



IES Francisco de Quevedo
Instituto bilingüe de la Comunidad de Madrid



Desarrollo de Aplicaciones Web

Curso académico: 2023– 2024

Proyecto Fin de Ciclo

Creación de una tienda online especializada en la venta de instrumentos musicales, equipamiento de estudio y audio profesional.



Autor: Jaime Pastor López-Puigcerver

Tutor: Carlos González Sánchez

Junio 2024



Índice

1. Contenido del documento.....	3
1.1. Introducción	3
1.2. Justificación del proyecto y objetivos.....	3
1.3. Planificación.....	4
1.4. Parte experimental.	5
1.4.1. Análisis. ¿Qué hará la aplicación?.....	5
1.4.2 Diseño ¿Cómo se hará la aplicación?	8
1.4.3 Implementación y pruebas	11
1.4.4 Implantación y documentación	21
1.4.5 Resultados y discusión	22
1.5. Conclusiones	26
1.6. Bibliografía y referencias	25



1. Contenido del documento:

1.1. Introducción

En qué consiste el proyecto: Este PCF explora en profundidad el proceso de creación y desarrollo de una tienda en línea que satisface las necesidades de músicos y profesionales del audio, como de las personas interesadas por el mundo de la música y el audio, buscando el sonido más profesional, pero sin dedicarse a ello.

Módulos implicados en el desarrollo del proyecto:

- Desarrollo web en entorno cliente
- Desarrollo web en entorno servidor
- Diseño de interfaces web
- Bases de Datos

1.2. Justificación del proyecto y objetivos.

Por qué has elegido este proyecto: He elegido este proyecto de crear una tienda en línea especializada en la venta de instrumentos musicales, equipamiento de estudio y audio profesional por una serie de razones profundamente arraigadas en mi pasión y experiencia en el mundo de la música y el sonido, como comprador ya pasé por el proceso de buscar la mejor tienda online de este sector y la mejor y la que acabé utilizando fue una alemana, la idea es crear una tienda igual de profesional pero en España.

La industria de la música y el audio profesional ha demostrado un crecimiento constante, impulsado en parte por la digitalización de la música y el aumento de la producción musical independiente. La pandemia de COVID-19 también ha impulsado la demanda de equipos y suministros para estudios caseros y grabaciones remotas.

Existen numerosas tiendas en línea que ofrecen una amplia gama de productos, desde guitarras y teclados hasta micrófonos y equipos de grabación.

Algunas tiendas se especializan en ciertos nichos, como instrumentos vintage, equipos de audio de gama alta o productos para músicos específicos (por ejemplo, bateristas o DJ), la idea de esta página es englobar todo en una sola tienda con productos de todo tipo.

El objetivo principal de este proyecto es establecer una tienda en línea de renombre en el mercado de la música y el audio profesional que ofrezca una amplia gama de productos de alta calidad y servicios excepcionales.



El perfil del público al que se dirige la aplicación la resumiría en tres grupos:

Músicos y Aficionados, desde principiantes hasta músicos con experiencia. Estos clientes buscan instrumentos musicales de calidad, como guitarras, teclados, baterías, etc., así como accesorios y equipos para mejorar su experiencia musical, buscan tener las mejores marcas en base a la calidad y precio, o la mejor calidad posible.

Productores Musicales y Compositores, buscan equipamiento de estudio de alta gama, software de producción musical y herramientas para crear y grabar música de calidad profesional.

Profesionales del Sonido y Técnicos de Audio, este segmento se compone de ingenieros de sonido, técnicos de audio en vivo y otros profesionales de la industria del sonido. Necesitan equipos profesionales de audio, como mesas de mezclas, micrófonos de estudio, altavoces y accesorios.

1.3. Planificación

Ideas que llevar a cabo en la aplicación:

Configuración de la Tienda.

Como propietario de la tienda, configurar una cuenta de administrador.

Navegación y Exploración de Productos.

Como cliente, quiero poder navegar por las categorías de productos (por ejemplo, guitarras, teclados, micrófonos) para encontrar lo que necesito, poder buscar productos por marca y modelo para comparar especificaciones, ver reseñas de productos y calificaciones de otros usuarios para tomar decisiones de compra informadas.

Gestión de Carrito de Compras y Compra de Productos.

Como comprador, quiero poder agregar productos a mi carrito de compras y luego finalizar la compra de manera segura, poder aplicar códigos de descuento o promociones a mi compra y recibir un correo electrónico de confirmación de compra después de completar mi pedido.

Perfil de Usuario.

Como músico, quiero crear un perfil de usuario en la tienda para recibir recomendaciones personalizadas y acceder a mi historial de compras.

Educación y Recursos.



Como principiante en la música, quiero acceder a tutoriales y guías para aprender a tocar un instrumento.

Gestión de Inventario.

Como propietario de la tienda, quiero gestionar el inventario y recibir notificaciones cuando los productos estén agotados o requieran reposición.

