
Diseño y Arquitectura

Proyecto: Sistema de Gestión de una Tienda en Línea.
Análisis y Diseño 2.



Diciembre del 2023

Integrantes de Diseño.

Grupo #

Fernando José Jr. Serrano Mejía

20171039

Sergio Mynor David Felipe Zapeta

200715274

Carlos Esteban Godinez Delgado

200819214

Índice

Introducción.....	1
Objetivo del Documento.....	1
Contenido Relevante.....	2
Descripción del Proyecto.	3
Requerimientos:.....	3
Diagrama de Clase.....	4
Frameworks.....	5
Frameworks Frontend:.....	5
Frameworks Backend:	7
Framework Por Elegir:.....	10
Patrones de Diseño.	11
Arquitectura e Implementación.....	12
Descripción:.....	12

Introducción.

En el vertiginoso mundo del desarrollo de software, la planificación y el diseño constituyen los cimientos sobre los cuales se erige el éxito de un proyecto. El presente documento de diseño es la manifestación tangible de la meticulosidad y la visión estratégica que guían la materialización de nuestro proyecto de un sistema de gestión de una tienda en línea. Este documento detallado aborda los elementos esenciales del diseño, desde la conceptualización hasta la implementación, proporcionando una hoja de ruta clara para los desarrolladores, arquitectos y demás partes interesadas involucradas en el proceso.

Objetivo del Documento.

El propósito fundamental de este documento es consolidar y comunicar las decisiones fundamentales de diseño que orientarán el desarrollo del proyecto. Desde los diagramas de clases que delinean la estructura conceptual hasta un cuadro comparativo exhaustivo de los frameworks evaluados, cada sección tiene la intención de proporcionar una comprensión profunda de las elecciones de diseño fundamentales.

Contenido Relevante.

El documento abarcará una variedad de elementos críticos, entre los que se incluyen:

1. Diagramas de Clases: Estos representan la anatomía conceptual de nuestro sistema, brindando una visión clara de las entidades clave y sus interrelaciones. A través de estos diagramas, se busca establecer una base sólida para la implementación eficiente y la gestión efectiva de la complejidad.
2. Cuadro Comparativo de Frameworks: La elección del marco de trabajo es una decisión estratégica. Para respaldar esta elección, hemos realizado un análisis comparativo de los frameworks más relevantes, evaluando sus fortalezas, debilidades y su idoneidad para los requisitos específicos de nuestro proyecto.
3. Tipos de Patrones de Diseño: Los patrones de diseño representan soluciones probadas para problemas recurrentes. Su implementación adecuada no solo mejora la eficiencia del código, sino que también promueve la mantenibilidad y escalabilidad del sistema. En este documento, se describirán y justificarán los patrones de diseño seleccionados.
4. Otros Elementos de Diseño Relevantes: Además de los aspectos mencionados, se abordarán otros elementos cruciales del diseño, como consideraciones de seguridad, rendimiento y escalabilidad, entre otros.

Descripción del Proyecto.

