



Tecnológico de Monterrey

Plan de Pruebas

Planeación de sistemas de software (Gpo 104)

Guillermo Montemayor Marroquin	A01722402
Mauricio Noriega Chapa	A01722543
David Mireles Gutierrez	A00836010
Marcelo Cárdenas Leal	A00836049
Paulina Mijares Blanco	A01383803
Danny Xiao Yong Wu	A01286362

12 de marzo del 2025

1. Estrategia de Pruebas.....	3
1.1 Ciclo y Tiempos de Prueba.....	3
1.2 Manejo de Resultados y Registro de Defectos.....	4
1.2.1 Flujo de Manejo de Defectos.....	5
1.3 Priorización de Defectos.....	5
2. Alcance y Objetivos de las Pruebas.....	6
2.1 Alcance.....	6
2.2 Objetivos de las Pruebas.....	7
3. Características a ser probadas y a no ser probadas.....	8
3.1 Características a ser probadas.....	8
3.2 Características a no ser probadas.....	9
4. Infraestructura del Entorno de Pruebas.....	10
4.1 Justificación del uso de un solo entorno.....	10
4.2 Infraestructura del Entorno.....	10
4.3 Estrategia para minimizar riesgos en un único entorno.....	11
4.4 Consideraciones Finales.....	11
5. Capacitaciones requeridas para pruebas.....	12
6. Evaluación de Riesgos en las Pruebas.....	12
6.1 Estrategias Generales de Mitigación.....	13
6.2 Identificación y evaluación de riesgos.....	13
6.2.1 Lista de riesgos:.....	13
7. IA en nuestra solución.....	16
8. Anexos.....	16

1. Estrategia de Pruebas

El objetivo del plan de pruebas es garantizar que todos los requerimientos funcionales y no funcionales sean validados a través de pruebas estructuradas. Para ello, se implementarán distintos tipos de pruebas según el objetivo de validación:

Tipo de Prueba	Descripción	Herramientas
Pruebas Unitarias	Validan la funcionalidad de componentes individuales.	Selenium
Pruebas de Integración	Verifican la comunicación entre módulos.	Postman
Pruebas de Seguridad	Validan vulnerabilidades en accesos y datos.	SAP Security Tools
Pruebas de Rendimiento	Evalúan la respuesta del sistema bajo carga	Nodemon
Pruebas de Aceptación	Validación final con criterios de negocio.	Jira Test Cases

Tabla 1. Tipo de pruebas

Cada una de estas pruebas se documentará siguiendo el formato definido en la sección de Manejo de Resultados y Registro de Defectos, asegurando una correcta trazabilidad de los errores.

Se probarán los requerimientos funcionales a lo máximo 5 días después de que se consideren finalizados por el desarrollador que los creó. En caso de encontrar un error con priorización crítica se espera que los desarrolladores corrijan en un lapso de 24 horas; los errores de baja priorización se agendaron al siguiente sprint. Los errores se priorizarán de acuerdo a su importancia descrita en la sección de requerimientos funcionales. Se registrarán las pruebas en un documento Google Docs y se llevará a cabo su seguimiento en Jira.

1.1 Ciclo y Tiempos de Prueba

- Las pruebas se ejecutarán en un máximo de 5 días después de que el desarrollador haya finalizado el requerimiento correspondiente.
- Se recomienda realizar la prueba el mismo día o al día siguiente para una validación ágil.
- Todas las pruebas se documentan en Google Docs utilizando el formato anterior.

1.2 Manejo de Resultados y Registro de Defectos

Para asegurar trazabilidad de las pruebas y la rápida resolución de errores, todos los resultados serán documentos siguiendo el siguiente formato en Jira y Google Docs. Los defectos encontrados durante las pruebas serán registrados con información detallada, incluyendo su impacto, prioridad y pasos a seguir.

Para ello, se utilizará el siguiente formato para documentar las pruebas:

ID de la prueba	Requerimiento asociado	
Descripción de la prueba		
Criterios de aceptación		
Datos de entrada necesarios para ejecutar la prueba, y su tipo (String, Int, Float, etc.)		
Procedimiento para llevar a cabo la prueba (Qué programa / función / script se utilizará. Ej. Postman para APIs)		
Resultado Esperado	Resultado Obtenido	Estado de prueba
Notas adicionales		

Descripción de cada campo del formato de prueba:

Campo	Descripción
ID de la prueba	Código único de la prueba (Ej: PR01)
Requerimiento Asociado	ID del requerimiento funcional o no funcional que valida la prueba.
Descripción de la Prueba	Explicación detallada del objetivo de la prueba.
Criterios de aceptación	Condiciones que deben cumplirse para que la prueba sea exitosa.
Datos de entrada necesarios para ejecutar la prueba	Valores requeridos para ejecutar la prueba.

