

**Sprint 1: Configuración del entorno Frontend y Backend + Diseño de la BD**

Tarea	Duración (horas)	Prioridad	Descripción	Validación
Configuración del entorno React	8	Alta	Instalación y configuración inicial de React con JavaScript.	Verificación de que React se ejecute correctamente en el entorno local.
Configuración del entorno Django	8	Alta	Instalación y configuración de Django como backend.	Verificación de que Django funcione correctamente con el servidor local.
Diseño de la base de datos	10	Alta	Creación del diseño del modelo de base de datos en PostgreSQL.	Validación de que las tablas estén correctamente relacionadas y sean funcionales.

**Sprint 2: Desarrollo de la interfaz de login y registro de usuarios**

Tarea	Duración (horas)	Prioridad	Descripción	Validación
Desarrollo del formulario de login en React	8	Alta	Creación del formulario de inicio de sesión en el frontend usando React.	Verificación de que los usuarios puedan iniciar sesión con credenciales válidas.
Implementación de autenticación con JWT	8	Alta	Desarrollo del sistema de autenticación	Verificación de que el sistema de login genera y

			usando tokens JWT en Django.	valida los tokens correctamente.
<b>Conexión frontend con backend</b>	8	Alta	Integración de React con Django para el manejo de la autenticación.	Validación de que las solicitudes de login desde React funcionan correctamente con el backend.

### **Sprint 3: Implementación de la gestión de productos (Agregar, editar, eliminar)**

<b>Tarea</b>	<b>Duración (horas)</b>	<b>Prioridad</b>	<b>Descripción</b>	<b>Validación</b>
<b>Desarrollo del formulario de gestión de productos en React</b>	8	Alta	Creación de formulario en React para agregar, editar y eliminar productos.	Verificación de que se pueda agregar, editar y eliminar productos en el frontend.
<b>Implementación de operaciones CRUD en Django</b>	8	Alta	Desarrollo de endpoints RESTful en Django para manejar las operaciones CRUD de productos.	Validación de que las operaciones CRUD (crear, leer, actualizar, eliminar) funcionen correctamente en el backend.
<b>Conexión frontend con backend para CRUD de productos</b>	8	Alta	Integración de React con Django para las operaciones CRUD de productos.	Verificación de que las operaciones CRUD funcionen correctamente entre el frontend y backend.

**Sprint 4: Gestión de ventas (Registro de ventas y control de stock)**

Tarea	Duración (horas)	Prioridad	Descripción	Validación
Desarrollo del formulario de ventas en React	8	Alta	Creación de formulario en React para registrar ventas.	Verificación de que el formulario registre correctamente las ventas.
Implementación de registro de ventas en Django	8	Alta	Desarrollo de endpoints para registrar ventas y actualizar inventario en Django.	Validación de que las ventas se registran correctamente en la base de datos.
Actualización del inventario tras venta	8	Alta	Desarrollo de lógica en Django para actualizar el inventario después de cada venta.	Verificación de que el stock de productos se actualiza correctamente tras cada venta.

**Sprint 5: Implementación de la predicción de demanda**

Tarea	Duración (horas)	Prioridad	Descripción	Validación
Preparación de datos para Machine Learning	10	Alta	Preprocesamiento de datos históricos de ventas para el entrenamiento del modelo.	Validación de que los datos estén bien preprocesados y listos para el modelo.
Entrenamiento del modelo de predicción de demanda	10	Alta	Desarrollo de un modelo de Machine Learning (regresión lineal)	Validación de que el modelo genere predicciones de

			para predecir la demanda.	demanda precisas.
<b>Exposición del modelo mediante una API REST</b>	8	Alta	Creación de un endpoint en Django para exponer las predicciones de demanda.	Verificación de que el modelo se pueda consultar a través de la API.

