

Índice

Introducción.....	2
Objetivos.....	2
Tecnologías Utilizadas.....	2
Tecnologías escogidas y justificación.....	2
Motor de bases de datos.....	3
Diseño de la Aplicación.....	3
Casos de Uso.....	3
Modelo de Entidad-Relación de la Base de Datos.....	5
Arquitectura de la Aplicación.....	6
Estructura del Proyecto.....	6
Casos de Prueba.....	7
Login:.....	7
Crear Cuenta.....	7
Agregar/Eliminar Producto del Carrito.....	7
Editar el Carrito.....	7
Confirmar Pedido.....	7
Pagar Pedido.....	8
Pagar Pedido Pendiente desde Mis Pedidos.....	8
Crear/Actualizar/Eliminar Producto en Admin.....	8
Editar/Eliminar Usuario en Admin.....	8
Eliminar Pedido en Admin.....	8

Introducción

Este proyecto se enfoca en la creación de una aplicación web diseñada para ser utilizada como una tienda virtual especializada en la venta de alfombras y tapices. La aplicación está compuesta por varias interfaces, las cuales se adaptan tanto en apariencia como en funcionalidad dependiendo del rol que tenga el usuario registrado.

La aplicación está desarrollada como una plataforma web, lo que permite a los usuarios acceder a ella sin necesidad de instalar ningún tipo de software en sus dispositivos. El único requisito para acceder a la aplicación es contar con un navegador web y una conexión a internet.

Además, la aplicación puede ser accedida desde una variedad de dispositivos electrónicos, tales como teléfonos celulares, tabletas o computadoras, lo que brinda flexibilidad y comodidad a los usuarios al permitirles acceder a la tienda virtual desde cualquier lugar y en cualquier momento.

Objetivos

El principal objetivo de este proyecto es proporcionar a los usuarios una tienda virtual especializada en la venta de alfombras y tapices, con el fin de que puedan encontrar productos que se ajusten específicamente a sus necesidades de decoración para cualquier tipo de ambiente, evitando así la pérdida de tiempo en la búsqueda de artículos específicos en tiendas más generales.

Otro propósito es habilitar a los usuarios con una mayor capacidad de interacción, permitiéndoles visualizar productos, añadir o eliminar artículos del carrito, realizar compras al instante o guardar pedidos para su posterior pago, así como revisar el historial de pedidos. Además, se incluye la opción de crear una cuenta, iniciar y cerrar sesión, restablecer contraseñas, y otras funcionalidades. Asimismo, se ha previsto un área para que un administrador pueda acceder a toda la información de la tienda y gestionarla, incluyendo productos, usuarios, pedidos, entre otros aspectos.

Tecnologías Utilizadas

Tecnologías escogidas y justificación

La aplicación se desarrolla mayormente utilizando PHP como lenguaje de servidor. Sin embargo, también incluye secciones realizadas en JavaScript como lenguaje de cliente.

Es importante destacar que todo el software utilizado es de código abierto, lo que resultará en ahorros significativos en concepto de licencias para el dueño de la tienda. La elección de este software no se basa únicamente en su gratuidad, sino también en que es una de las tecnologías más populares en la actualidad para el desarrollo de aplicaciones web, gracias a su confiabilidad y versatilidad.

Además, se emplea CSS para el diseño de la página web, lo que contribuye a la presentación visual atractiva y funcional.

En cuanto a la programación, se utilizó el sistema de MVC (Modelo-Vista-Controlador). Este enfoque organiza el código en tres partes principales, lo que facilita la mantenibilidad y la escalabilidad del sistema. El modelo se encarga de la gestión de datos, la vista se encarga de la presentación y la interfaz con el usuario, y el controlador coordina la interacción entre el modelo y la vista. Esta estructura modular proporciona una base sólida para el desarrollo y la gestión de la aplicación.