Servicio al Cliente y Soporte.

Como cliente, quiero tener acceso a un servicio de atención al cliente en línea para resolver consultas y problemas.

Proceso de Pago y Entrega

Como comprador, quiero poder seleccionar opciones de envío y realizar un seguimiento del estado de mi pedido.

1.4. Parte experimental

1.4.1. Análisis ¿Qué hará la aplicación?

Diagrama de modelo de datos:

Entidades Principales:

Productos

Atributos: ID de producto (PK), nombre, descripción, precio, cantidad en stock, marca, modelo, categoría, imágenes, calificaciones, reseñas.

Productos

id (PK)

cantidad_en_stock

descripcion

imagen

nombre

precio

idcategoria (FK)

idmarca (FK)



Cientes

Atributos: ID de cliente (PK), nombre, dirección de correo electrónico, dirección de envío, historial de compras, carrito de compras.

Cientes

id (PK)
correo_electronico
direccion_envio
nombre
password

Pedidos

Atributos: ID de pedido (PK), fecha de pedido, ID del cliente, productos en el pedido, dirección de envío, estado del pedido, total.

Pedidos

id (PK)
estado_pedido
fecha_pedido
total
idcliente (FK)

Categorías de Productos

Atributos: ID de categoría (PK), nombre, descripción.

Categorías

id (PK)
descripcion
nombre

Marcas de Productos

Atributos: ID de marca (PK), nombre, descripción.

Marcas

id (PK)
descripcion
nombre



Reseñas y Calificaciones

Atributos: ID de reseña (PK), ID del producto (FK), ID del cliente (FK), calificación, comentario, fecha de la reseña.

Reseñas

id (PK)

calificacion

comentario

fecha_resena

idcliente (FK)

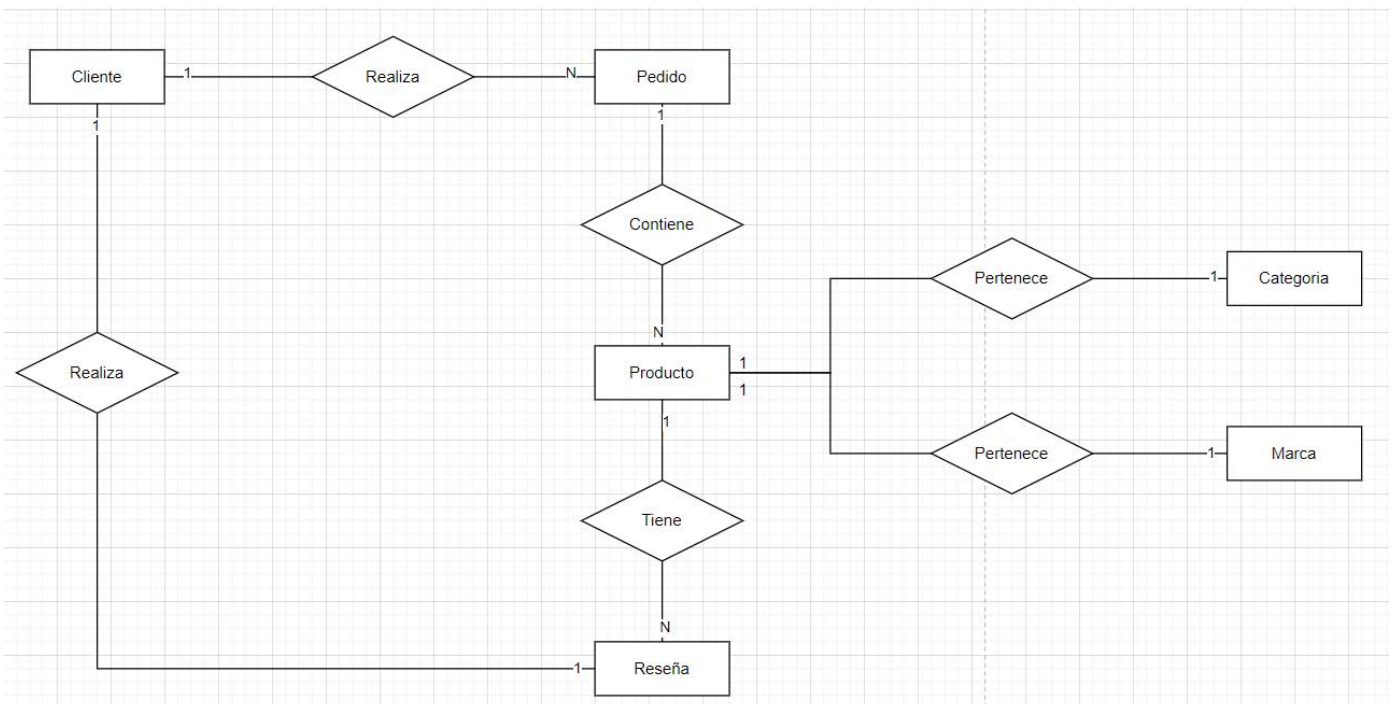
idproducto (FK)



Relaciones:

Un producto está relacionado con una categoría y una marca.
Un producto puede tener una sola categoría y una sola marca, mientras que una categoría puede tener varios productos y una marca puede estar asociada con múltiples productos.
Un cliente puede realizar varios pedidos.
Un pedido está asociado a un solo cliente.
Un pedido también puede contener múltiples productos (relación de muchos a muchos entre pedido y producto).
Las reseñas están relacionadas con los productos y los clientes que las han realizado.

