**ESCUELA SUPERIOR POLITECNICA DE CHIMBORAZO**

**FACULTAD DE INFORMATICA Y ELECTRONICA**

**ESCUELA DE SOFTWARE**



**NOMBRE**:

SAAID GUALA

**CODIGO:**

6873

**TEMA**:

DEFINICIÓN DE TECNOLOGÍAS Y PRODUCTO BACKLOG

**FECHA**:

18/11/2024

**MATERIA**:

APLICACIONES INFORMATICAS II

**DOCENTE:**

ING. JULIO SANTILLAN

# Definición de tecnologías

|  |  |  |
| --- | --- | --- |
|  | **Herramienta** | **Lenguaje** |
| **Entorno de desarrollo** | Visual Studio Code | - |
| **Lenguaje de programación** | - | Python |
| **Framework Frontend** | Angular | Typescript |
| **Framework Backend** | Django | Python |
| **Base de datos** | PostgreSQL | SQL, relacional |

* Angular con TypeScript para desarrollar la interfaz de usuario.
* Django para gestionar la lógica del negocio y exponer los datos a través de una API REST.
* Predicción de demanda estarán implementados en Python y expuestos mediante el backend.
* PostgreSQL para almacenar los datos de productos, inventario, ventas, usuarios y resultados de predicciones.

**PRODUCT BACKLOG 12 SPRINTS**

**Sprint 1: Configuración Inicial**

|  |  |
| --- | --- |
| **Título** | Configuración del entorno y autenticación básica. |
| **Capa** | Capa de Aplicación (Backend) y Capa de Persistencia (Base de Datos). |
| **Descripción** | Configurar el entorno de desarrollo, base de datos, y la autenticación de usuarios. |
| **Criterios de**  **Aceptación** | - Usuarios pueden autenticarse con usuario y contraseña. |
|  | - Roles definidos (administrador, vendedor). |
| **Tareas** | - Configurar Django, Angular y PostgreSQL. |
|  | - Implementar autenticación básica en Django. |
|  | - Crear base de datos para usuarios y roles. |
| **Pruebas** | - Pruebas de autenticación y conexiones con la base de datos. |
| **Esfuerzo** | 12 horas. |
| **Prioridad** | Alta. |

# Sprint 2: Registro de Productos

|  |  |
| --- | --- |
| **Título** | **Diseño de interfaces en figma** |
| **Capa** | Capa de Presentación (Frontend), Capa de Aplicación y Persistencia. |
| **Descripción** | Creación del diseño visual de la interfaz de inicio de sesión en Figma. |
| **Criterios de**  **Aceptación** | Validación y aprobación visual del personal de Abastos el Ahorro |
|  | Crear diseño visual |
| **Tareas** |  |
|  | - Escoger los componentes que se van a utilizar |
| **Pruebas** | - Validación visual de los componentes que se van a utilizar en las interfaces |
| **Esfuerzo** | 10 horas. |
| **Prioridad** | Alta. |

**Sprint 3: Actualización Automática del Inventario**

|  |  |
| --- | --- |
| **Título** | **Actualización automática del inventario tras operaciones.** |
| **Capa** | Capa de Aplicación y Persistencia. |
| **Descripción** | Asegurar que el inventario se actualice tras ventas o ingresos de productos. |
| **Criterios de**  **Aceptación** | - Inventario actualizado en tiempo real. |
|  | - Validación de stock negativo. |
| **Tareas** | - Implementar lógica de actualización en Django. |
|  | - Crear endpoint para actualización automática de inventario. |
|  | - Configurar base de datos para reflejar cambios. |
| **Pruebas** | - Validación de lógica en ventas e ingresos. |
| **Esfuerzo** | 14 horas. |
| **Prioridad** | Alta. |

**Sprint 4: Registro de Ventas**

|  |  |
| --- | --- |
| **Título** | **Módulo de registro de ventas.** |
| **Capa** | Capa de Presentación, Aplicación y Persistencia. |
| **Descripción** | Implementar funcionalidad para registrar ventas con detalles importantes. |
| **Criterios de**  **Aceptación** | - Ventas guardadas correctamente en la base de datos. |
|  | - Historial visible desde la interfaz. |
| **Tareas** | - Crear formulario en Angular para ventas. |
|  | - Crear endpoint REST para guardar ventas. |
| **Pruebas** | - Verificación de cálculos y datos almacenados. |
| **Esfuerzo** | 12 horas. |
| **Prioridad** | Alta. |

**Sprint 5: Búsqueda y Filtros**

|  |  |
| --- | --- |
| **Título** | **Implementación de filtros para búsqueda de productos.** |
| **Capa** | Capa de Presentación y Aplicación. |
| **Descripción** | Implementar búsqueda avanzada por nombre, categoría y precio. |
| **Criterios de Aceptación** | - Resultados rápidos y relevantes según los filtros aplicados. |
| **Tareas** | - Crear componente de búsqueda en Angular. |
|  | - Configurar consultas optimizadas en Django. |
| **Pruebas** | - Validación de rendimiento en búsquedas. |
| **Esfuerzo** | 10 horas. |
| **Prioridad** | Alta. |