Motor de bases de datos

El sistema de gestión de bases de datos elegido es MySQL. Las razones que respaldan esta elección son las siguientes:

- Es de adquisición gratuita, lo que contribuye a la reducción de costos para el cliente.
- Es compatible con múltiples plataformas, incluyendo Windows, Linux y Mac, los cuales son los sistemas operativos más utilizados, lo que permite su disponibilidad en cualquier entorno.
- Es ampliamente utilizado en la comunidad de desarrolladores, lo que facilita la obtención de asistencia y soporte.
- La tarea de mantenimiento de una base de datos MySQL es sencilla debido a su enfoque en funciones más simples en comparación con otros sistemas de gestión. Aunque esto podría considerarse una desventaja, permite que el propio desarrollador se encargue del mantenimiento de la aplicación, sin necesidad de recurrir a un administrador de bases de datos.
- Es escalable, lo que garantiza una adaptación eficiente a futuros crecimientos y demandas de la tienda virtual.

Además, MySQL destaca por su capacidad para manejar eficientemente grandes cantidades de datos y alto tráfico en aplicaciones web. Su rendimiento sólido y su velocidad de procesamiento lo hacen ideal para entornos comerciales en línea que requieren un sistema confiable y rápido para gestionar su información. Aunque este es un proyecto educativo en el cual no se manejarán grandes cantidades de datos, es muy útil utilizar MySQL debido a sus beneficios y su relevancia en el ámbito del desarrollo web.