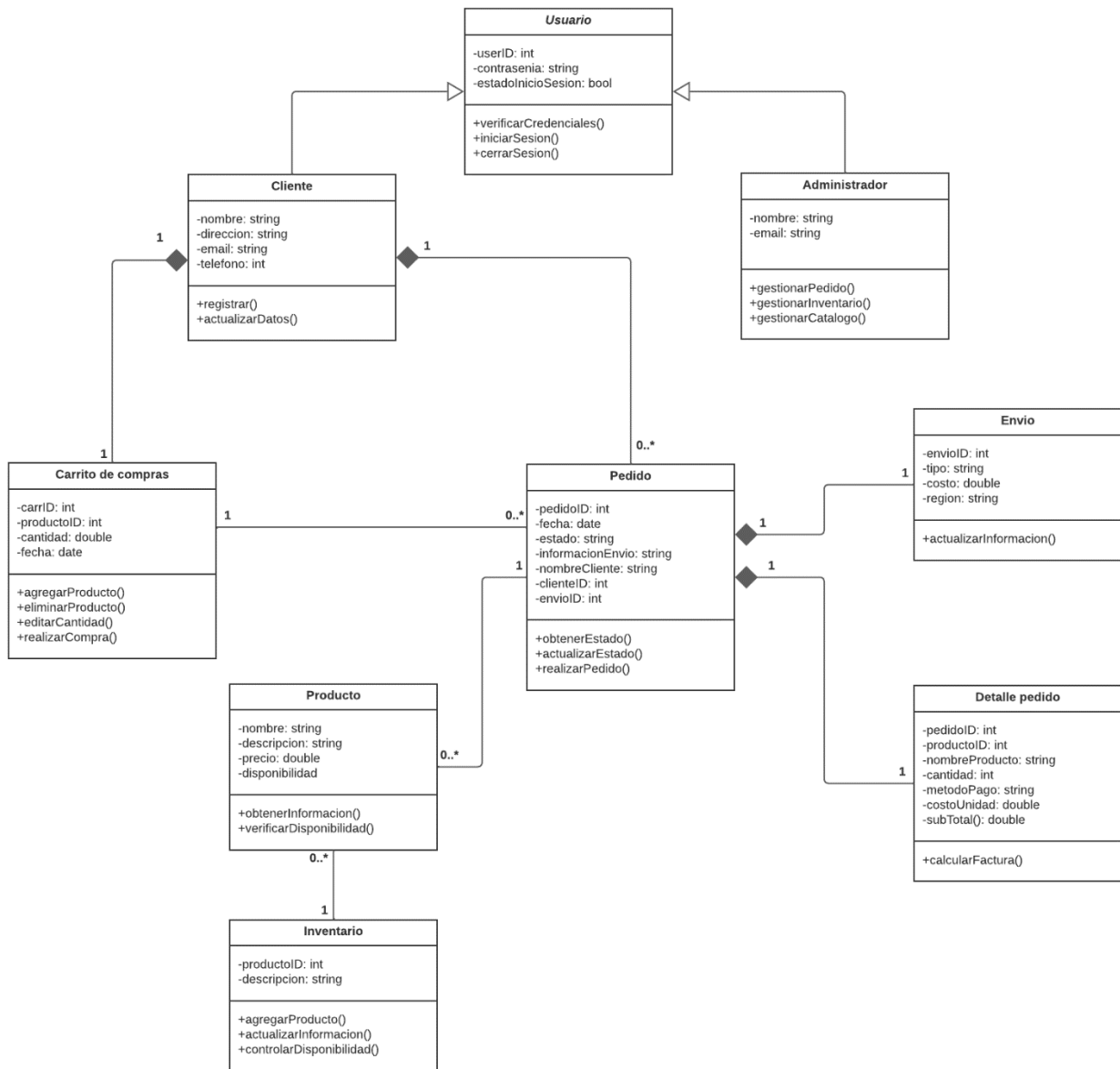
Se requiere el desarrollo de un sistema de gestión para una tienda en línea que permita a los usuarios realizar compras de productos de manera fácil y segura. El sistema debe proporcionar una interfaz amigable para los clientes, así como funcionalidades de gestión para los administradores de la tienda.

Requerimientos:

1. Registro de usuarios: Los clientes podrán registrarse en el sistema proporcionando información personal como nombre, dirección y datos de contacto.
2. Catálogo de productos: El sistema debe mostrar un catálogo de productos disponibles para la compra, con información detallada de cada producto, incluyendo descripción, precio y disponibilidad.
3. Carrito de compras: Los clientes podrán agregar productos al carrito de compras y gestionar su contenido, permitiendo agregar, eliminar o modificar la cantidad de productos seleccionados.
4. Proceso de compra: El sistema debe permitir a los clientes completar el proceso de compra, incluyendo el ingreso de datos de envío, selección de método de pago y confirmación de la orden.
5. Gestión de inventario: Los administradores del sistema podrán gestionar el inventario de productos, incluyendo la adición de nuevos productos, actualización de información y control de disponibilidad.
6. Administración de pedidos: Los administradores podrán ver y gestionar los pedidos realizados por los clientes, incluyendo su estado, envío y facturación.