### **Sprint 6: Implementación de la funcionalidad de proveedores**

<b>Tarea</b>	<b>Duración (horas)</b>	<b>Prioridad</b>	<b>Descripción</b>	<b>Validación</b>
<b>Desarrollo del formulario de proveedores en React</b>	8	Alta	Creación de formulario para gestionar proveedores en React.	Verificación de que se puede agregar, editar y eliminar proveedores desde el frontend.
<b>Implementación de operaciones CRUD para proveedores en Django</b>	8	Alta	Desarrollo de endpoints RESTful en Django para gestionar proveedores.	Validación de que las operaciones CRUD funcionen correctamente con los proveedores.
<b>Conexión frontend con backend para proveedores</b>	8	Alta	Integración de React con Django para las operaciones CRUD de proveedores.	Verificación de que las operaciones CRUD de proveedores funcionen correctamente entre frontend y backend.

**Sprint 7: Implementación de gestión de caja (Ingreso y salida de dinero)**

Tarea	Duración (horas)	Prioridad	Descripción	Validación
Desarrollo del formulario de caja en React	8	Alta	Creación de formulario en React para registrar ingresos y egresos de dinero.	Verificación de que los ingresos y egresos se registren correctamente en la interfaz.
Implementación de lógica de caja en Django	8	Alta	Desarrollo de lógica en Django para registrar transacciones de caja.	Validación de que las transacciones se registren correctamente en la base de datos.
Conexión frontend con backend para la caja	8	Alta	Integración de React con Django para registrar los ingresos y egresos.	Verificación de que las transacciones de caja se registren correctamente en el backend.

**Sprint 8: Implementación de control de transacciones**

Tarea	Duración (horas)	Prioridad	Descripción	Validación
Desarrollo del sistema de registro de transacciones en React	8	Alta	Creación de formulario para registrar transacciones de ventas.	Verificación de que las transacciones se registren correctamente en la interfaz.
Implementación de registro de	8	Alta	Desarrollo de endpoints para registrar las	Validación de que las transacciones

transacciones en Django			transacciones de ventas en Django.	se guarden correctamente en la base de datos.
Verificación de las transacciones registradas	8	Alta	Verificación de que todas las transacciones se registren correctamente en el sistema.	Validación de que las transacciones se reflejan correctamente en el sistema y base de datos.

### Sprint 9: Reportes de ventas (Generación de reportes y estadísticas)

Tarea	Duración (horas)	Prioridad	Descripción	Validación
Desarrollo de la funcionalidad para generar reportes en React	8	Alta	Crear funcionalidad para generar reportes de ventas en el frontend.	Verificación de que los reportes sean visibles y exportables.
Implementación de generación de reportes en Django	8	Alta	Crear endpoints para generar reportes de ventas desde Django.	Validación de que los reportes se generen correctamente y sean exportables en formatos como CSV o PDF.
Integración de exportación de reportes	8	Alta	Implementación de exportación de reportes a formatos como CSV o PDF.	Verificación de que los reportes se exporten correctamente a los formatos deseados.

### **Sprint 10: Implementación de la funcionalidad de precios y actualizaciones**

<b>Tarea</b>	<b>Duración (horas)</b>	<b>Prioridad</b>	<b>Descripción</b>	<b>Validación</b>
<b>Desarrollo del sistema para actualizar precios en React</b>	8	Alta	Crear funcionalidad para actualizar los precios de los productos.	Verificación de que los precios se actualicen correctamente en la interfaz.
<b>Implementación de actualización de precios en Django</b>	8	Alta	Crear endpoint para actualizar los precios de los productos en la base de datos.	Validación de que los precios se actualicen correctamente en la base de datos y afecten las transacciones.
<b>Conexión entre el frontend y backend para actualizar precios</b>	8	Alta	Integrar la actualización de precios entre React y Django.	Verificación de que los cambios de precio se reflejen correctamente en el sistema.