Diseño de la Aplicación

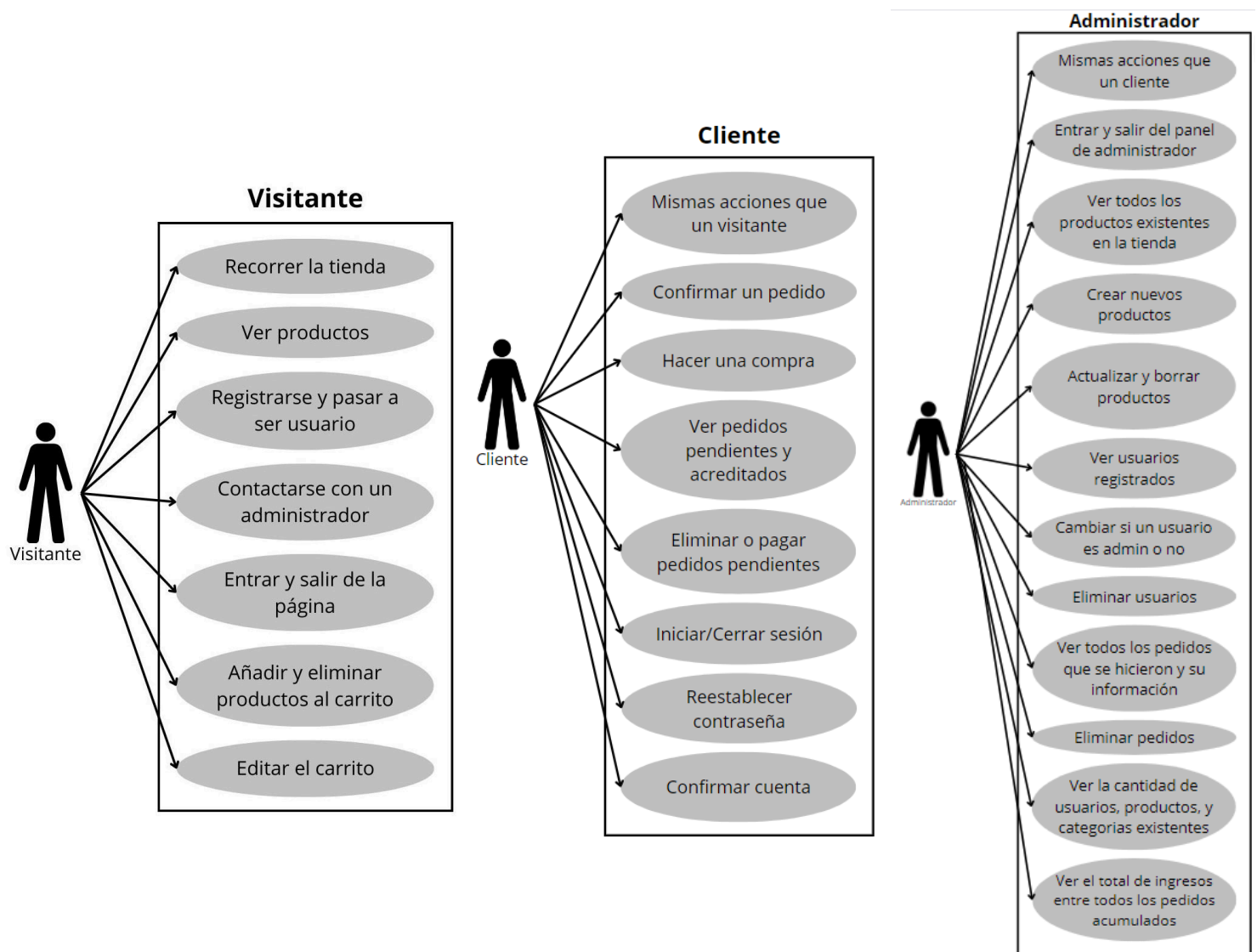
Casos de Uso

En la tienda virtual, existen 3 tipos de usuarios:

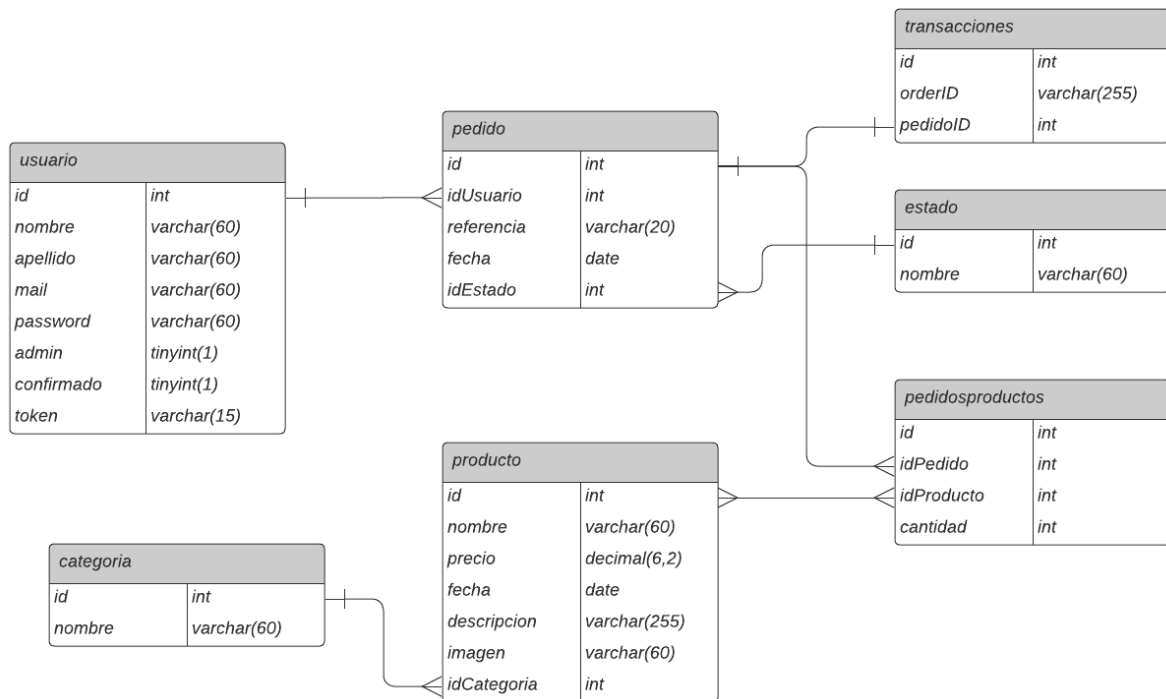
- **Visitantes:** Son usuarios no registrados en la tienda y, por lo tanto, no han creado una cuenta. Aun así, tienen la posibilidad de explorar la tienda, visualizar todos los productos y agregar artículos al carrito. Sin embargo, no cuentan con la capacidad de realizar un pedido ni efectuar el pago, para lo cual necesitarán registrarse.