Diagrama de Clase.

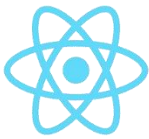

El diagrama de clases proporciona una visión estructurada de las entidades clave y sus relaciones en nuestro sistema de gestión de una tienda en línea. A continuación, se muestra la relación entre cada una de las clases principales:



Frameworks.

En esta sección, se presenta un cuadro comparativo exhaustivo que abarca los diversos frameworks disponibles tanto para el desarrollo en el backend como en el frontend. Este análisis detallado tiene como objetivo proporcionar una visión integral de las características, capacidades y ventajas de cada framework, permitiendo tomar decisiones informadas al seleccionar la plataforma más adecuada para el proyecto.



Frameworks Frontend:

Logo	Nombre	Descripción	Ventajas / Desventajas
	React JS	<p>Es una biblioteca de JavaScript desarrollada por Facebook. Se utiliza para construir interfaces de usuario interactivas y reutilizables mediante la creación de componentes modulares. React se centra en la eficiencia y la capacidad de actualización rápida de la interfaz de usuario.</p> <p>Lenguaje: JavaScript</p>	<p>Ventajas: React se destaca por su eficiente manejo del DOM virtual, que mejora el rendimiento al minimizar las actualizaciones directas del DOM. Además, su enfoque en componentes reutilizables facilita el desarrollo modular y la integración con otras bibliotecas.</p> <p>Desventajas: Como una biblioteca enfocada en la vista, React a menudo requiere el uso de otras bibliotecas o herramientas para la gestión del estado y la manipulación del enrutamiento, lo que puede aumentar la complejidad.</p>
	Angular	<p>Desarrollado por Google, es un framework completo de desarrollo web para la construcción de aplicaciones SPA (Single Page Application). Ofrece una estructura robusta, incorpora inyección de dependencias y se centra en la modularidad y la escalabilidad.</p>	<p>Ventajas: Angular es un framework completo que proporciona una estructura robusta y modular para el desarrollo de aplicaciones SPA. Ofrece características como la inyección de dependencias y un sistema de enrutamiento incorporado, lo que facilita el</p>

		Lenguaje: TypeScript (JavaScript también es compatible)	<p>desarrollo y la escalabilidad de aplicaciones más grandes.</p> <p>Desventajas: La curva de aprendizaje inicial de Angular puede ser empinada para algunos desarrolladores. Además, su complejidad y tamaño pueden hacer que no sea la mejor opción para proyectos más pequeños o simples.</p>
	Vue.js	<p>Es un framework progresivo para construir interfaces de usuario. Es fácil de integrar en proyectos existentes y adoptar de manera incremental. Vue.js se destaca por su flexibilidad y simplicidad, permitiendo a los desarrolladores construir aplicaciones interactivas de manera rápida.</p> <p>Lenguaje: JavaScript</p>	<p>Ventajas: Vue.js se destaca por su simplicidad y flexibilidad. Es fácil de aprender y adoptar incrementalmente en proyectos existentes. Además, su tamaño compacto y rendimiento eficiente lo hacen una opción atractiva para proyectos de diversos tamaños.</p> <p>Desventajas: Aunque ha ganado popularidad rápidamente, Vue.js puede carecer de algunas características avanzadas que ofrecen React y Angular. Además, su comunidad, aunque creciente, puede ser más pequeña en comparación con las otras dos.</p>
	Django	<p>Es un framework web de alto nivel para Python. Se centra en la simplicidad y la productividad, proporcionando un conjunto completo de herramientas para el desarrollo rápido de aplicaciones web. Django incluye un ORM, un sistema de administración automática</p>	<p>Ventajas: Django destaca por su enfoque "baterías incluidas", proporcionando muchas características integradas como un sistema de administración automática, un ORM sólido y herramientas de seguridad. Es conocido por su simplicidad y su capacidad</p>

