

PROYECTO FINAL

NOMBRE: JOSEBETH LLUMIQUINGA

DOCUMENTACIÓN

Descripción del Proyecto

Objetivos del proyecto.

El objetivo principal es hacer una tienda de chocolate online y proporcionar a los clientes una plataforma conveniente y accesible para comprar una amplia variedad de chocolates de manera atractiva.

Esta tienda debe proporcionar y ofrecer una experiencia de usuario atractiva y de compra fácil, segura y agradable, satisfaciendo las necesidades y preferencias de los amantes del chocolate.

Funcionalidades Principales:

Catálogo de Productos:

Visualización de diferentes chocolates con detalles como nombre, descripción, precio, y disponibilidad.

Filtrado y búsqueda por categorías (ej., chocolate negro, blanco, con leche)

Página de Detalle del Producto:

Información detallada de cada chocolate, incluyendo ingredientes, opiniones, y recomendaciones.

Carrito de Compras:

Añadir productos al carrito con opciones de actualización de cantidades y eliminación.

Vista previa del total del carrito en cualquier momento.

Proceso de Compra:

Flujo de checkout con pasos de confirmación de productos, introducción de detalles de envío y pago.

Integración con un sistema de pago (ej., tarjeta de crédito, PayPal).

Autenticación y Gestión de Usuarios:

Registro e inicio de sesión de usuarios.

Perfil de usuario con historial de pedidos y detalles de la cuenta.

Panel de Administración:

Gestión de productos (añadir, editar, eliminar).

Gestión de pedidos y usuarios.

Público Objetivo:

Amantes del chocolate que buscan comprar productos en línea.

Propietarios de la tienda que desean gestionar su catálogo y pedidos.

Planificación

Cronograma del Proyecto:

4 al 7 de junio (3 días): Análisis de requisitos y diseño del sistema (documentación).

6 al 10 de junio (4 días): Desarrollo del frontend.

11 al 14 (3 días): Desarrollo del backend y base de datos.

13 y 14 de junio (2 días): Pruebas y despliegue.

Requisitos Funcionales:

- Visualización de productos.
- Carrito de compras.
- Autenticación de usuarios.
- Gestión de productos y pedidos

Requisitos No Funcionales:

- Rendimiento rápido.
- Diseño responsivo.
- Seguridad en las transacciones.

Estimaciones de Tiempo y Recursos:

Desarrollo Frontend: 1semana.

Desarrollo Backend: 1 semana.

Pruebas y Despliegue:1semana.

ANALISIS Y DISEÑO

Diagramas de flujo y UML

Diagrama de Casos de Uso para la tienda de chocolates en línea

Entidades

Clase Usuario

ATRIBUTOS

ID: INT, PK, AUTO_INCREMENT

nombre: VARCHAR(100), NOT NULL

correoElectronico: VARCHAR(100), NOT NULL

contrasena: VARCHAR(100), NOT NULL

METODOS

Registrar Usuario

Iniciar Sesión

Actualizar Información

Eliminar Cuenta

Clase Producto

ATRIBUTOS

ID: INT, PK, AUTO_INCREMENT

nombre: VARCHAR(100), NOT NULL

descripcion: TEXT

precio: DECIMAL(10, 2), NOT NULL

imagen: VARCHAR(200)

stock: INT, NOT NULL

METODOS

Agregar Producto

Actualizar Producto

Eliminar Producto

Consultar Producto

Ver Stock

Clase Carrito

ATRIBUTOS

ID: INT, PK, AUTO_INCREMENT

usuarioID: INT, FK, REFERENCES Usuario(ID)

productoID: INT, FK, REFERENCES Producto(ID)

cantidad: INT

METODOS

Agregar al Carrito

Eliminar del Carrito

Actualizar Cantidad

Vaciar Carrito

Ver Carrito

Clase Reseña

ATRIBUTOS

ID: INT, PK, AUTO_INCREMENT

usuarioID: INT, FK, REFERENCES Usuario(ID)

productoID: INT, FK, REFERENCES Producto(ID)