Producto (1) - (1) Categoría
Producto (1) - (1) Marca
Cliente (1) - (*) Pedido
Pedido (1) - (*) Producto
Reseña (1) - (1) Producto
Reseña (1) - (1) Cliente





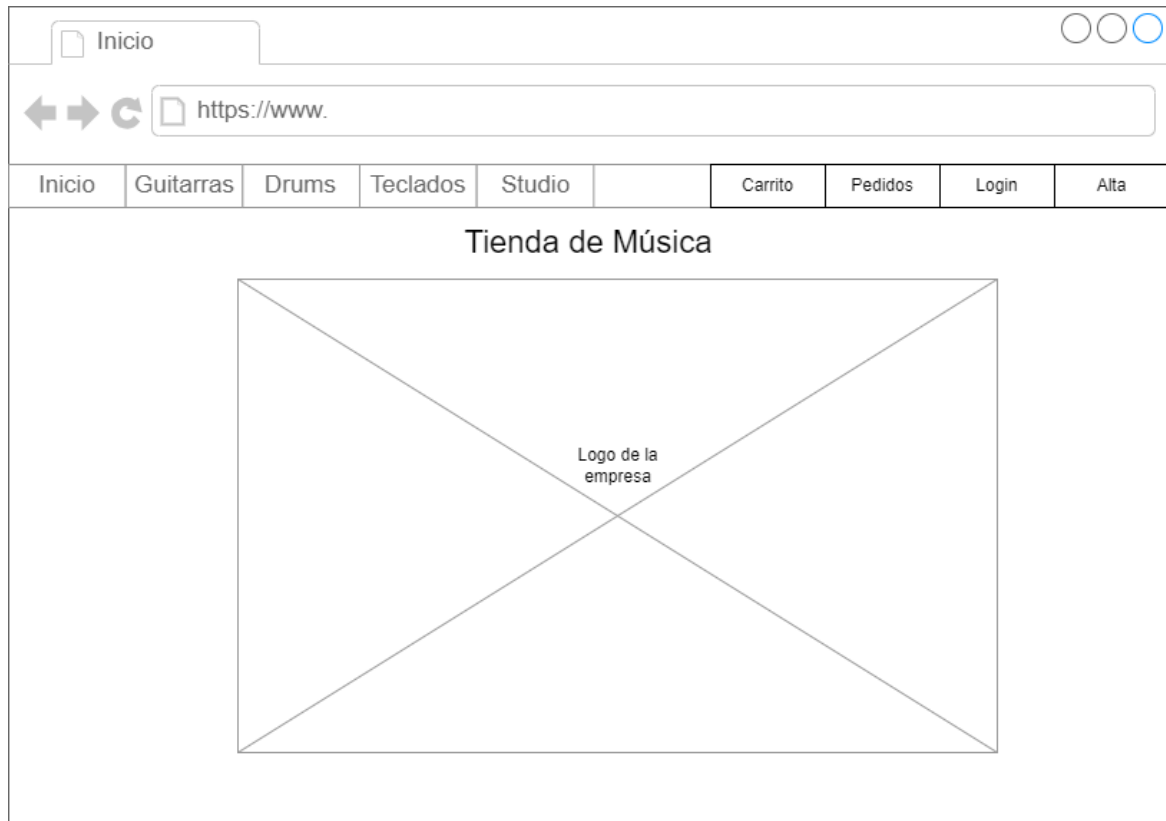
Prototipo de la web

Inicio sin login:

Página principal con el logo de la empresa.

Incluye un menú en forma de barra con enlaces a todas las páginas de la tienda.

Como no hay ningún usuario iniciado, incluye la opción de iniciar sesión o darse de alta.