		<p>y sigue el patrón de diseño MVC.</p> <p>Lenguaje: Python</p>	<p>para facilitar el desarrollo rápido.</p> <p>Desventajas: La flexibilidad de Django puede ser menor en comparación con otros frameworks, ya que sigue un enfoque más opinativo. Además, su curva de aprendizaje puede ser empinada para aquellos nuevos en Python o en el desarrollo web en general.</p>
--	--	---	--

Frameworks Backend:

Logo	Nombre	Descripción	Ventajas / Desventajas
	Express.js	<p>Es un framework web para Node.js que simplifica el desarrollo de APIs. Es minimalista y flexible, permitiendo a los desarrolladores construir de manera rápida y eficiente servicios escalables y de alto rendimiento.</p> <p>Lenguaje: JavaScript (Node.js)</p>	<p>Express.js destaca por su capacidad para construir aplicaciones web y APIs de manera rápida y eficiente en Node.js. Su enfoque minimalista y su amplia adopción en la comunidad hacen que sea ideal para proyectos que requieren escalabilidad y alto rendimiento. Sin embargo, su minimalismo puede requerir la incorporación de más bibliotecas para funcionalidades avanzadas, y algunos desarrolladores pueden preferir frameworks más estructurados.</p>
	Flask	<p>Es un framework ligero para el desarrollo de servicios de API en Python. Es conocido por su simplicidad y flexibilidad, proporcionando herramientas esenciales para la creación de</p>	<p>Flask es apreciado por su simplicidad y flexibilidad en el desarrollo web en Python. Es ideal para proyectos pequeños y medianos, proporcionando</p>

		<p>aplicaciones web y APIs de manera rápida y sencilla.</p> <p>Lenguaje: Python</p>	<p>herramientas esenciales sin imponer una estructura rígida. Sin embargo, para aplicaciones más grandes, algunos desarrolladores pueden optar por frameworks más robustos que ofrezcan más características integradas y estructuras más definidas.</p>
	ASP.NET Core	<p>ASP.NET Core es un framework de desarrollo web de código abierto desarrollado por Microsoft para construir aplicaciones modernas y escalables. Es multiplataforma, modular y sigue el paradigma de diseño MVC (Modelo-Vista-Controlador), facilitando la creación de servicios web y aplicaciones web.</p> <p>Lenguaje: C#</p>	<p>ASP.NET Core es un framework versátil que destaca por su capacidad multiplataforma y modularidad. Es particularmente fuerte en el desarrollo de aplicaciones empresariales y servicios web, aprovechando la potencia del lenguaje C#. Sin embargo, algunos desarrolladores pueden percibirlo como más pesado y complejo en comparación con frameworks más ligeros.</p>
	Gin	<p>Gin es un framework web minimalista para el lenguaje de programación Go. Está diseñado para ser rápido y eficiente, ofreciendo un enrutamiento flexible y soporte para middlewares. Gin es adecuado para el desarrollo de servicios web y APIs en Go.</p> <p>Lenguaje: Go (Golang)</p>	<p>Es ideal para construir servicios web y APIs en Go, proporcionando un enrutamiento rápido y un manejo eficiente de middlewares. Sin embargo, su minimalismo puede significar menos características integradas en comparación con frameworks más completos, y algunos desarrolladores pueden preferir opciones con una comunidad más grande. La elección entre estos</p>