rating: INT

comentario: TEXT

METODOS

Agregar Reseña

Actualizar Reseña

Eliminar Reseña

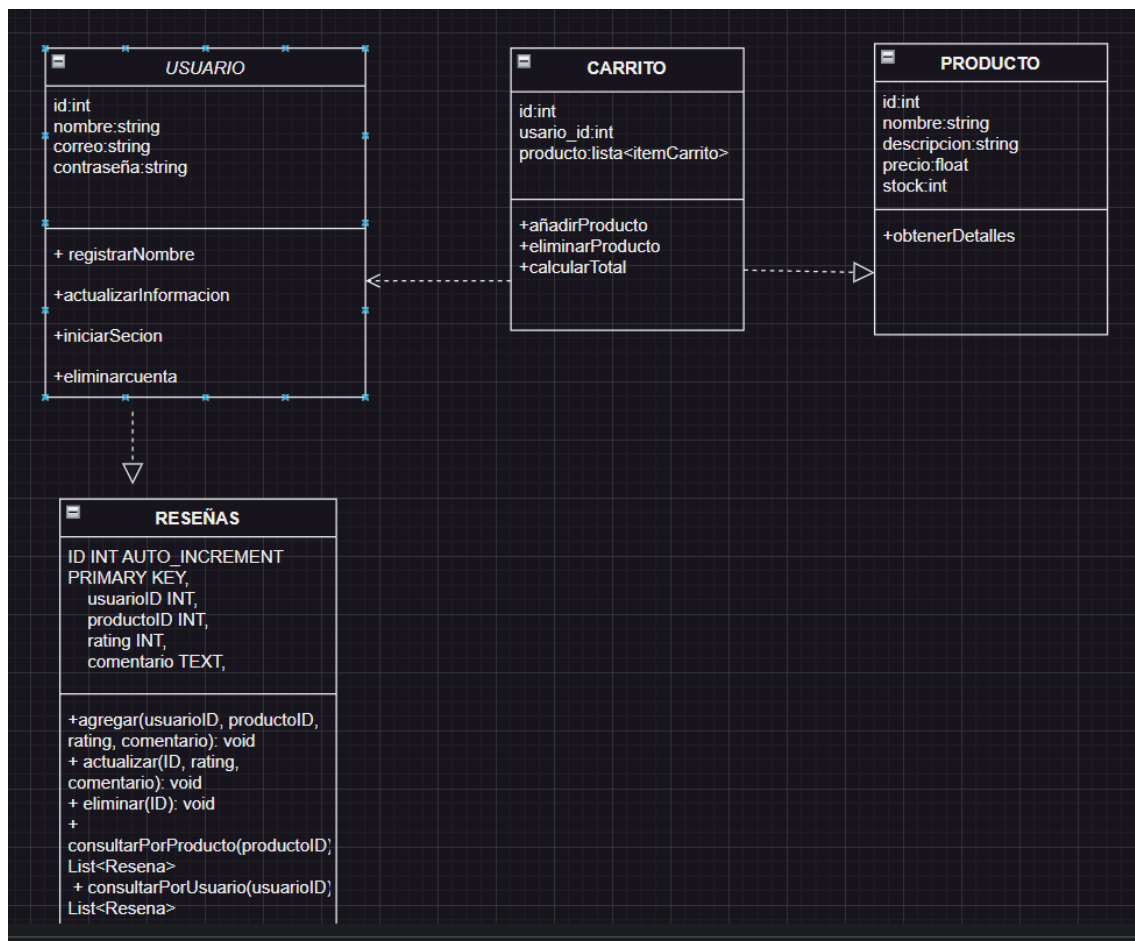
Consultar Reseñas por Producto

Consultar Reseñas por Usuario

Relaciones

- Usuario tiene una relación uno a muchos con Carrito y Resena.
- Producto tiene una relación uno a muchos con Carrito y Resena.
- Carrito relaciona Usuario con Producto en una relación de muchos a muchos con una entidad intermedia (Carrito).
- Reseña relaciona Usuario con Producto para la función de reseñar.

DIAGRAMA UML



Mockups y Wireframes de la Interfaz de Usuario:

Página de Inicio:

- Cabecera con logo y menú de navegación.
- Banner promocional.
- Listado de productos destacados.

Página de Detalle del Producto:

- Imagen del producto.
- Descripción, precio, y botón para añadir al carrito.

Carrito de Compras:

- Listado de productos añadidos.
- Opciones para modificar cantidades o eliminar productos.

Proceso de Checkout:

- Formularios para dirección de envío y detalles de pago.

Diseño de la Arquitectura del Sistema:

- **Frontend:** HTML, CSS, JavaScript, Bootstrap.
- **Backend:** Java, Spring Boot, Hibernate.
- **Base de Datos:** MySQL o PostgreSQL.

IMPLEMENTACION

Explicación del código utilizado

Descripción detallada de las tecnologías y herramientas utilizadas

Lenguaje de Programación: JavaScript (para Backend).

