

# **Ejercicio 8**

## **Metodologías Ágiles**

Alejo Nardon

# Índice:

Índice:.....	2
1. Análisis de Requerimientos.....	3
Ejercicio 1).....	3
Requisitos No Funcionales:.....	3
Ejercicio 2).....	4
2. Diseño de sistema.....	6
Ejercicio 3).....	6
Ejercicio 4).....	6
4. Diseño del programa.....	7
Ejercicio 5).....	7
Ejercicio 6).....	7
4. Diseño.....	9
Ejercicio 7).....	9
Diagrama de Dominio:.....	9
Diagrama de Robustez:.....	10
Diagrama de Secuencia:.....	10
Diagrama de Clases:.....	10
Prototipos:.....	11
5. Pruebas.....	11
Ejercicio 9).....	11
6. Despliegue del Programa.....	12
Ejercicio 11).....	12
7. Mantenimiento.....	12
Ejercicio 13).....	12
8. Nos preparamos para nuevos retos.....	13

# 1. Análisis de Requerimientos

## Ejercicio 1)

### *Requisitos Funcionales:*

#### **ABM de Productos**

#### **ABM de Categorías**

#### **Control de Stock**

- Actualizar la cantidad de productos en stock.
- Registrar entradas y salidas de inventario.

#### **Búsqueda y Filtrado**

- Buscar productos por nombre, categoría o código.
- Filtrar productos por categoría, cantidad en stock, etc.

#### **Reportes**

- Generar reportes de inventario actual.
- Reportes de movimientos de stock (entradas y salidas).

#### **Alertas**

- Notificaciones para productos con stock bajo.
- Alertas para fechas de caducidad de productos (si aplica).

#### **Gestión de Usuarios y Roles**

### *Requisitos No Funcionales:*

#### **Usabilidad**

- Interfaz fácil de usar y navegar.
- Diseño responsivo para acceso desde dispositivos móviles.

## Rendimiento

- Respuesta rápida a las consultas y actualizaciones de inventario.
- Soportar un número elevado de usuarios concurrentes.

## Seguridad

- Autenticación y autorización de usuarios.
- Encriptación de datos sensibles.

## Mantenimiento

- Sistema de backup y restauración de datos.

## Ejercicio 2)

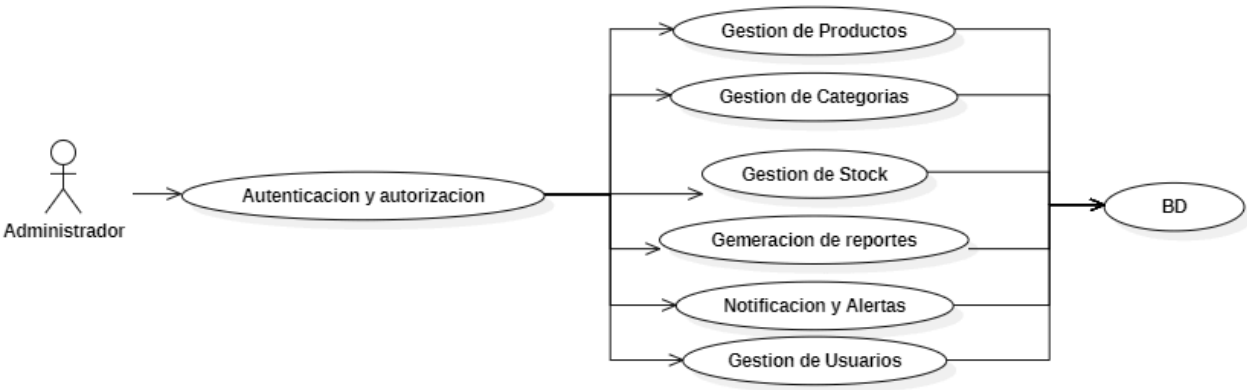
1.

	Especificación de caso de uso	Fecha:2/6/2024
Código	CU-001	
Nombre	Agregar un Nuevo Producto	
Referencias	Requisitos Funcionales, Sección 2.1	
Estado	Pendiente	
Autor	Alejo Nardon	
Revisor	Alejandro Sartorio	
Versión	001	
Descripción	Este caso de uso permite a un administrador agregar un nuevo producto al inventario a través de la aplicación web.	