			frameworks depende de los requisitos específicos del proyecto y las preferencias del equipo de desarrollo.
	Spring Boot	<p>Spring Boot es una extensión del framework Spring que simplifica el desarrollo de aplicaciones web y servicios en Java. Proporciona un conjunto de convenciones y configuraciones por defecto para acelerar el proceso de desarrollo, permitiendo a los desarrolladores centrarse en la lógica de negocio.</p> <p>Lenguaje: Java</p>	<p>Spring Boot es una elección popular para el desarrollo en Java, especialmente en el ámbito empresarial. Ofrece una amplia gama de características, desde la gestión de dependencias hasta la creación rápida de aplicaciones. La modularidad y la escalabilidad son puntos fuertes, pero algunos desarrolladores pueden encontrar que su curva de aprendizaje inicial es más pronunciada en comparación con frameworks más simples.</p>
	Laravel	<p>Es un framework de desarrollo web para PHP que sigue el patrón de diseño Modelo-Vista-Controlador (MVC). Ofrece una sintaxis elegante, herramientas para la gestión de bases de datos, y abstracciones que facilitan el desarrollo rápido y eficiente de aplicaciones web modernas.</p> <p>Lenguaje: PHP</p>	<p>Ventajas: Laravel es conocido por su sintaxis elegante, su enfoque en la productividad y su integración con Eloquent, un ORM potente. Facilita el desarrollo de aplicaciones web PHP modernas con características como el enrutamiento simple, la gestión de sesiones y la creación de APIs.</p> <p>Desventajas: Algunos desarrolladores pueden encontrar que el rendimiento de Laravel no es tan rápido como algunos de los frameworks más livianos,</p>

			y su estructura MVC puede resultar excesiva para proyectos más pequeños.
--	--	--	--

Framework Por Elegir:

1. Laravel (Backend):

a. Ventajas:

- i. Moderno y elegante.
- ii. Ecosistema activo y comunidad fuerte.
- iii. Potente ORM (Eloquent), sistema de rutas, controladores, etc.
- iv. Facilidades para la gestión de base de datos, migraciones, seeders, etc.

b. Uso recomendado:

- i. Desarrollo de la lógica del servidor, API, gestión de la base de datos, etc.

2. Laravel Livewire (Frontend):

a. Ventajas:

- i. Integración sencilla con Laravel.
- ii. Desarrollo de interfaces interactivas sin tener que escribir JavaScript.
- iii. Actualizaciones en tiempo real sin recargar la página (ajuste dinámico de las vistas).

b. Uso recomendado:

- i. Desarrollo de componentes de frontend interactivos dentro de tus vistas Blade.

Utilizando esta combinación, Laravel en el backend y Laravel Livewire en el frontend, se puede aprovechar la integración natural entre los dos y desarrollar aplicaciones de manera coherente. Sin embargo, hay que tomar en cuenta que, si la aplicación frontend se vuelve muy compleja y se necesita un manejo avanzado de la interfaz de usuario, se podría necesitar integrar también un framework frontend dedicado como Vue.js o React.

Patrones de Diseño.

En el diseño y desarrollo de la tienda en línea, se han aplicado cuidadosamente varios patrones de diseño para lograr una arquitectura robusta y modular. Entre los patrones utilizados se encuentran:

1. Factory Method y Strategy:

Se implementará el patrón Factory Method para la creación de objetos, permitiendo la creación de instancias de productos específicos de manera uniforme. Además, se integró el patrón Strategy para manejar diferentes estrategias, como el cálculo de precios de productos, proporcionando flexibilidad para cambios futuros en las estrategias de negocio.