Base de Datos: MySQL

Frontend: HTML, CSS, JavaScript, y un framework como Bootstrap para diseño.

Herramientas:

IDE: Visual Studio Code.

Gestor de Base de Datos: MySQL Workbench.

Sistema de Control de Versiones: Git.

Entorno Virtual: Virtualenv.

Bibliotecas: SQLAlchemy para ORM, Flask o Django Rest Framework para la API, Axios o Fetch API para llamadas AJAX.

BASE DE DATOS

```
CREATE DATABASE SHUNGOLATE;
USE SHUNGOLATE;

CREATE TABLE Usuario (
  ID INT AUTO_INCREMENT PRIMARY KEY,
  nombre VARCHAR(100) NOT NULL,
  correoElectronico VARCHAR(100) NOT NULL,
  contrasena VARCHAR(100) NOT NULL
);

INSERT INTO Usuario (nombre, correoElectronico, contrasena)
VALUES
  ('Juan Pérez', 'juan@gmail.com', 'clave123'),
  ('María López', 'maria@gmail.com', 'clave456'),
  ('Pedro Martínez', 'pedro@gmail.com', 'clave789'),
  ('Ana García', 'ana@gmail.com', 'claveabc'),
  ('Luis Rodríguez', 'luis@gmail.com', 'clavexyz'),
  ('Laura Fernández', 'laura@gmail.com', 'clave567'),
  ('Carlos Sánchez', 'carlos@gmail.com', 'clave890'),
  ('Sofía Ramírez', 'sofia@gmail.com', 'clave1234'),
  ('Daniel Gómez', 'daniel@gmail.com', 'clave5678'),
  ('Elena Torres', 'elena@gmail.com', 'claveabcd');

CREATE TABLE Producto (
  ID INT AUTO_INCREMENT PRIMARY KEY,
  nombre VARCHAR(100) NOT NULL,
  descripcion TEXT,
  precio DECIMAL(10, 2) NOT NULL,
  imagen VARCHAR(200),
```

```
(9, 10, 1);

CREATE TABLE Resena (
  ID INT AUTO_INCREMENT PRIMARY KEY,
  usuarioID INT,
  productoID INT,
  rating INT,
  comentario TEXT,
  FOREIGN KEY (usuarioID) REFERENCES Usuario(ID),
  FOREIGN KEY (productoID) REFERENCES Producto(ID)
);

-- Ejemplo de registros en la tabla Resena
INSERT INTO Resena (usuarioID, productoID, rating, comentario)
VALUES
  (1, 1, 4, 'Muy buen chocolate, pero un poco amargo para mi gusto.'),
  (2, 3, 5, '¡Mi chocolate favorito! Dulce y suave.'),
  (3, 5, 3, 'Me esperaba más del relleno de caramelo.'),
  (4, 2, 4, 'Buen sabor, pero un poco caro.'),
  (5, 4, 5, 'Excelente combinación de chocolate y almendras.'),
  (6, 6, 2, 'No soy fan de las avellanas, pero el chocolate es bueno.'),
  (7, 7, 4, 'Me encanta el toque de naranja en este chocolate.'),
  (8, 8, 5, 'Perfecto para después de las comidas.'),
  (9, 9, 3, 'Las fresas le dan un sabor fresco al chocolate.'),
  (10, 10, 4, 'El chocolate con pasas es una combinación interesante.');
```

HTML FRONTEND

```

<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Tienda de Chocolates</title>
  <link rel="stylesheet" href="styles.css">
  <!-- Font Awesome para iconos -->
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/font-awesome/6.0.0-beta3/css/all.min.css">
</head>
<body>
  <div class="container">
    <header class="header">
      <div class="header-content">
        <h1>Delicias de Chocolate</h1>
        <nav>
          <ul class="nav-links">
            <li><a href="#productos">Productos</a></li>
            <li><a href="#carrito">Carrito</a></li>
            <li><a href="#resenas">Resenas</a></li>
          </ul>
        </nav>
      </div>
    </header>

    <main>
      <!-- Productos -->
      <section id="productos">
        <h2>Catálogo de Chocolates</h2>
        <div class="product-list" id="productList"></div>
      </section>

      <!-- Carrito -->
      <section id="carrito">
        <h2>Tu Carrito</h2>
        <table class="cart-table">
          <thead>
            <tr>
              <th>Producto</th>
              <th>Cantidad</th>
              <th>Precio Unitario</th>
              <th>Total</th>
              <th>Acciones</th>
            </tr>
          </thead>
          <tbody id="checkoutCart"></tbody>
        </table>
        <div class="cart-total">
          <h3>Total:</h3>
        </div>
      </section>
    </main>
  </div>
</body>
</html>