<b>Actores</b>	Usuario y administrador.
<b>Condición</b>	El actor debe estar autenticado en el sistema.
<b>Pre-condición</b>	El actor ha iniciado sesión correctamente. El actor tiene los permisos necesarios para agregar productos.
<b>CU Extensión</b>	-
<b>Puntos de Extensión</b>	Validación de datos del producto Notificación de producto agregado
<b>Curso Básico</b>	<ol style="list-style-type: none"> <li>1. El actor accede a la sección "Gestión de Productos" desde el menú principal.</li> <li>2. El actor selecciona la opción "Agregar Producto".</li> <li>3. El sistema muestra un formulario para ingresar los detalles del producto.</li> <li>4. El actor completa el formulario con el nombre, descripción, cantidad, precio y categoría del producto.</li> <li>5. El actor hace clic en "Guardar".</li> <li>6. El sistema valida los datos ingresados.</li> <li>7. Si los datos son válidos, el sistema guarda el nuevo producto en la base de datos.</li> <li>8. El sistema muestra un mensaje de confirmación indicando que el producto se ha agregado exitosamente.</li> <li>9. El nuevo producto aparece en la lista de productos del inventario.</li> </ol>
<b>Curso Alternativos</b>	<p>5.1 El actor selecciona el botón "Cancelar".</p> <p>5.2 El sistema retorna al CU-10 Gestionar Tramos.</p> <p>7.2 El administrador ingresa datos inválidos.</p> <p>7.2.1 El sistema muestra un mensaje de error indicando los campos que necesitan corrección.</p> <p>7.2.2 El actor corrige los datos y vuelve a intentar guardar el producto.</p>
<b>Pos-condición</b>	El actor creó con éxito el nuevo producto

# 2. Diseño de sistema

## Ejercicio 3)



## Ejercicio 4)

Inicio

Productos

Usuarios

Stock

Usuario X

Salir

Lista de productos a vencer

Producto XXXX - Vence en 30 dias xx/xx/xxxx

Producto XXXX - Vence en 30 dias xx/xx/xxxx

Producto XXXX - Vence en 30 dias xx/xx/xxxx

Producto XXXX - Vence en 30 dias xx/xx/xxxx

Producto XXXX - Vence en 30 dias xx/xx/xxxx

Lista de Productos mas movidos

Producto XXXX - xx Salidas

Producto XXXX - xx Salidas

Producto XXXX - xx Salidas

Producto XXXX - xx Salidas

Producto XXXX - xx Salidas

Total de productos

1500

Valor Tot de inventario

\$10.000.500

Entradas/Salidas:

Ent: 44 Sal: 61

Productos a vencer:

12 productos

## 4. Diseño del programa

### Ejercicio 5)

La arquitectura de la app será:

- Para el **Back-end**: NodeJs con TypeScript para generar una API que funcione con un fuerte tipado, para mitigar la mayor tasa de errores.
- Para el **Front-end**: Se desarrollará con el framework React
- Para la capa de datos se usará MySQL

La elección de esta arquitectura se debe a su gran eficiencia y su gran utilización en el desarrollo de aplicaciones web apps actuales.

### Ejercicio 6)

Código MySQL de las tablas

```
-- Crear la base de datos
```

```
CREATE DATABASE gestion_inventario;
```

```
USE gestion_inventario;
```

```
-- Crear la tabla Roles
```

```
CREATE TABLE Roles (
```

```
  id INT AUTO_INCREMENT PRIMARY KEY,
```

```
  nombre VARCHAR(50) NOT NULL UNIQUE,
```

```
  descripcion TEXT
```

```
);
```

```
-- Crear la tabla Usuarios
```

```
CREATE TABLE Usuarios (
```

```
  id INT AUTO_INCREMENT PRIMARY KEY,
```

```
nombre VARCHAR(100) NOT NULL,  
email VARCHAR(100) NOT NULL UNIQUE,  
contraseña VARCHAR(255) NOT NULL,  
rol_id INT,  
FOREIGN KEY (rol_id) REFERENCES Roles(id)  
);
```

-- Crear la tabla Categorías

```
CREATE TABLE Categorías (  
id INT AUTO_INCREMENT PRIMARY KEY,  
nombre VARCHAR(100) NOT NULL,  
descripcion TEXT  
);
```

-- Crear la tabla Productos

```
CREATE TABLE Productos (  
id INT AUTO_INCREMENT PRIMARY KEY,  
nombre VARCHAR(100) NOT NULL,  
descripcion TEXT,  
cantidad INT NOT NULL,  
precio DECIMAL(10, 2) NOT NULL,  
categoria_id INT,  
fecha_creacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
fecha_modificacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,  
FOREIGN KEY (categoria_id) REFERENCES Categorías(id)  
);
```

-- Crear la tabla Movimientos de Stock

```
CREATE TABLE Movimientos_de_Stock (  
id INT AUTO_INCREMENT PRIMARY KEY,  
producto_id INT,  
cantidad INT NOT NULL,  
tipo ENUM('entrada', 'salida') NOT NULL,  
fecha TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
usuario_id INT,  
FOREIGN KEY (producto_id) REFERENCES Productos(id),  
FOREIGN KEY (usuario_id) REFERENCES Usuarios(id)
```



