

El proyecto incluye los archivos y directorios bases necesarios para comenzar el desarrollo:

- Archivos de configuración de Laravel (config/).  
- Estructura de scaffolding de las vistas principales.  
- Implementación inicial de los recursos frontend (Bootstrap, JS, CSS).

### 5.2. Esquema de Base de Datos (Estructura de Tablas)

|  |  |  |
| --- | --- | --- |
| Tabla | Propósito | Relación Clave |
| Categorias | Almacena la información de las categorías de productos. | Ninguna |
| Productos | Almacena los productos de la tienda. | category\_id (Clave Foránea a categories) |
| users | Almacena las credenciales para la autenticación simulada. | Ninguna (Autenticación interna de Laravel) |