**Sprint 6: Predicción de Demanda**

|  |  |
| --- | --- |
| **Título** | **Implementación de modelo de Machine Learning para predicciones.** |
| **Capa** | Capa de Servicios (Machine Learning). |
| **Descripción** | Crear un modelo para predecir demanda basado en datos históricos de ventas. |
| **Criterios de**  **Aceptación** | - Modelo funcional con datos históricos entrenados. |
|  | - Endpoint REST para realizar consultas. |
| **Tareas** | - Entrenar modelo con Python y Scikit-learn. |
|  | - Conectar modelo a Django. |
| **Pruebas** | - Validación de predicciones con datos de prueba. |
| **Esfuerzo** | 20 horas. |
| **Prioridad** | Alta. |

**Sprint 7: Visualización de Predicciones**

|  |  |
| --- | --- |
| **Título** | **Mostrar predicciones en gráficos y reportes.** |
| **Capa** | Capa de Presentación. |
| **Descripción** | Crear gráficos interactivos para visualizar predicciones de demanda. |
| **Criterios de Aceptación** | - Gráficos actualizados según nuevas predicciones. |
| **Tareas** | - Crear componente de gráficos en Angular. |
|  | - Conectar gráficos al endpoint de predicciones. |
| **Pruebas** | - Validación de datos mostrados. |
| **Esfuerzo** | 15 horas. |
| **Prioridad** | Media. |

**Sprint 8: Generación de Reportes**

|  |  |
| --- | --- |
| **Título** | **Generación de reportes descargables.** |
| **Capa** | Capa de Presentación, Aplicación y Persistencia. |
| **Descripción** | Crear reportes de inventario y ventas con datos relevantes y exportables. |
| **Criterios de**  **Aceptación** | - Reportes en formato PDF descargables. |
|  | - Incluyen datos de ventas, inventario y predicciones. |
| **Tareas** | - Diseñar lógica de generación de reportes en Django. |
|  | - Conectar frontend para exportar reportes. |
| **Pruebas** | - Verificación de datos en los reportes generados. |
| **Esfuerzo** | 14 horas. |
| **Prioridad** | Media. |

**Sprint 9: Gestión de Proveedores**

|  |  |
| --- | --- |
| **Título** | **Registro y gestión de proveedores.** |
| **Capa** | Capa de Presentación, Aplicación y Persistencia. |
| **Descripción** | Crear funcionalidad para gestionar información de proveedores. |
| **Criterios de**  **Aceptación** | - Permite registrar, editar y eliminar proveedores. |
|  | - Incluye historial de compras realizadas. |
| **Tareas** | - Crear formulario para registrar proveedores en Angular. |
|  | - Crear API REST en Django para CRUD de proveedores. |
| **Pruebas** | - Validación de correcto registro y manejo de datos. |
| **Esfuerzo** | 12 horas. |
| **Prioridad** | Alta. |

**Sprint 10: Alertas de Stock Bajo**

|  |  |
| --- | --- |
| **Título** | **Notificaciones automáticas de stock bajo.** |
| **Capa** | Capa de Aplicación y Presentación. |
| **Descripción** | Implementar alertas automáticas para productos con stock bajo. |
| **Criterios de**  **Aceptación** | - Notificaciones visibles en la interfaz. |
|  | - Umbrales configurables por producto. |
| **Tareas** | - Diseñar lógica de alertas en Django. |
|  | - Crear notificaciones visuales en Angular. |
| **Pruebas** | - Validación del sistema de alertas para diferentes niveles de inventario. |
| **Esfuerzo** | 10 horas. |
| **Prioridad** | Alta. |

**Sprint 11: Seguridad y Backups**

|  |  |
| --- | --- |
| **Título** | **Medidas de seguridad y copias de respaldo automáticas.** |
| **Capa** | Capa de Aplicación y Persistencia. |
| **Descripción** | Asegurar la protección de datos sensibles y respaldos automáticos. |
| **Criterios de**  **Aceptación** | - Implementación de encriptación para datos sensibles. |
|  | - Copias de respaldo automáticas configuradas en la nube o localmente. |
| **Tareas** | - Configurar seguridad en Django con encriptación. |
|  | - Implementar backups diarios automáticos en PostgreSQL. |
| **Pruebas** | - Verificación de datos respaldados y recuperados. |
| **Esfuerzo** | 12 horas. |
| **Prioridad** | Alta. |

**Sprint 12: Pruebas Finales y Optimización**

|  |  |
| --- | --- |
| **Título** | **Validación del sistema y ajustes finales.** |
| **Capa** | Todas las capas. |
| **Descripción** | Realizar pruebas integrales, resolver errores y preparar documentación final. |
| **Criterios de**  **Aceptación** | - Todas las funcionalidades son estables. |
|  | - Documentación técnica y de usuario completa. |
| **Tareas** | - Pruebas unitarias e integrales para validar el sistema. |
|  | - Resolución de errores detectados. |
|  | - Documentación de uso y manual de instalación. |
| **Pruebas** | - Pruebas de carga y estrés para validar rendimiento. |
| **Esfuerzo** | 15 horas. |
| **Prioridad** | Alta. |