);

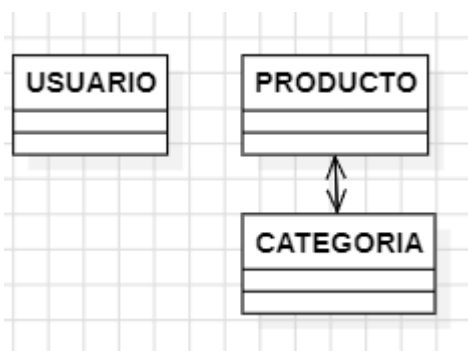
-- Crear la tabla Historial de Cambios

```
CREATE TABLE Historial_de_Cambios (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  usuario_id INT,  
  producto_id INT,  
  accion VARCHAR(100) NOT NULL,  
  fecha TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
  FOREIGN KEY (usuario_id) REFERENCES Usuarios(id),  
  FOREIGN KEY (producto_id) REFERENCES Productos(id)  
);
```

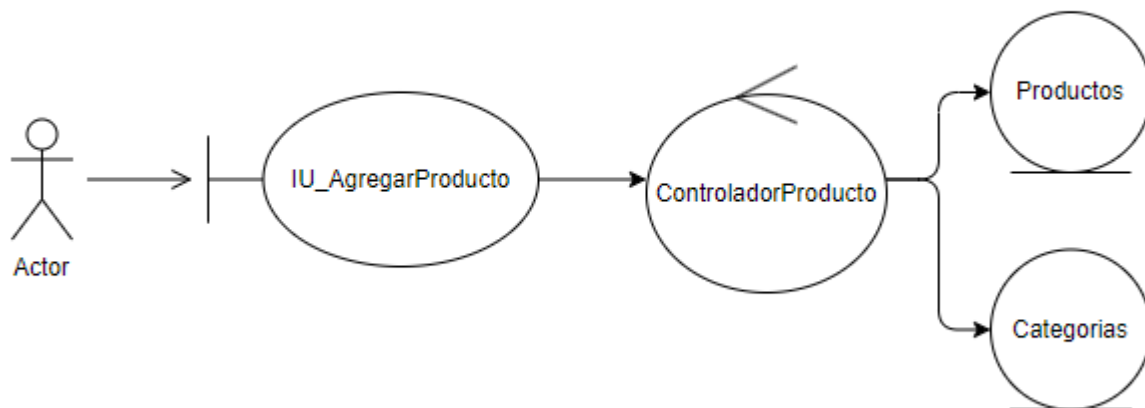
## 4. Diseño

### Ejercicio 7)

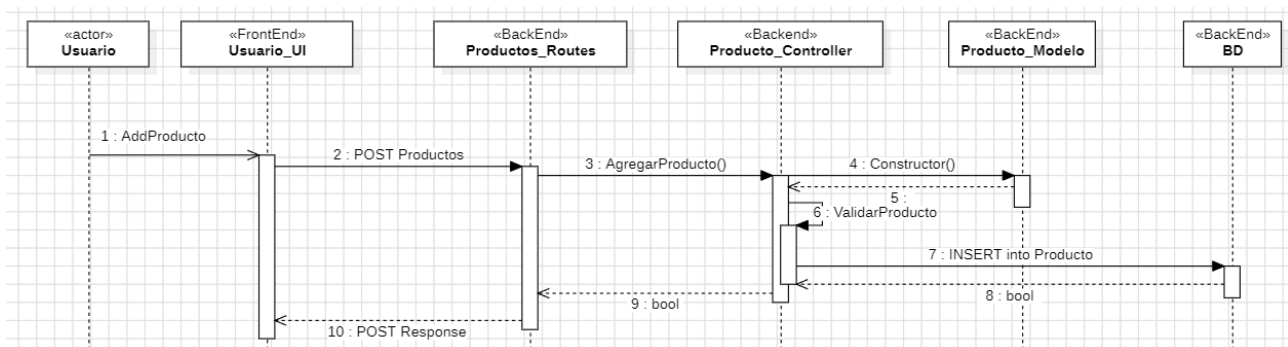
#### Diagrama de Dominio:



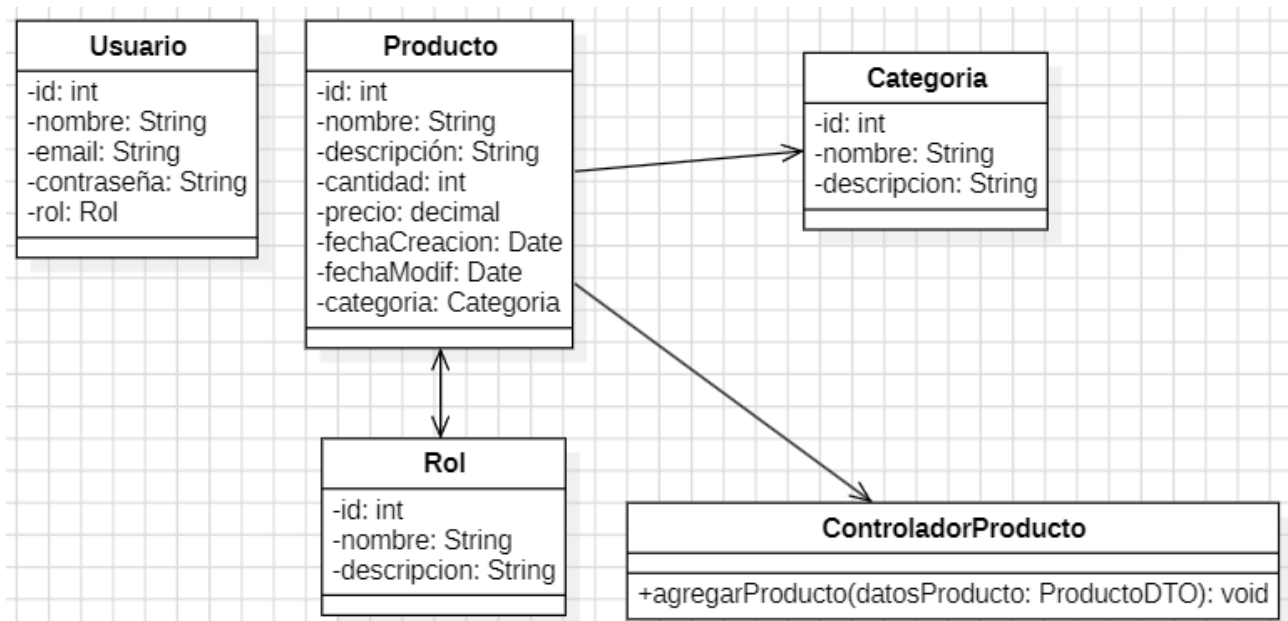
## Diagrama de Robustez:



## Diagrama de Secuencia:



## Diagrama de Clases:



## Prototipos:

El prototipo muestra una interfaz de usuario para agregar un nuevo producto. A la izquierda hay una barra de navegación vertical azul con iconos para inicio, lista, agregar y perfil. El formulario principal, titulado 'Nuevo Producto', contiene los siguientes campos:

- Nombre: Campo de texto único.
- Descripción: Campo de texto único.
- Cantidad: Campo de texto único.
- Precio: Campo de texto único.
- Categoría: Selector de lista con una flecha hacia abajo.
- Siguiente: Botón de acción naranja.

## 5. Pruebas

### Ejercicio 9)

#### Pruebas unitarias para “Agregar un nuevo Producto”

- **Nombre del Producto Vacío:** Verificar que el sistema no permite agregar un producto sin nombre.
- **Descripción del Producto Vacía:** Verificar que el sistema permite agregar un producto sin descripción.
- **Cantidad Negativa:** Verificar que el sistema no permite agregar un producto con cantidad negativa.
- **Precio Negativo:** Verificar que el sistema no permite agregar un producto con precio negativo.

- **Categoría Inexistente:** Verificar que el sistema no permite agregar un producto con una categoría que no existe.

## 6. Despliegue del Programa

### Ejercicio 11)

Plan de despliegue, realizaremos un desarrollo en cascada para la aplicación por lo tanto una vez terminada cada etapa se pasará a la siguiente en el siguiente orden correspondientemente:

1. Análisis
2. Diseño
3. Implementación
4. Testing
5. Despliegue
6. Mantenimiento

## 7. Mantenimiento

### Ejercicio 13)

- Programa actualizaciones de seguridad para todas las dependencias y el sistema operativo.
- Revisa y actualiza regularmente las librerías y frameworks utilizados.
- Configura copias de seguridad automáticas diarias de la base de datos.

- Realiza copias de seguridad del código fuente antes de implementar nuevas versiones.
- Realiza copias de seguridad del código periódicamente de la base de datos.

## 8. Nos preparamos para nuevos retos

### Ejercicio 15)

Equipo de trabajo y roles para el dominio de la app:

- **Jefe de Proyecto:** Planificación, seguimiento del proyecto y Coordinación del equipo y gestión de riesgos.
- **Analista de Sistemas:** Elicitación de requisitos, escritura de especificaciones y casos de uso.
- **Arquitecto de Software:** Diseño de la arquitectura del sistema y base de datos.
- **Desarrolladores de Software:** Desarrollo del código.
- **Administrador de Bases de Datos**
- **Tester**