

PureGlow

Documentación Técnica del Proyecto

1. Introducción

PureGlow es una aplicación web tipo ecommerce de skincare desarrollada bajo el patrón de arquitectura Modelo-Vista-Controlador (MVC), cuyo objetivo principal es gestionar la interacción entre compradores, vendedores y administradores dentro de una plataforma de comercio electrónico.

El sistema fue diseñado con enfoque en escalabilidad, separación de responsabilidades, integridad de datos y mantenimiento a largo plazo.

2. Arquitectura del Sistema

El proyecto implementa el patrón MVC con el propósito de estructurar el código de forma modular y organizada:

- **Modelo (Model):** Maneja la lógica de negocio y la interacción con la base de datos.
- **Vista (View):** Gestiona la presentación de la información al usuario.
- **Controlador (Controller):** Interpreta las solicitudes del usuario y coordina la comunicación entre modelo y vista.

Esta estructura permite mantener el código desacoplado, facilitando futuras mejoras y mantenimiento.

3. Tecnologías Utilizadas

- **Backend:** PHP
- **Arquitectura:** MVC
- **Base de Datos:** MySQL
- **Administrador de Base de Datos:** phpMyAdmin
- **Frontend:** HTML, CSS, JavaScript
- **Integración de Pagos:** API PayPal (Sandbox)
- **Seguridad:** Consultas preparadas (Prepared Statements)

4. Gestión de Usuarios y Control de Acceso

El sistema implementa un esquema de roles con control de permisos:

- Comprador
- Vendedor
- Administrador
- Super Administrador

Funcionalidades principales:

- Registro de usuarios con validaciones
- Inicio de sesión con control de sesión
- Gestión de perfiles
- Control de acceso por rol
- Restricción de funcionalidades según permisos

El control de acceso se realiza mediante validación de sesión en backend, asegurando que cada usuario solo pueda acceder a las funcionalidades correspondientes a su rol.

5. Gestión de Productos

El sistema permite la administración completa de productos mediante operaciones CRUD (Create, Read, Update, Delete).

Funcionalidades:

- Creación de productos por parte de vendedores
- Aprobación de los nuevos productos por administradores
- Edición de productos
- Asociación de productos a categorías
- Eliminación de categorías
- Control de inventario

Las consultas fueron estructuradas para mantener coherencia relacional y optimización básica de resultados.

6. Sistema de Búsqueda y Filtrado

Se implementaron mecanismos de filtrado dinámico y búsqueda:

- Búsqueda por nombre de producto
- Filtrado por categoría
- Consultas condicionadas mediante parámetros dinámicos
- Resultados renderizados en tiempo real

El diseño de consultas fue orientado a mantener eficiencia y claridad en la obtención de datos.

7. Wishlist

El sistema incluye funcionalidad de lista de deseos:

- Asociación de productos a usuarios
- Almacenamiento relacional en base de datos
- Gestión individual por usuario autenticado

8. Sistema de Compras y Procesamiento de Pagos

PureGlow integra un flujo completo de compra:

Proceso implementado:

1. Agregado de productos al carrito.
2. Confirmación de pedido.
3. Selección de método de pago.
4. Procesamiento mediante:
 - Pago con tarjeta
 - Integración con API PayPal (Sandbox)
5. Validación de respuesta de la API.
6. Calificación de los productos.
7. Registro de la compra en base de datos.

Se implementó validación tanto en frontend como en backend para garantizar consistencia e integridad de la información.

9. Sistema de Mensajería

Se desarrolló un sistema interno de comunicación entre compradores y vendedores:

- Envío de mensajes asociados a productos que se pudieran cotizar
- Almacenamiento en base de datos relacional
- Relación entre usuario emisor y receptor

Este módulo permite interacción directa dentro de la plataforma sin depender de servicios externos.

10. Diseño y Estructura de Base de Datos

La base de datos fue diseñada bajo principios de normalización para evitar redundancia y mantener integridad referencial.

Características principales:

- Uso de llaves primarias en todas las entidades
- Llaves foráneas para relaciones entre tablas
- Relaciones entre:
 - Usuarios
 - Productos
 - Categorías

- Compras
- Wishlist
- Mensajes
- Imágenes
- Videos
- Etc.

Se aplicaron restricciones para garantizar consistencia en inserciones, actualizaciones y eliminaciones.

11. Seguridad y Buenas Prácticas

- Implementación de consultas preparadas para prevenir inyección SQL.
- Validación de datos de entrada.
- Control de sesiones para autenticación.
- Separación clara entre lógica de negocio y presentación.
- Organización modular del código bajo MVC.

12. Enfoque de Desarrollo

El desarrollo del proyecto se realizó con enfoque en:

- Resolución autónoma de problemas técnicos.
- Organización estructural del código.
- Escalabilidad futura.
- Mantenimiento y claridad.
- Traducción de requerimientos funcionales en lógica backend eficiente.