También tienen la opción de contactar a los administradores a través de un formulario de contacto.

- **Cientes:** Estos usuarios se han registrado y disponen de una cuenta propia. Tienen acceso a las mismas funciones que un visitante, con la diferencia de que al estar registrados, tienen la capacidad de realizar pedidos y efectuar los pagos. Además, cuentan con una interfaz que les permite visualizar todos los pedidos realizados, incluyendo aquellos que ya han sido procesados, así como los pedidos pendientes de pago, que pueden abonar en cualquier momento o eliminarlos.
- **Administradores:** Son los encargados de la gestión de la tienda. Tienen la autoridad para crear, actualizar y eliminar productos, así como administrar a los usuarios existentes. Poseen la capacidad de eliminar usuarios o asignarles el rol de administrador, con la facultad de revertir esta acción si así lo desean. Además, tienen acceso a la información detallada de todos los pedidos realizados, con la posibilidad de eliminarlos. En la página principal del panel de administrador, pueden visualizar información relevante como las ganancias totales, la cantidad de productos, alfombras, tapices, pedidos, pedidos pendientes y acreditados, entre otros datos. Es importante destacar que también tienen libertad para explorar la tienda de la misma manera que un cliente.



Modelo de Entidad-Relación de la Base de Datos



Las relaciones entre las tablas son las siguientes:

- **Usuario-Pedido:** Esta relación es de uno a muchos, donde un usuario puede tener varios pedidos, pero cada pedido está asociado a un único usuario.
- **Pedido-Transacción:** Esta relación es de uno a uno, lo que significa que cada pedido está vinculado a una única transacción y cada transacción está asociada a un solo pedido.
- **Pedido-Estado:** Esta relación es de uno a muchos, donde un estado está asociado a varios pedidos, pero cada pedido tiene un único estado.
- **Pedido-Producto:** Esta relación es de muchos a muchos, lo que se logra a través de la tabla de unión "productosproductos". Cada pedido puede contener varios productos y cada producto puede estar presente en varios pedidos.
- **Categoría-Producto:** Esta relación es de uno a muchos, donde una categoría puede estar asociada a varios productos, pero cada producto pertenece a una única categoría.

Teniendo en cuenta estas relaciones, es evidente la utilidad del modelo de entidad-relación para representar la estructura de la base de datos. Este modelo proporciona una manera clara y visual de comprender cómo las entidades (como Usuario, Pedido, Transacción, Estado, Producto y Categoría) están relacionadas entre sí, lo que facilita el diseño, la comprensión y el mantenimiento de la base de datos.

Además, el modelo de entidad-relación permite identificar de manera eficiente las restricciones y dependencias entre las entidades, lo que resulta fundamental para garantizar la integridad de los datos y la coherencia de la información almacenada. Asimismo, facilita la identificación de las claves primarias y foráneas, lo que es crucial para establecer relaciones sólidas entre las entidades.

Arquitectura de la Aplicación

Estructura del Proyecto

- **classes:** Contiene una clase que se utiliza para usar mailtrap en el proyecto.
- **controllers:** Contiene los controladores que dan funcionalidad a la aplicación.
- **includes:** Contiene la configuración de la base de datos, plantillas utilizadas en varias partes de la página, así como archivos adicionales para el funcionamiento de la aplicación.
 - **config:** Contiene la configuración de la base de datos.
 - **templates:** Contiene unas plantillas utilizadas en muchas partes de la página. Ejemplo: header
- **models:** Contiene todos los modelos que representan la estructura de datos de la aplicación.
- **public:** Contiene los archivos accesibles públicamente en el servidor web.
 - **build:** Contiene los archivos generados después de compilar y procesar los recursos del proyecto, listos para ser desplegados en el servidor web.
 - **css:** Contiene todas las hojas de estilos.
 - **base:** Contiene hojas de estilos que se utilizan como globales en la página.
 - **layout:** Contiene hojas de estilos que le dan distintos aspectos a cada interfaz.
 - **img:**
 - **alfombras:** Contiene imágenes de alfombras.
 - **generales:** Contiene imágenes importantes que se utilizan en distintas interfaces.
 - **iconos:** Contiene los iconos que se utilizan en la página.
 - **tapices:** Contiene imágenes de tapices.
 - **js:** Contiene un archivo JavaScript que implementa varias funcionalidades en la página.
 - **imagenes:** Contiene las imágenes de los productos que se han creado,
- **vendor:** Contiene las librerías externas, como PHPMailer, utilizadas en la aplicación
- **views:** Contiene las vistas de admin, auth, home, templates, y el layout principal.
 - **admin:** Contiene las vistas relacionadas a las interfaces del panel de administrador.
 - **auth:** Contiene las vistas relacionadas a las interfaces de inicio de sesión, crear cuenta, etc.
 - **home:** Contiene las vistas relacionadas a las interfaces principales de la tienda.
 - **templates:** Contiene la vista de una plantilla de alertas.