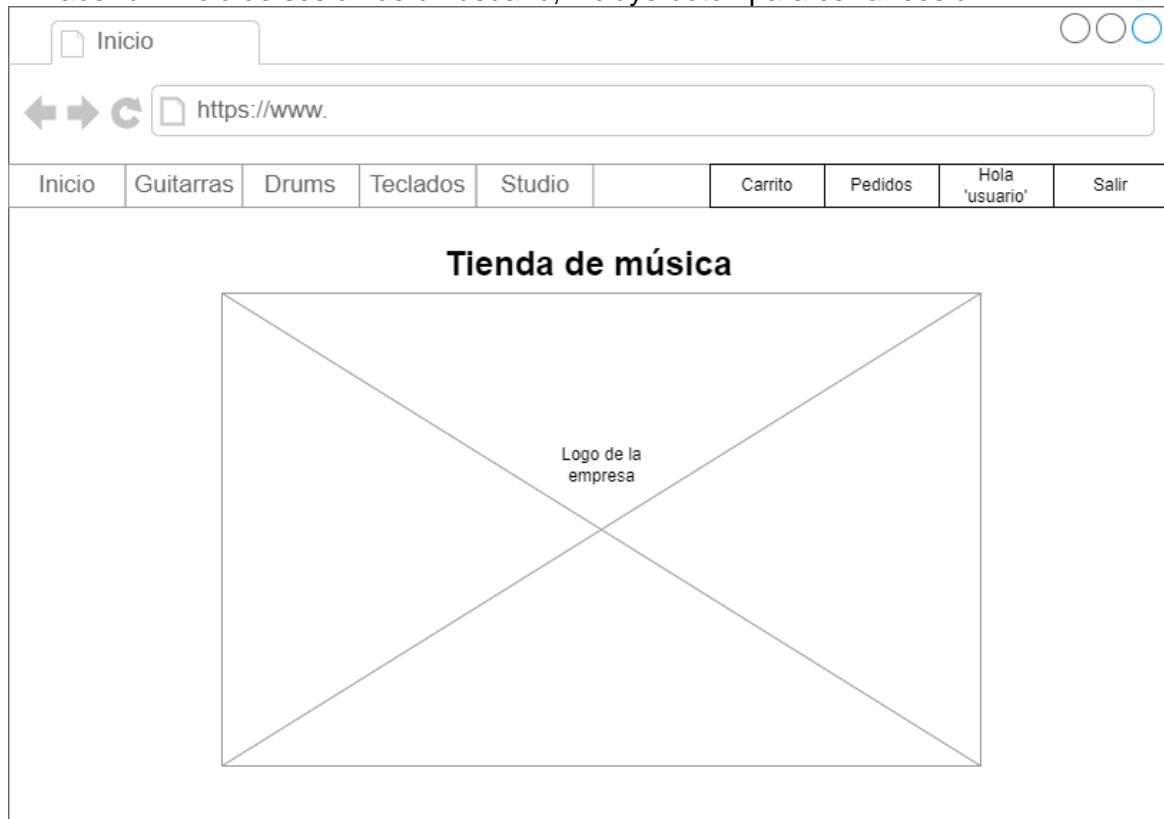




Inicio con login:

Inicio de la página con una sesión iniciada.

Al haber un inicio de sesión de un usuario, incluye botón para cerrar sesión.





Login:
Pagina para iniciar sesión con nombre y contraseña

Inicio

https://www.

Inicio	Guitarras	Drums	Teclados	Studio		Carrito	Pedidos	Login	Alta
--------	-----------	-------	----------	--------	--	---------	---------	-------	------

Login

Nombre

Contraseña

Enviar

Alta:
Formulario para dar de alta a un nuevo cliente
Campos para Nombre, contraseña, correo electrónico y dirección

Inicio

https://www.

Inicio	Guitarras	Drums	Teclados	Studio		Carrito	Pedidos	Login	Alta
--------	-----------	-------	----------	--------	--	---------	---------	-------	------

Alta de Clientes

Nombre

Contraseña

Correo electronico

Dirección de envío

Enviar





Catálogo de productos:

Lista de productos con su imagen y un enlace a la página en detalle del producto

Inicio

https://www.

Inicio

Guitarras

Drums

Teclados

Studio

Carrito

Pedidos

Login

Alta

Lista de productos



Imagen

[Producto 1](#)



Imagen

[Producto 2](#)



Imagen

[Producto 3](#)



Producto:

Página del producto

Incluye una imagen del producto en cuestión, su descripción, el precio, la cantidad en stock, la marca y un botón para añadirla al carro.

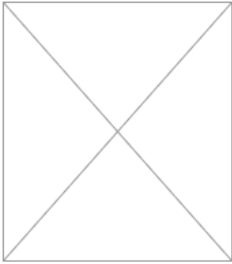
En la parte de abajo las reseñas de otros usuarios sobre el producto, y un botón para añadir una reseña. Si no hay ninguna sesión iniciada o es el usuario admin, el botón para añadir reseñas no aparecerá.

Inicio

https://www.

Inicio	Guitarras	Drums	Teclados	Studio		Carrito	Pedidos	Hola 'usuario'	Salir
--------	-----------	-------	----------	--------	--	---------	---------	----------------	-------

Producto



Nombre

Descripción: "Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua."