```

BACKEND

```

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;

import java.util.List;
import java.util.Optional;

@RestController
@RequestMapping("/carrito")
public class CarritoController {

    @Autowired
    private CarritoRepository carritoRepository;

    @GetMapping
    public List<Carrito> getAllCarritos() {
        return carritoRepository.findAll();
    }

    @GetMapping("/{id}")
    public ResponseEntity<Carrito> getCarritoById(@PathVariable Long id) {
        Optional<Carrito> carrito = carritoRepository.findById(id);
        if (carrito.isPresent()) {
            return ResponseEntity.ok(carrito.get());
        } else {
            return ResponseEntity.notFound().build();
        }
    }

    @PostMapping
    public Carrito createCarrito(@RequestBody Carrito carrito) {
        return carritoRepository.save(carrito);
    }

    @PutMapping("/{id}")
    public ResponseEntity<Carrito> updateCarrito(@PathVariable Long id, @RequestBody Carrito carritoDetails)

```

Pruebas

Plan de Pruebas

Pruebas Unitarias:

Pruebas para métodos CRUD en el backend.

Validación de datos de entrada.

Pruebas de Integración:

Verificar que las APIs se comuniquen correctamente con la base de datos.

Asegurar que el frontend muestre datos correctos desde la API.

Pruebas de Sistema:

Flujo completo de registro de usuario, navegación, selección de productos y compra.

Pruebas de interfaz para verificar la usabilidad.

Resultados de las Pruebas

Caso de Prueba	Resultado	Observaciones
Registro de Usuario	Aprobado	El usuario se registró correctamente.
Agregar Producto al Carrito	Aprobado	Producto agregado al carrito sin errores.
Realizar Compra	Aprobado	Compra realizada correctamente con actualización de stock.
Resena de Producto	Aprobado	Reseña agregada correctamente a la base de datos.

Casos de Prueba y su Documentación

Caso de Prueba 1: Registro de Usuario

Objetivo: Verificar que un nuevo usuario pueda registrarse.

Pasos:

Ir a la página de registro.

Llenar el formulario con datos válidos.

Enviar el formulario.

Resultado Esperado: Usuario registrado exitosamente y mensaje de confirmación mostrado.

Caso de Prueba 2: Agregar Producto al Carrito

Objetivo: Verificar que un producto se pueda agregar al carrito.

Pasos:

Navegar a la página del producto.

Hacer clic en el botón "Agregar al Carrito".

Resultado Esperado: Producto agregado al carrito y cantidad actualizada.

MANTENIMIENTO

Posibles Mejoras y Actualizaciones

Funcionalidades:

Agregar sistema de cupones de descuento.

Implementar notificaciones por email.

Interfaz de Usuario:

Mejorar la UI/UX con un diseño más moderno.

Implementar una versión móvil de la aplicación.

Manual del Usuario

Instrucciones Detalladas para el Uso de la Aplicación

Registro e Inicio de Sesión:

Navegar a la página de registro y completar el formulario.

Usar el correo electrónico y la contraseña para iniciar sesión.

Navegación y Compra:

Navegar a la sección de productos para ver los chocolates disponibles.

Seleccionar un producto y hacer clic en "Agregar al Carrito".

Revisar el carrito y proceder al pago.

Gestionar Perfil:

Ir a la página de perfil para actualizar la información personal.

Cambiar la contraseña si es necesario.

Solución de Problemas Comunes

Problema: No puedo iniciar sesión.

Solución: Verificar que el correo electrónico y la contraseña sean correctos.

Solución Alternativa: Usar la opción "Recuperar Contraseña" para restablecer la contraseña.

Problema: No se agregan productos al carrito.

Solución: Asegurarse de estar conectado a internet y que la API esté funcionando.

Solución Alternativa: Limpiar la caché del navegador y volver a intentarlo.

Problema: La página no carga correctamente.

Solución: Verificar la conexión a internet y recargar la página.

Solución Alternativa: Probar en otro navegador o dispositivo.

Este plan proporciona una visión completa de la implementación, pruebas, despliegue, mantenimiento y manual del usuario para la tienda de chocolates. Puedes ajustar y expandir cada sección según los detalles específicos de tu aplicación y sus requisitos.