Procedimiento para llevar a cabo la prueba	Pasos para ejecutar la prueba.
Resultado Esperado	Lo que se espera obtener si la funcionalidad es correcta.
Resultado Obtenido	Lo que realmente ocurre durante la prueba.
Estado de prueba	El estado de la prueba Éxito / Fallo / Pendiente de revisión
Notas adicionales	Capturas de pantalla, errores del sistema o logs relevantes.

Tabla 2. Descripción del formato para documentar pruebas

Las herramientas donde se registrarán estas pruebas serán en Google Docs documentados manualmente en un Manual de pruebas y su seguimiento y gestión de defectos será en Jira.

1.2.1 Flujo de Manejo de Defectos

Cuando una prueba falle, se generará un tarjeta de Jira con el siguiente flujo de trabajo:

1. Reporte del Defecto
 - Se documenta el error en Jira con los campos detallados anteriormente.
 - Se asigna al desarrollador responsable.
2. Análisis del Defecto
 - El desarrollador revisa el error, repuce el fallo y analiza su causa.
 - Se actualiza el Estado en Jira (En Revisión)
3. Corrección del Defecto
 - Se impronta la solución y se sube a GitHub.
 - Se actualiza el estado en Jira (Resuelto)
4. Reejecución de la Prueba
 - El equipo de pruebas válida la corrección.
 - Si la prueba es exitosa, se marca como Cerrado en Jira.
 - Si la prueba persiste, el defecto vuelve a En revisión.

1.3 Priorización de Defectos

Los defectos serán clasificados según su impacto y prioridad de resolución. La prioridad de efectos se asignará una vez analizando el defecto y esta calificación puede ser de Crítico siendo un defecto con muy alto impacto y bajo siendo un defecto con bajo impacto.

Prioridad del defecto	Impacto	Tiempo de Corrección
Crítico	Bloqueo funciones esenciales del sistema.	Máximo 24 horas
Alto	Impacta funcionalidades clave, pero el sistema puede seguir funcionando.	Máximo 48 horas
Medio	Afecta funcionalidades pero sin bloquear funciones principales.	5 días hábiles
Bajo	Errores visuales.	Se agenda en siguientes sprints

Tabla 3. Tiempo de corrección por prioridad

2. Alcance y Objetivos de las Pruebas

Las pruebas del sistema WuSolutions tiene como objetivo asegurar que la solución cumpla con los requerimientos funcionales, no funcionales y de despliegue. Se implementarán diferentes tipos de pruebas para validar la solidez, seguridad y rendimiento del sistema antes de su entrega final.

2.1 Alcance

1. Requerimientos Funcionales:

- Se validará que cada funcionalidad cumpla con los criterios definidos en la especificación de requerimientos.
- Se ejecutarán pruebas unitarias, de integración y de aceptación según corresponda.

Tipo de Prueba	Objetivo
Pruebas Unitarias	Validar el correcto funcionamiento de funciones y métodos individuales
Pruebas de Integración	Comprobar que los módulos del sistema se comunican correctamente entre sí.
Pruebas de Aceptación	Asegurar que el sistema cumpla con los requisitos establecidos.

Tabla 4. Tipo de pruebas para requerimiento funcionales

2. Requerimientos No Funcionales:

- Se realizarán pruebas para garantizar la integridad y consistencia de los datos.

- Validar la seguridad de la información, incluyendo autenticación y encriptación de datos en la base de datos.
- Evaluar la usabilidad de la interfaz de usuario para mejorar la experiencia de usuario.

Aspectos Evaluado	Objetivo
Seguridad de Datos	Verificar la encriptación y acceso seguro a información sensible.
Usabilidad	Evaluar la facilidad de uso e intuitividad de la interfaz gráfica.
Disponibilidad	Probar la estabilidad del sistema ante caídas inesperadas y su capacidad de recuperación.

Tabla 5. Tipo de pruebas para requerimiento no funcionales

3. Pruebas de Despliegue y Compilación:

- Se verificará que el código pueda compilarse y desplegarse correctamente en el entorno de producción sin errores.
- Se evaluará la correcta configuración del entorno de ejecución.