Precio: 99.99

Cantidad en stock: 5

Marca: marca

Añadir al carrito

Reseñas

Añadir reseña

Nombre

Nota y descripción



Carrito:
Pagina donde salen todos los pedidos del usuario.
Si accedes como usuario admin aparecerán los pedidos

Inicio									



1.4.2. **Diseño. ¿Cómo se hará la aplicación?**

- Estructura de la base de datos:

Para la base de datos he elegido MySQL como sistema de gestión de bases de datos. La estructura de la base de datos se organizará en varias tablas clave:

Tabla de Productos: Esta tabla almacena información sobre los productos que se ofrecerán en la tienda, como nombre, descripción, precio, cantidad en stock, marca, modelo, categoría, imágenes, calificaciones y reseñas.

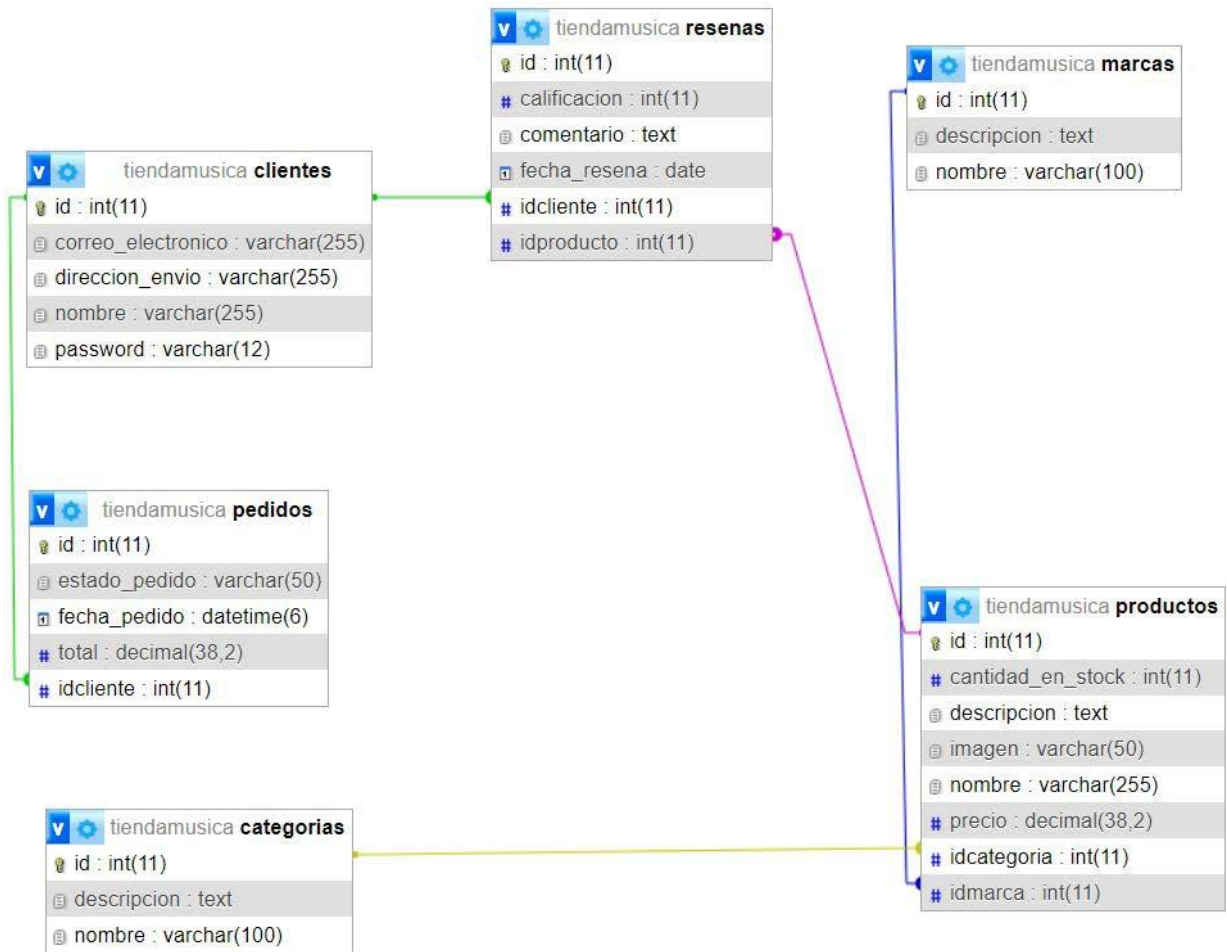
Tabla de Clientes: Aquí almacenaremos datos sobre nuestros clientes, incluyendo su nombre, dirección de correo electrónico, dirección de envío y carrito de compras.

En el caso de querer incluir un historial de compras podríamos hacer otra tabla llamada "Compras", donde almacenar detalles de la fecha de compra, el monto total y una referencia del cliente que realizó la compra.

Tabla de Pedidos: Esta tabla rastrea todos los pedidos realizados por clientes, almacena datos como fecha de pedido, productos en el pedido, dirección de envío, estado del pedido, total y una relación con la tabla cliente

Tablas Categorías y Marcas: Estas tablas permiten clasificar y etiquetar nuestros productos. Cada categoría y marca tiene un ID único, nombre y descripción.