### **Sprint 11: Implementación de la gestión de stock**

<b>Tarea</b>	<b>Duración (horas)</b>	<b>Prioridad</b>	<b>Descripción</b>	<b>Validación</b>
<b>Desarrollo del sistema de visualización de stock en React</b>	8	Alta	Crear funcionalidad para consultar el stock de productos.	Verificación de que el stock de cada producto se muestra correctamente.
<b>Implementación de actualización de stock en Django</b>	8	Alta	Crear lógica para actualizar el stock de productos	Verificación de que el stock se actualiza correctamente

			después de cada venta.	tras las transacciones.
<b>Integración frontend y backend para la gestión de stock</b>	8	Alta	Conectar la funcionalidad de stock entre React y Django.	Validación de que el stock se actualice en tiempo real después de cada venta.

### **Sprint 12: Pruebas finales y corrección de errores**

<b>Tarea</b>	<b>Duración (horas)</b>	<b>Prioridad</b>	<b>Descripción</b>	<b>Validación</b>
<b>Realización de pruebas unitarias y de integración</b>	8	Alta	Realizar pruebas para verificar la funcionalidad de cada módulo.	Verificación de que todas las funcionalidades del sistema estén operativas.
<b>Corrección de errores encontrados</b>	8	Alta	Corregir errores detectados durante las pruebas.	Validación de que los errores se hayan corregido sin afectar otras funcionalidades.
<b>Verificación final del sistema</b>	8	Alta	Realizar pruebas generales del sistema en su totalidad.	Confirmar que el sistema esté libre de errores y cumpla con los requisitos.



## Historias de Usuario

### Historia de Usuario 1:

**Título:** Registro de productos en el inventario

**Descripción:** Como **administrador**, quiero **registrar nuevos productos** en el inventario para llevar un control adecuado de los artículos disponibles.

**Criterios de Aceptación:**

- El sistema debe permitir ingresar nombre, categoría, precio y cantidad.
- Los datos registrados deben reflejarse en la base de datos correctamente.

### Historia de Usuario 2:

**Título:** Autenticación de usuario

**Descripción:** Como **usuario**, quiero poder **iniciar sesión** con mis credenciales para acceder al sistema y gestionar los productos y ventas.

**Criterios de Aceptación:**

- El sistema debe permitir iniciar sesión con nombre de usuario y contraseña.
- Debe manejar correctamente los intentos fallidos y las credenciales incorrectas.

### Historia de Usuario 3:

**Título:** Gestión de productos

**Descripción:** Como **administrador**, quiero **agregar, editar y eliminar productos** en el inventario para gestionar eficientemente el stock disponible.

**Criterios de Aceptación:**

- El sistema debe permitir agregar, editar y eliminar productos.
- Los cambios deben reflejarse en la base de datos correctamente.

### Historia de Usuario 4:

**Título:** Registro de ventas realizadas

**Descripción:** Como **vendedor**, quiero registrar las ventas realizadas para actualizar automáticamente el inventario y generar reportes de ventas.

**Criterios de Aceptación:**

- El sistema debe permitir registrar los productos vendidos, la cantidad, el precio y el total de la venta.
- El inventario debe actualizarse automáticamente después de cada venta.

### Historia de Usuario 5:

**Título:** Predicción de demanda de productos

**Descripción:** Como **administrador**, quiero que el sistema utilice **Machine Learning** para predecir la demanda futura de productos para optimizar el inventario y evitar faltantes o excesos.

**Criterios de Aceptación:**

- El sistema debe generar predicciones de demanda futura para cada producto.
- Los resultados deben ser mostrados de manera clara y comprensible en la interfaz de usuario.

**Historia de Usuario 6:**

**Título:** Generación de reportes de ventas

**Descripción:** Como **administrador**, quiero generar reportes de ventas por periodo para analizar el rendimiento de las ventas y tomar decisiones informadas.