2. Singleton y Observer:

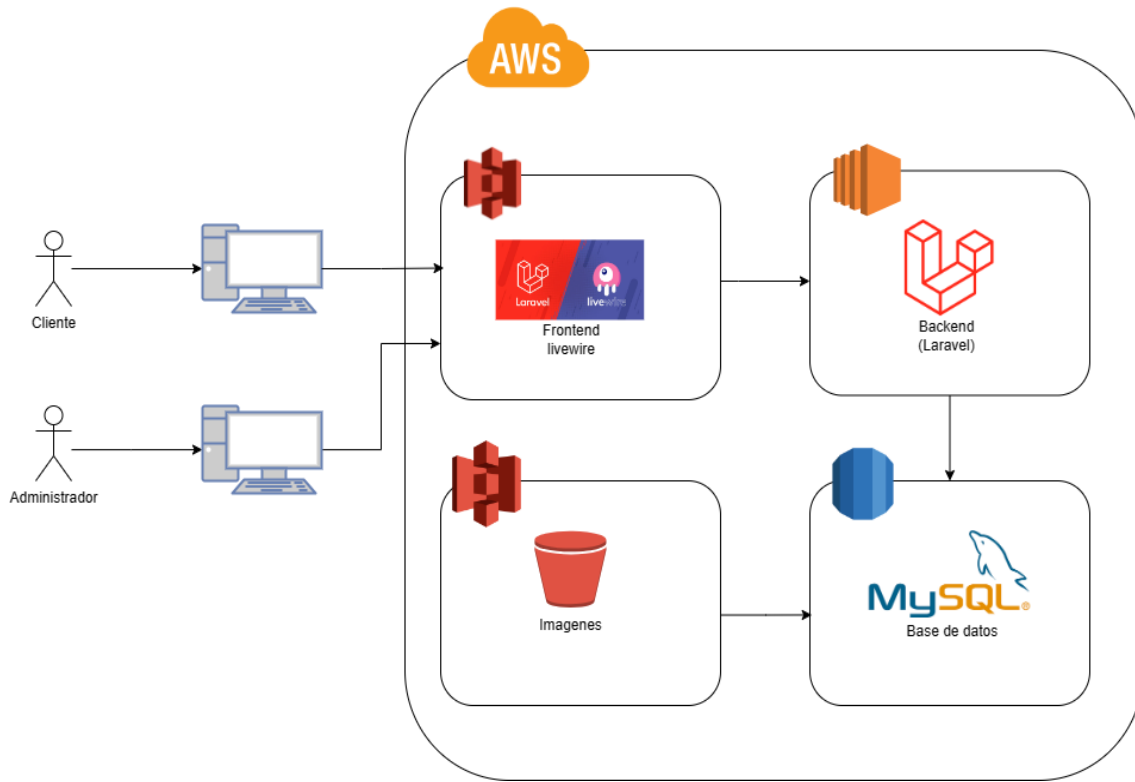
La gestión de la conexión a la base de datos MySQL en RDS se beneficiará del patrón Singleton, garantizando una única instancia compartida para la gestión de la conexión. También se aplicará el patrón Observer para notificar a los clientes sobre cambios en el catálogo de productos y mantenerlos informados sobre la disponibilidad de nuevos productos.

3. Abstract Factory y Composite:

Para la creación de los productos relacionados y la representación jerárquica de productos y variantes, se utilizará el patrón Abstract Factory junto con el patrón Composite. Esto permitirá manejar diferentes tipos de productos y sus variantes de manera eficiente.

Estos patrones de diseño que se integrarán de manera sinérgica para lograr un sistema flexible, escalable y fácil de mantener, proporcionando una base sólida para la funcionalidad y la experiencia del usuario en la tienda en línea.

Arquitectura e Implementación.



Descripción:

La arquitectura anterior demuestra como funcionarían todos los servicios de AWS internamente en nuestra aplicación.

- **S3:**
 - La aplicación está alojada dentro de un bucket de S3 funcionando como un sitio web estático.
 - En otro bucket almacenará todas las imágenes de los productos de la tienda que se suban desde la página web.
- **EC2:**
 - Se tendrán 1 instancias de EC2 donde se montarán 1 servidor para el backend.
 - El lenguaje del servidor sera en PHP.
 - El sistema operativo de la instancia sera Ubuntu en su versión 22.04.
- **RDS:**
 - Se utilizará MySQL en una instancia de RDS, para almacenar todos los datos de la tienda en línea.