Tabla de Reseñas: Para capturar las opiniones de los clientes sobre los productos utilizaremos esta tabla. Almacena información, el comentario y la fecha de la reseña, así como las relaciones con las tablas Producto y Cliente.



Detalle de las tablas utilizadas, diagrama generado con PhpMyAdmin



Creación de las tablas SQL:

Tabla Productos:

```
CREATE TABLE `productos` (  
  `id` int(11) NOT NULL,  
  `cantidad_en_stock` int(11) NOT NULL,  
  `descripcion` text,  
  `imagen` varchar(50) NOT NULL,  
  `nombre` varchar(255) NOT NULL,  
  `precio` decimal(38,2) NOT NULL,  
  `idcategoria` int(11) DEFAULT NULL,  
  `idmarca` int(11) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Tabla Clientes:

```
CREATE TABLE `clientes` (  
  `id` int(11) NOT NULL,  
  `correo_electronico` varchar(255) NOT NULL,  
  `direccion_envio` varchar(255) DEFAULT NULL,  
  `nombre` varchar(255) NOT NULL,  
  `password` varchar(12) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Tabla Pedidos:

```
CREATE TABLE `pedidos` (  
  `id` int(11) NOT NULL,  
  `estado_pedido` varchar(50) NOT NULL,  
  `fecha_pedido` datetime(6) NOT NULL,  
  `total` decimal(38,2) NOT NULL,  
  `idcliente` int(11) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Tabla Categorías:

```
CREATE TABLE `categorias` (  
  `id` int(11) NOT NULL,  
  `descripcion` text,  
  `nombre` varchar(100) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Tabla Marcas:

```
CREATE TABLE `marcas` (  
  `id` int(11) NOT NULL,  
  `descripcion` text,  
  `nombre` varchar(100) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```



Tabla Reseñas:

```
CREATE TABLE `resenas` (  
  `id` int(11) NOT NULL,  
  `calificacion` int(11) NOT NULL,  
  `comentario` text,  
  `fecha_resena` date NOT NULL,  
  `idcliente` int(11) DEFAULT NULL,  
  `idproducto` int(11) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Ejemplo de cómo sería introducir datos en la tabla de productos:

```
INSERT INTO `productos` (`id`, `cantidad_en_stock`, `descripcion`, `imagen`,  
  `nombre`, `precio`, `idcategoria`, `idmarca`) VALUES  
(7, 9, 'El Shure SM7B es un micrófono dinámico de estudio con un rango de  
frecuencia plano para grabaciones limpias y naturales, muy valorado por su  
aislamiento de ruido y su capacidad para capturar voces con claridad y  
calidez.', 'imagen7.jpg', 'Shure SM 7 B', 389.00, 4, 1);
```

- Arquitectura del proyecto: Tecnologías utilizadas en cada parte del proyecto

La aplicación seguirá una arquitectura basada en Java y utilizará los siguientes componentes clave:

- Capa de controladores (paquete 'controladores'): Aquí residirán las clases Java que gestionarán las solicitudes HTTP entrantes. Utilizaremos Spring MVC para este propósito. Los controladores dirigirán las solicitudes a las operaciones adecuadas en la capa de servicios.
- Capa entidades (paquete 'entidades'): Las clases en esta capa representan las entidades principales en nuestra base de datos, como Productos, Clientes, Pedidos y otras.
- Capa de repositorios (paquete 'repositorios'): Los repositorios serán responsables de interactuar con la base de datos a través de Spring Data JPA. Ofrecen operaciones CRUD en las entidades y simplifican la comunicación con la base de datos.
- Capa de servicios (paquete 'servicios'): Aquí implementaremos la lógica de negocio de la aplicación. Las clases de servicio coordinarán la interacción entre los controladores y los repositorios.
- Capa de configuración (paquete 'config'): Este paquete contendrá clases de configuración para establecer aspectos como la conexión con la base de datos y configuraciones relacionadas con la seguridad.



- Capa de seguridad (paquete 'seguridad'): Implementaremos medidas de seguridad utilizando Spring Security para garantizar la autenticación y la autorización de los usuarios.

Base de Datos:

En el desarrollo de la tienda online, he optado por utilizar MySQL como sistema de gestión de base de datos, específicamente con el motor de almacenamiento InnoDB. InnoDB es una elección fundamental para aplicaciones que requieren un procesamiento confiable y seguro de transacciones, gracias a sus características de integridad referencial y soporte para transacciones ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad). Esta elección asegura un manejo eficiente de los datos, esencial para las operaciones comerciales y la gestión de usuarios de nuestra plataforma.

Frontend:

Para la interfaz de usuario, se utilizará HTML5, proporcionando una estructura clara. CSS se empleará para el diseño y la estética visual. Thymeleaf será el motor de plantillas para generar vistas dinámicas y permitir una integración fluida con los modelos de datos de Spring.