**Criterios de Aceptación:**

- El sistema debe permitir generar reportes de ventas por fecha, producto y otras métricas.
- Los reportes deben ser exportables en formatos como CSV o PDF.

**Historia de Usuario 7:**

**Título:** Actualización de precios

**Descripción:** Como **administrador**, quiero poder actualizar los precios de los productos para asegurarme de que siempre se reflejen las tarifas correctas en todas las ventas.

**Criterios de Aceptación:**

- El sistema debe permitir actualizar los precios de los productos de manera sencilla.
- Los cambios deben reflejarse en todas las transacciones futuras.

**Historia de Usuario 8:**

**Título:** Gestión de proveedores

**Descripción:** Como **administrador**, quiero gestionar los proveedores para tener un registro actualizado de las compras realizadas y los proveedores con los que trabajo.

**Criterios de Aceptación:**

- El sistema debe permitir agregar, editar y eliminar proveedores.

- Los proveedores deben estar registrados correctamente en la base de datos.

#### **Historia de Usuario 9:**

**Título:** Control de stock

**Descripción:** Como **administrador**, quiero ver y actualizar el stock de los productos en tiempo real para tener un control preciso del inventario disponible.

**Criterios de Aceptación:**

- El sistema debe permitir ver el stock disponible de cada producto.
- El stock debe actualizarse automáticamente tras cada venta o ingreso.

#### **Historia de Usuario 10:**

**Título:** Control de transacciones

**Descripción:** Como **administrador**, quiero tener un registro detallado de todas las transacciones realizadas para poder hacer auditorías y controlar las ganancias del negocio.

**Criterios de Aceptación:**

- El sistema debe registrar cada transacción de manera detallada.
- Las transacciones deben estar accesibles para consultas e informes.

#### **Historia de Usuario 11:**

**Título:** Gestión del flujo de caja

**Descripción:** Como **administrador**, quiero gestionar el flujo de caja, registrando ingresos y egresos de dinero para tener un control de las finanzas del negocio.

**Criterios de Aceptación:**

- El sistema debe permitir registrar ingresos y egresos de dinero.
- Los registros deben reflejarse correctamente en la base de datos.

#### **Historia de Usuario 12:**

**Título:** Sistema libre de errores

**Descripción:** Como **usuario**, quiero que el sistema funcione correctamente sin errores para que mi experiencia sea fluida y confiable.

**Criterios de Aceptación:**

- El sistema debe ser probado exhaustivamente y libre de errores críticos.

## Historias Técnicas

### Historia Técnica 1:

**Título:** Implementación del registro de productos en el inventario

**Descripción:** Como desarrollador, debo crear los **endpoints RESTful** para registrar productos en el inventario y asegurarme de que los datos se almacenen correctamente en la base de datos.

**Criterios de Aceptación:**

- Crear un endpoint en Django para manejar el registro de productos.
- Validar que los productos se guarden correctamente en la base de datos PostgreSQL.

### Historia Técnica 2:

**Título:** Implementación de la autenticación de usuarios con JWT

**Descripción:** Como desarrollador, debo implementar un sistema de autenticación utilizando **tokens JWT** para garantizar que los usuarios puedan iniciar sesión de manera segura y acceder al sistema.

**Criterios de Aceptación:**

- Crear endpoints para el login y registro de usuarios utilizando Django REST Framework.
- Los tokens JWT deben ser generados para los usuarios al iniciar sesión.

### Historia Técnica 3:

**Título:** Implementación del CRUD de productos

**Descripción:** Como desarrollador, debo implementar las funcionalidades para crear, leer, actualizar y eliminar productos en el sistema, asegurando que las operaciones afecten correctamente a la base de datos.

**Criterios de Aceptación:**

- Crear endpoints RESTful para realizar las operaciones CRUD de productos.
- Validar que los datos de los productos sean correctos antes de ser almacenados o modificados en la base de datos.