Fase	Objetivo
Compilación	Verificar que no haya errores al generar el código.
Despliegue	Asegurar que la aplicación se ejecuta correctamente en SAP Cloud Foundry.
Configuración	Revisar que las conexiones con SAP HANA estén correctamente establecidas.

Tabla 6. Fases para el despliegue y compilación

2.2 Objetivos de las Pruebas

El objetivo principal de las pruebas de WuSolutions son validar que el sistema cumpla con los requerimientos definidos. Poder detectar defectos antes de la implementación final, minimizando riesgos. Asegurar que la integración sea correcta entre los diferentes componentes del sistema. Garantizar que la solución sea segura y proteja los datos almacenados. Nuestro sistema se espera que sea funcional, seguro y estable.

3. Características a ser probadas y a no ser probadas

Esta sección detalla los módulos y funcionalidades que serán evaluados durante el proceso de pruebas, así como aquellos que quedan fuera del alcance de la validación.

3.1 Características a ser probadas

Se evaluaron los siguientes aspectos del sistema WuSolutions para asegurar su correcto funcionamiento:

- Se validaron los requerimientos funcionales RF01 a RF11.
- Se validaron los requerimientos no funcionales de Producto-Usabilidad, Producto-Dependabilidad, Producto-Seguridad.

Dentro de las características generales a ser probadas se encuentran las siguientes:

Módulo	Roles aplicables	Descripción
Autenticación de usuarios	Back-end, Encargado de Pruebas	Validación de credenciales y acceso a pantallas correspondientes.
Navegación	Front-end, Encargado de Pruebas	Verificación de rutas según el tipo de usuario y su acceso a las distintas secciones.
Manejo de errores	Front-end, Encargado de Pruebas	Control de excepciones y verificar la existencia de los mensajes de error en los campos correspondientes.
Almacenamiento de datos	Back-end, Arquitecto, Encargado de Pruebas	Verificar que los datos están siendo almacenados de forma correspondiente en la base de datos, según las entradas en la aplicación web.
Funciones CRUD	Back-end, Front-end, Encargado de Pruebas	Operaciones sobre los datos almacenados según el tipo de rol del usuario.
Funcionalidades de búsqueda	Back-end, Front-end, Encargado de Pruebas	Verificación de la eficacia de los filtros y acciones de búsqueda, comprobar que sean correctas las salidas.

Tabla 7. Características generales a ser probadas

3.2 Características a no ser probadas

Algunas características no serán evaluadas en esta fase debido a las restricciones de alcance, dependencia de sistemas externos o porque su validación se realizará en etapas posteriores.

Características	Justificación y Métodos Alternativos de Validación
Integración con sistemas externos	No se validará la conexión con sistemas externos como APIs de terceros o servicios en la nube fuera del entorno de pruebas controlado. La comunicación con SAP BTP y SAP Cloud Foundry será validada en etapas posteriores.
Compatibilidad con dispositivos y navegadores distintos a los planeados.	No se realizarán pruebas en múltiples navegadores o tamaños de pantalla distintos al navegador Chrome en resolución Full HD (1920 x 1080).
Validación del Documento de Especificación de Requerimientos (SRS)	No se validará la redacción del documento SRS, calidad, y/o sus cualidades bajo el formato de pruebas establecido.
Manual de usuario	No se realizarán pruebas sobre la claridad, calidad, y/o cualidades del manual bajo el formato de pruebas establecido. Sin embargo se revisará que las instrucciones sean claras y funcionales para los usuarios finales.
Documentación del código fuente	No se aplicarán herramientas automatizadas para validar la estructura o comentarios en el código, ni se harán pruebas sobre la documentación bajo el formato de pruebas establecido.
Límite superior de escalabilidad del producto	No se probarán escenarios de estrés que midan la capacidad del sistema para soportar grandes volúmenes de datos o usuarios simultáneos.
Velocidad y eficiencia de la CI/CD pipeline	No se evaluará el rendimiento de la integración y despliegue continuo más allá de su correcto funcionamiento en un entorno de pruebas.
Cantidad y horario de commits en GitHub	No se analizarán métricas relacionadas con los commits, pull o merges dentro del repositorio.

Tabla 8. Características generales a no ser probadas

4. Infraestructura del Entorno de Pruebas

Para el desarrollo, pruebas y despliegue del sistema, se utilizará un único entorno para desarrollo y producción. Esta decisión se ha tomado considerando la optimización de recursos y simplicidad en la gestión.