Gestión de dependencias:

Para la gestión de dependencias y la construcción del proyecto, se utilizará Maven. Esto permitirá un manejo automatizado de las librerías necesarias.

En resumen, la aplicación se basará en tecnologías Java, con Spring MVC para la gestión de solicitudes web y Spring Data JPA para la integración con la base de datos MySQL. El frontend se construirá con HTML5 y CSS, utilizando Thymeleaf como motor de plantillas para las vistas. Maven se encargará de la gestión de dependencias, facilitando la construcción y el despliegue del proyecto. Esta arquitectura proporcionará una base sólida para el desarrollo de una tienda en línea de instrumentos musicales.

1.4.3. Implementación y pruebas

Desarrollo de la aplicación con los lenguajes y plataformas elegidas:

La aplicación será desarrollada en el marco de desarrollo Spring, en particular, Spring Boot y Spring MVC, que se basa en Java. Esto proporciona una base sólida para desarrollar una aplicación web de compra de instrumentos musicales y equipamiento de audio profesional.

A continuación, explicaré algunos de los aspectos más relevantes del código fuente y los scripts utilizados para el proyecto:



1. Estructura del proyecto:
El proyecto estará estructurado en paquetes que siguen el enfoque de la arquitectura MVC (Model-View.Controller). Esto facilita la organización del código.
2. Controladores:
Los controladores son los responsables de manejar solicitudes HTTP entrantes. En el código fuente, encontrarás clases de controladores que manejan rutas como la página de inicio, la lista de productos, el carrito de compras, etc.
3. Modelos:
Los modelos representan las entidades y la lógica de negocio. En el proyecto los modelos representan las tablas de la base de datos, como Clientes, Productos, Compras, etc. Se utilizan anotaciones de Spring como '@Entity' y '@Repository', para definir las clases de modelos y facilitar la interacción con la base de datos.



4. Vistas:
Las vistas son las páginas HTML que se muestra al usuario. En el proyecto, se utilizarán plantillas HTML y Thymeleaf como motor de plantillas para generar las páginas dinámicas. En los controladores, se especifican las vistas a las que se debe dirigir el usuario.
5. Seguridad:
Se implementará la seguridad del usuario y la autenticación para garantizar que solo los usuarios autorizados tengan acceso a ciertas áreas de la aplicación. Se utilizarán anotaciones de seguridad de Spring como '@Secured' para restringir el acceso a rutas específicas.
6. Persistencia de Datos:
Se utilizará Spring Data JPA para interactuar con la base de datos MySQL. Las consultas SQL se generarán automáticamente a partir de métodos de Spring Data JPA, lo que simplifica el acceso a los datos.
7. Mapeo de entidades:
Las entidades de la base de datos se mapean a objetos Java utilizando anotaciones de JPA, como '@Entity', '@Table', '@ManyToOne', '@OneToMany', lo que permite una fácil manipulación de los datos.
8. Script SQL
En la fase de desarrollo es posible que necesite scripts SQL para inicializar la base de datos con datos de prueba o configuraciones iniciales. Estos scripts se pueden ejecutar en el servidor de la base de datos MySQL para crear tablas y cargar datos iniciales.
9. Configuración:
Spring Boot permite una configuración centralizada a través de archivos de propiedades. Esto facilitará la gestión de la configuración de la aplicación, como las credenciales de la base de datos o la configuración de la seguridad.
10. Manejo de sesiones y carrito de compras:
Utilizaremos cookies y sesiones para el manejo de sesiones y el carrito de compras. Esto ofrece persistencia de datos y una experiencia continua para los usuarios.



1.4.4. Implantación y documentación

Herramientas requeridas el funcionamiento de la aplicación:

- Java Development Kit (JDK) 21: Es necesario tener instalado JDK 21 para compilar y ejecutar la aplicación.
- IDEA IntelliJ: Un entorno de desarrollo integrado (IDE) que soporta Spring Boot y Maven.
- Docker Desktop: Herramienta para crear y manejar contenedores de Docker.

Manual de instalación:

- Clonar el repositorio o descargar el paquete entregado
- Importar el proyecto a IntelliJ y resolver dependencias.
- Instalar Docker Desktop desde su página oficial y asegurarse de que esté ejecutándose.
- Crear un archivo Dockerfile en el directorio raíz del proyecto con el siguiente contenido:

```
FROM openjdk:21-jdk-slim

WORKDIR /app
#Copiar el jar del proyecto
COPY target/tiendamusica-0.0.1-SNAPSHOT.jar /app/tiendamusica.jar

EXPOSE 8080

ENTRYPOINT [ "java" , "-jar" , "/app/tiendamusica.jar" ]
```



