

Mini E-Commerce

Caso de implementación N° 1.

Descripción del caso de implementación

Implementar una plataforma compuesta por un backend y un frontend, donde se cumplan los módulos y las reglas de negocio mencionadas en este documento.

Módulo de usuarios

Descripción

El módulo de usuario contiene las funcionalidades para gestionar los diferentes usuarios de la plataforma y las validaciones de acción para cada uno de los roles.

Funcionalidades

- Gestión de usuarios.
- Validación de privilegios para ejecutar acciones de gestión.

Reglas de negocio

El sistema debe tener 3 roles:

- Administrador
- Empleado
- Cliente

Un administrador debe poder ejecutar las siguientes acciones:

- Registrar empleados
- Bloquear y desbloquear empleados

Módulo de autenticación y autorización

Descripción

Este módulo debe contener las funcionalidades necesarias para identificar los usuarios del sistema (autenticación) y los permisos que tienes para ejecutar las acciones en la plataforma (autorización)

Funcionalidades

- Iniciar y cerrar sesión
- Registro de usuarios
- Reiniciar contraseña

Reglas de negocio

- Un usuario no puede iniciar sesión si está en estado bloqueado
- Después de 3 intentos fallidos de autenticación, se bloquea automáticamente el usuario

Módulo de gestión de productos

Descripción

Este módulo debe contener las funcionalidades necesarias para la gestión de los productos para los empleados y la búsqueda de productos para el cliente.

Funcionalidades

- CRUD de productos
- Búsqueda de productos por nombre

Reglas de negocio

Campos del producto

- Id
 - Entidad: ProductId
 - Tipo de valor: Numérico
 - Restricciones
 - No puede estar nulo
 - Debe ser superior a 1
- Nombre
 - Entidad: Name
 - Tipo de valor: Cadena de texto
 - Restricciones
 - No puede ser nulo
 - No puede estar vacío
 - Puede tener máximo 100 caracteres
- Descripción
 - Entidad: Description
 - Tipo de valor: Cadena de texto
 - Restricciones

- No puede ser nulo
 - No puede estar vacío
 - Puede tener máximo 280 caracteres
- Precio base
 - Entidad: BasePrice
 - Tipo de valor: Decimal
 - Restricciones
 - No puede ser nulo
 - No puede ser inferior a 0
- Tasa de impuestos
 - Entidad: TaxRate
 - Tipo de valor: Decimal
 - Restricciones
 - No puede ser nulo
 - No puede ser inferior a 0
 - No puede ser superior a 1
- Estado: Utilizar un objeto enumerable (ENUM)
 - Entidad: ProductStatus
 - Tipo de valor: ENUM
 - Posibles valores:
 - Borrador
 - Publicado
- Cantidad en inventario
 - Entidad: InventoryQuantity
 - Tipo de valor: Numérico
 - Restricciones
 - No puede ser nulo
 - No puede ser inferior a 0

Módulo de órdenes

Descripción

Este módulo debe contener las funcionalidades necesarias para que un cliente pueda crear, pagar y ver el estado de sus órdenes, y además, para que un empleado pueda hacer gestión de las órdenes y confirmar sus diferentes estados.

Funcionalidades

- Cliente: Crear órdenes
- Cliente: Pagar orden
- Cliente: Ver el estado de sus órdenes
- Empleado: Confirmar estado de orden
- Empleado: Gestión de órdenes (CRUD)

Reglas de negocio

- Las órdenes deben componerse de uno o múltiples productos de la plataforma.
- Cada 3ra orden del mismo cliente tiene 10% de descuento.
- Las órdenes tienen 3 estados:
 - Registrada: Cuando el cliente crea la orden.
 - Pagada: Cuando el cliente transfiere fondos a la plataforma. Los empleados deben poder verificar esto desde la bandeja de órdenes (Dashboard).
 - Entregada: Cuando los productos son entregados al cliente. Los empleados deben poder verificar esto desde la bandeja de órdenes (Dashboard).
- Los estados de las órdenes deben ser notificados en tiempo real y todas las personas involucradas (cliente y empleados) deben poder ver el último estado de la orden
- Para que un cliente pague una orden, debe presionar un botón “Pagar” en el detalle de la orden.

Pruebas (Testing)

- Debe incluir 3 pruebas unitarias en el Backend
- Debe incluir 2 pruebas de integración en el Backend
- Debe incluir 3 pruebas unitarias en el Frontend
- Puede incluir 2 pruebas de integración en el Frontend
- Puede incluir una prueba End-to-End (E2E)

Documentación

- Incluir documentación para 1 de los módulos.

Consideraciones técnicas

Se puede utilizar Docker y Docker-compose para la creación y gestión del ambiente de desarrollo.

Stack Tecnológico

- Puede utilizar como lenguaje de backend Java (Spring), Node JS Javascript o TypeScript (Express o GraphQL) o Python (Django), esto según la posición a la cual esté aplicando en la empresa.
- Puede utilizar para el frontend Angular o React.
- Puede utilizar para la base de datos MongoDB, PostgreSQL o MySQL.