Casos de Prueba

Login:

- **Resultado:** La función de inicio de sesión funciona correctamente. Los usuarios pueden iniciar sesión con sus credenciales y acceder a sus cuentas sin problemas.
- **Observaciones:** No se encontraron errores o fallos durante la prueba.

Crear Cuenta

- **Resultado:** El proceso de creación de cuentas es exitoso. Los usuarios pueden registrarse correctamente ingresando su información y reciben confirmación de creación de cuenta.
- **Observaciones:** No se observaron problemas durante la prueba. El flujo de registro es intuitivo y fácil de seguir.

Agregar/Eliminar Producto del Carrito

- **Resultado:** Se pudo agregar y eliminar productos del carrito sin inconvenientes. Los productos se reflejan correctamente en el carrito y se actualiza el total de la compra correctamente.
- **Observaciones:** El sistema responde de manera rápida y precisa a las acciones del usuario en el carrito de compras.

Editar el Carrito

- **Resultado:** La funcionalidad para editar el carrito de compras funciona correctamente. Los usuarios pueden modificar la cantidad de productos en su carrito y se actualiza el total de la compra de manera adecuada.
- **Observaciones:** La interfaz de edición del carrito es intuitiva y fácil de usar.

Confirmar Pedido

- **Resultado:** Los usuarios pueden confirmar sus pedidos sin problemas. Se muestra un resumen detallado de la orden antes de la confirmación, y los usuarios pueden proceder con la compra exitosamente.
- **Observaciones:** El proceso de confirmación de pedido es claro y transparente para los usuarios.

Pagar Pedido

- **Resultado:** El proceso de pago de los pedidos se completó con éxito. Los usuarios pueden seleccionar el método de pago deseado y finalizar la transacción sin problemas.
- **Observaciones:** No se observaron errores durante el proceso de pago. La transacción se procesa de manera segura y eficiente.

Pagar Pedido Pendiente desde Mis Pedidos

- **Resultado:** Los usuarios pueden pagar pedidos pendientes desde la sección "Mis Pedidos" sin inconvenientes. El sistema registra el pago correctamente y actualiza el estado del pedido.
- **Observaciones:** El flujo para pagar pedidos pendientes es claro y fácil de seguir para los usuarios.

Crear/Actualizar/Eliminar Producto en Admin

- **Resultado:** Las funciones de administración de productos están operativas. Los administradores pueden crear, actualizar y eliminar productos de manera efectiva.
- **Observaciones:** La interfaz de administración proporciona todas las herramientas necesarias para gestionar los productos de la tienda de manera eficiente.

Editar/Eliminar Usuario en Admin

- **Resultado:** Los administradores pueden editar y eliminar usuarios del sistema sin problemas. Las acciones se realizan correctamente y se reflejan en la base de datos.
- **Observaciones:** La función de administración de usuarios es fácil de usar y proporciona control completo sobre las cuentas de usuario.

Eliminar Pedido en Admin

- **Resultado:** Los administradores pueden eliminar pedidos de manera efectiva desde la interfaz de administración. La acción se realiza correctamente y los registros se actualizan adecuadamente.
- **Observaciones:** El proceso de eliminación de pedidos es sencillo y directo para los administradores.