- Crear un archivo docker-compose.yml en el directorio raíz del proyecto con el siguiente contenido:

```
version: '3.8'
services:
  db:
    image: mysql
    container_name: tiendamusica_db
    environment:
      MYSQL_ROOT_PASSWORD: root
      MYSQL_DATABASE: tiendamusica
      MYSQL_USER: tiendamusica
      MYSQL_PASSWORD: tiendamusica
    ports:
      - '3366:3306'
    volumes:
      - 'dbdata:/var/lib/mysql'
      - './tiendamusica.sql:/docker-entrypoint-initdb.d/tiendamusica.sql'

  app:
    build: .
    container_name: tiendamusica_app
    ports:
      - '9000:9000'
    depends_on:
      - db
    environment:
      #-
      SPRING_DATASOURCE_URL=jdbc:mysql://db:3306/tiendamusica?useSSL=false&serverTimezone=UTC&useLegacyDatetimeCode=false&allowPublicKeyRetrieval
      - SPRING_DATASOURCE_URL=jdbc:mysql://db:3306/tiendamusica
      - SPRING_DATASOURCE_USERNAME=tiendamusica
      - SPRING_DATASOURCE_PASSWORD=tiendamusica

volumes:
  dbdata:
```

- En el directorio del proyecto, abrir una terminal y ejecutar el siguiente comando para construir y levantar los contenedores:

```
docker-compose up --build
```

- Navegar a <http://localhost:9000> en un navegador web para acceder a la aplicación.



- Otra documentación si fuera necesaria (como Anexos):

Instalación de Docker Desktop:

- Visita la página oficial de Docker Desktop y seleccione la versión correspondiente a su sistema operativo (Windows, Linux, Mac).
- Descarga el instalador y ejecuta el archivo descargado
- Sigue las instrucciones del asistente de instalación para completar el proceso.
- Una vez instalado, asegúrate de que Docker Desktop esté ejecutándose y funcionando correctamente.



1.4.5. Resultados y discusión

- Comentarios sobre el desarrollo, detallando la temporalización real respecto a la planificada

El desarrollo del proyecto ha tomado más tiempo del inicialmente previsto. Compaginar las responsabilidades del proyecto con el trabajo ha sido un reto significativo, lo cual ha afectado al cronograma. Además, la aparición de errores técnicos comunes ha llevado a retrasos adicionales.

Estos imprevistos han requerido un esfuerzo extra para la resolución de problemas y ajustes en la gestión del tiempo.

- Dificultades más importantes encontradas
 - Adaptación después de un Período sin Programar: Retomar el proyecto después de un largo período sin práctica en programación, debido a la naturaleza no técnica del trabajo diario, resultó ser un desafío significativo. Requirió un esfuerzo adicional para recuperar y actualizar conocimientos, así como para ajustarse nuevamente al ritmo y las demandas del desarrollo del proyecto.
 - Compaginación con Trabajo Diario: Planificar y administrar el tiempo para avanzar en el proyecto mientras se cumplen las responsabilidades laborales diarias fue un desafío continuo. El equilibrio entre ambas actividades demandó una gestión efectiva del tiempo y un compromiso constante para mantener el progreso del proyecto.

- Posibles mejoras y ampliaciones del proyecto

Entre las mejoras y ampliaciones consideradas para el proyecto de creación de la tienda online, se destacan las siguientes propuestas para enriquecer la funcionalidad y la experiencia del usuario:

- Mejora del Frontend de la Página:
Actualización y embellecimiento del diseño de la interfaz para mejorar la atracción visual y la usabilidad del sitio.
- Añadir una pestaña de la comunidad:
Implementación de una sección comunitaria donde los usuarios puedan compartir videos, consejos y participar en un blog o foro.
- Plataforma de Compraventa de Segunda Mano:
Inclusión de una opción para comprar y vender productos de segunda mano relacionados con la música.
- Servicios como Profesor de Instrumento: Facilitar la oferta y demanda de servicios de enseñanza musical dentro de la plataforma.



1.5. Conclusiones

- Qué ha supuesto para tu formación la realización de este proyecto

La realización de este proyecto ha sido una experiencia enriquecedora para mi formación. Me ha dado la oportunidad de aplicar los conocimientos adquiridos durante el ciclo en un entorno práctico y realista.

- En qué medida te ha servido para ampliar conocimientos

Este proyecto ha ampliado significativamente mis conocimientos, centrándose sobre todo en la programación backend, la gestión de bases de datos y la implementación de prácticas de seguridad en aplicaciones web. También mejorando en el uso de tecnologías específicas como Java, Spring Boot y MySQL.

1.6. Bibliografía y referencias

1. Curso de Spring Boot y Spring MVC y curso de Thymeleaf
<https://openwebinars.net/>
2. Recursos y contenedores para implementación en Docker.
<https://hub.docker.com/>
3. Búsqueda de errores puntuales
<https://es.stackoverflow.com/>
4. Pagina de referencia
<https://www.thomann.de/es/index.html>
5. Resolución de errores puntuales
<https://chat.openai.com/>
6. Creación de un logo personalizado para la portada del proyecto
<https://openai.com/dall-e-2>
7. Documentación de MYSQL
<https://dev.mysql.com/doc/>