4.1 Justificación del uso de un solo entorno

El equipo ha decidido utilizar un único entorno para desarrollo y producción debido a:

- Optimización de recursos: Se evita la necesidad de mantener múltiples servidores, reduciendo costos y esfuerzo administrativos.
- Menor complejidad: No será necesario gestionar la sincronización entre los entornos ni migrar cambios entre ellos.
- Desarrollo y pruebas en un ambiente realista: Las pruebas se realizan directamente en el sistema en uso, asegurando coherencia entre el desarrollo y la operación final.

4.2 Infraestructura del Entorno

DEV tendrá los siguientes requerimientos mínimos, ya que puede existir variación de ambientes en los dispositivos de los desarrolladores.

Recurso	Especificación de requerimientos mínimos
CPU	4 procesadores lógicos
Memoria RAM	8GB
Almacenamiento	32GB
Base de datos	SAP HANA con encriptación y respaldo
Plataforma de despliegue	SAP Cloud Foundry
Repositorio de código	Github
Monitoreo	Logs en Github en tiempo real

Tabla 9. Infraestructura del entorno

TEST y PROD tendrán los siguientes requerimientos mínimos, se utiliza el mismo ambiente en TEST y PROD ya que buscamos minimizar las diferencias para el usuario final.

Recurso	Especificación de requerimientos mínimos
CPU	2 procesadores lógicos
Memoria RAM	4GB
Almacenamiento	32GB
Base de datos	SAP HANA con encriptación y respaldo
Plataforma de despliegue	SAP Cloud Foundry
Repositorio de código	Github
Monitoreo	Logs en Github en tiempo real

Tabla 10. Infraestructura del entorno

4.3 Estrategia para minimizar riesgos en un único entorno

Dado a que solo usaremos un entorno, se implementará las siguientes medidas para evitar errores en producción:

- Uso de ramas en Github para controlar cambios: Solo cambios revisados y aprobados se integrarán en la rama principal. Se hará una rama pre-producción para probar componentes nuevos.
- Backups automáticos: Se harán respaldos de la base de datos diarios.
- Separación de datos en el mismo entorno: Se establecen usuarios y permisos diferentes. Los desarrolladores tendrán acceso a la herramienta de prueba y debugging mientras que el usuario final tendrá acceso restringido y solo usara funcionales pre-aceptadas.
- Monitoreo y alertas en tiempo real: Se hará integración como SAP Cloud Logging para detectar errores lo antes posible.

4.4 Consideraciones Finales

El uso de un solo entorno reduce costos y facilita el desarrollo, pero también incrementa riesgos. Para esto garantizamos la estabilidad con lo antes mencionado. Teniendo disciplina en la gestión del código, hacer backups diarios, monitorear constantemente y tener respaldo para mitigar impactos que afecten producción.

5. Capacitaciones requeridas para pruebas

Roles	Materiales de apoyo
Project Manager	Documentación de Jira - Click para acceso al Jira
Desarrollador front end	Documentación de Material UI - Click para ir a la documentación de Material UI Documentación React. - Click para ir a la documentación de React Documentación de SAP UI5 - Click para ir a la documentación de SAP UI5
Desarrolladores de back end	Documentación React. - Click para ir a la documentación de React Curso sobre SAP Hana. - Click para ir al curso de SAP HANA SAP Cloud Foundry. - Click para ir al curso de SAP Cloud Foundry SAP AI. - Click para ir al curso de SAP Analytics Cloud - Click para ir al curso de Hands-on Tutorial: Machine Learning with SAP HANA Cloud - Click para ir al curso de Predictive AI with SAP AI Core - SAP Business AI Predictive Replenishment - Documentación
Arquitectos de software	Demo proporcionada por SAP sobre conexión a la base de datos. - Click para ir al Demo proporcionado por SAP
Encargados de pruebas	Documentación de Postman. - Click para ir a la documentación de Postman Documentación de React. - Click para ir a la documentación de React Uso de formato de pruebas.
Todos	Discovering SAP Business Technology Platform: - Click para ir al curso de SAP Business Technology Platform Foundation (SAP BTP) Plática sobre innovación de parte de SAP. Plática sobre design thinking de parte de SAP.

Tabla 11. Capacitaciones necesarias por roles

6. Evaluación de Riesgos en las Pruebas

Para reducir el impacto de los riesgos identificados en el desarrollo del proyecto WuSolutions, se han definido estrategias para mitigar cada escenario. Estas estrategias buscan minimizar la probabilidad de ocurrencia de los riesgos y reducir su impacto en el proyecto.