### Historia Técnica 4:

**Título:** Implementación de la funcionalidad de registro de ventas

**Descripción:** Como desarrollador, debo implementar un sistema para registrar las ventas realizadas, actualizar el inventario y registrar el total de la venta.

**Criterios de Aceptación:**

- Crear un endpoint para registrar las ventas.

- Actualizar automáticamente el stock de productos después de cada venta.

#### **Historia Técnica 5:**

**Título:** Implementación del modelo de predicción de demanda utilizando Machine Learning

**Descripción:** Como **desarrollador**, debo entrenar un **modelo de predicción de demanda** utilizando **Machine Learning**, como la regresión lineal o redes neuronales, y exponerlo a través de una API en Django.

#### **Criterios de Aceptación:**

- El modelo de predicción debe ser entrenado con los datos históricos de ventas.
- Debe generar predicciones de demanda para cada producto basado en los patrones de ventas pasados.
- El modelo debe ser accesible a través de un endpoint API REST.
- Los resultados deben mostrarse en la interfaz de usuario (en React).

#### **Historia Técnica 6:**

**Título:** Implementación de la generación de reportes

**Descripción:** Como desarrollador, debo crear una funcionalidad que permita generar reportes de ventas por diferentes criterios y exportarlos a formatos como CSV o PDF.

#### **Criterios de Aceptación:**

- Crear un endpoint para generar reportes de ventas.
- Implementar la exportación a CSV y PDF.

#### **Historia Técnica 7:**

**Título:** Implementación de la actualización de precios

**Descripción:** Como desarrollador, debo crear la funcionalidad que permita actualizar los precios de los productos en la base de datos y reflejar esos cambios en el sistema.

#### **Criterios de Aceptación:**

- Crear un endpoint para actualizar precios.
- Validar que los cambios de precio se reflejen correctamente en las transacciones.

#### **Historia Técnica 8:**

**Título:** Implementación del CRUD de proveedores

**Descripción:** Como desarrollador, debo crear las funcionalidades para registrar, editar y eliminar proveedores en el sistema.

**Criterios de Aceptación:**

- Crear un endpoint para realizar operaciones CRUD sobre los proveedores.
- Validar que los datos de los proveedores sean correctos y se reflejen en la base de datos.

#### **Historia Técnica 9:**

**Título:** Implementación de la actualización de stock

**Descripción:** Como desarrollador, debo implementar un sistema que actualice el stock de productos en tiempo real cada vez que se registre una venta o ingreso de inventario.

**Criterios de Aceptación:**

- Crear lógica para actualizar el stock después de cada transacción.
- Asegurar que el stock siempre esté disponible y actualizado en tiempo real.

#### **Historia Técnica 10:**

**Título:** Implementación del sistema de caja

**Descripción:** Como desarrollador, debo crear la funcionalidad para registrar ingresos y egresos de dinero en la caja y actualizar correctamente las transacciones en la base de datos.

**Criterios de Aceptación:**

- Crear endpoints para registrar operaciones de caja (ingresos/egresos).
- Verificar que las transacciones se registren correctamente en la base de datos.

#### **Historia Técnica 11:**

**Título:** Implementación del registro de transacciones

**Descripción:** Como desarrollador, debo crear un sistema que registre todas las transacciones realizadas en el sistema, asegurando que los detalles se almacenen correctamente en la base de datos.

**Criterios de Aceptación:**

- Crear un endpoint para registrar las transacciones de ventas.
- Asegurar que los datos de las transacciones se guarden correctamente en la base de datos.

#### **Historia Técnica 12:**

**Título:** Pruebas y corrección de errores

**Descripción:** Como desarrollador, debo realizar pruebas exhaustivas del sistema para asegurarme de que todas las funcionalidades estén operativas y sin errores.

**Criterios de Aceptación:**

- Ejecutar pruebas unitarias y de integración para identificar posibles errores.
- Corregir errores encontrados y validar que las correcciones no afecten otras funcionalidades.