6.1 Estrategias Generales de Mitigación

1. Monitoreo continuo: Seguimiento de los riesgos en reuniones semanales, ajustando estrategias según el estado del proyecto.
2. Asignación de responsabilidades: Cada riesgo tiene un responsable designado para su gestión y mitigación.
3. Documentación y seguimiento: Uso de Jira para registrar los riesgos y establecer medidas de mitigación.
4. Capacitación: En caso de un riesgo técnico, el equipo realizará sesiones de aprendizaje para mitigar posibles riesgos futuros.

6.2 Identificación y evaluación de riesgos

Los riesgos se listarán bajo el siguiente formato:

idRiesgo	Título	Impacto
Descripción		
Responsable		
Posibles efectos		
Estrategia de mitigación		

6.2.1 Lista de riesgos:

R01	Tiempo limitado para el desarrollo del proyecto.	Alto
Subestimamos el tiempo requerido para el desarrollo del proyecto.		
Project Manager: Marcelo Cárdenas.		
<ol style="list-style-type: none">1. Entregas tardías.2. Proyecto incompleto.3. Retraso en otras áreas.		
El tiempo para completar el proyecto es restringido. Se aplicará una estrategia de planificación ágil con sprints de 1 semana de duración, asignando tareas específicas en Jira con fechas límite claras y realizando revisiones semanales para asegurar el cumplimiento de las tareas.		

R02	Falta de experiencia en el uso de herramientas SAP.	Medio
Se podrían presentar problemas debido a la falta de conocimiento sobre las herramientas SAP.		
Back End: Guillermo Montemayor		
<ol style="list-style-type: none"> 1. Entregas tardías. 2. Proyecto incompleto. 3. Retraso en otras áreas. 		
Se definirán cursos clave y se mantendrá contacto con el equipo técnico de SAP para seguir tutoriales y capacitarse en las herramientas necesarias.		

R03	Falta de experiencia en lenguajes de programación necesarios.	Medio
Se podrían presentar problemas debido a la inexperiencia en los lenguajes de programación utilizados para este proyecto. Implementación de pruebas tempranas en entornos controlados.		
Front end: Danny Wu y Paulina Mijares Back end: Guillermo Montemayor		
<ol style="list-style-type: none"> 1. Entregas tardías. 2. Proyecto incompleto. 3. Retraso en otras áreas. 4. Errores en los componentes creados. 5. Resultados incorrectos. 		
Utilizar los recursos disponibles en línea, como tutoriales y documentación. Asignar las tareas según las fortalezas de cada miembro.		

R04	Dificultades en la integración de diferentes componentes del proyecto.	Alto
Se podrían presentar problemas debido a la inexperiencia en el uso de los componentes.		
Front end: Danny Wu y Paulina Mijares. Back end: Guillermo Montemayor.		
<ol style="list-style-type: none"> 1. Entregas tardías. 		

2. Resultados erróneos. 3. Dificultad en integración.
Integrar las distintas partes del sistema puede generar problemas. Se realizarán pruebas de integración desde etapas tempranas y se documentan detalladamente las interfaces y dependencias del proyecto.

R05	Falta de comunicación y coordinación dentro del equipo.	Alto
Falta de comunicación sobre los requerimientos, progreso, necesidades de los componentes, y flujo de información.		
Project Manager: Marcelo Cárdenas.		
1. Integraciones incorrectas. 2. Resultados erróneos. 3. Retrasos en el progreso. 4. Retrasos en otras áreas.		
Se usará Jira para la gestión del proyecto, se establecerán roles claros, se harán reuniones semanales de seguimiento, se fomentará una comunicación frecuente para así eliminar los problemas de comunicación y organización interna. Se usará Discord y Whatsapp para comunicación diaria.		

R06	Los entregables no cumplen con los estándares de calidad previamente definidos.	Alto
Los estándares de calidad no son respetados durante el desarrollo, o son incumplidos por alguna razón.		
Encargado de pruebas: Mauricio Noriega		
1. Resultados erróneos. 2. Requerimientos faltantes. 3. Tener que regresar, y modificar código por incumplimiento.		
Existe el riesgo de que los entregables no cumplan con los estándares esperados. Se definirán desde el inicio del proyecto y se realizarán revisiones periódicas para asegurar la calidad.		

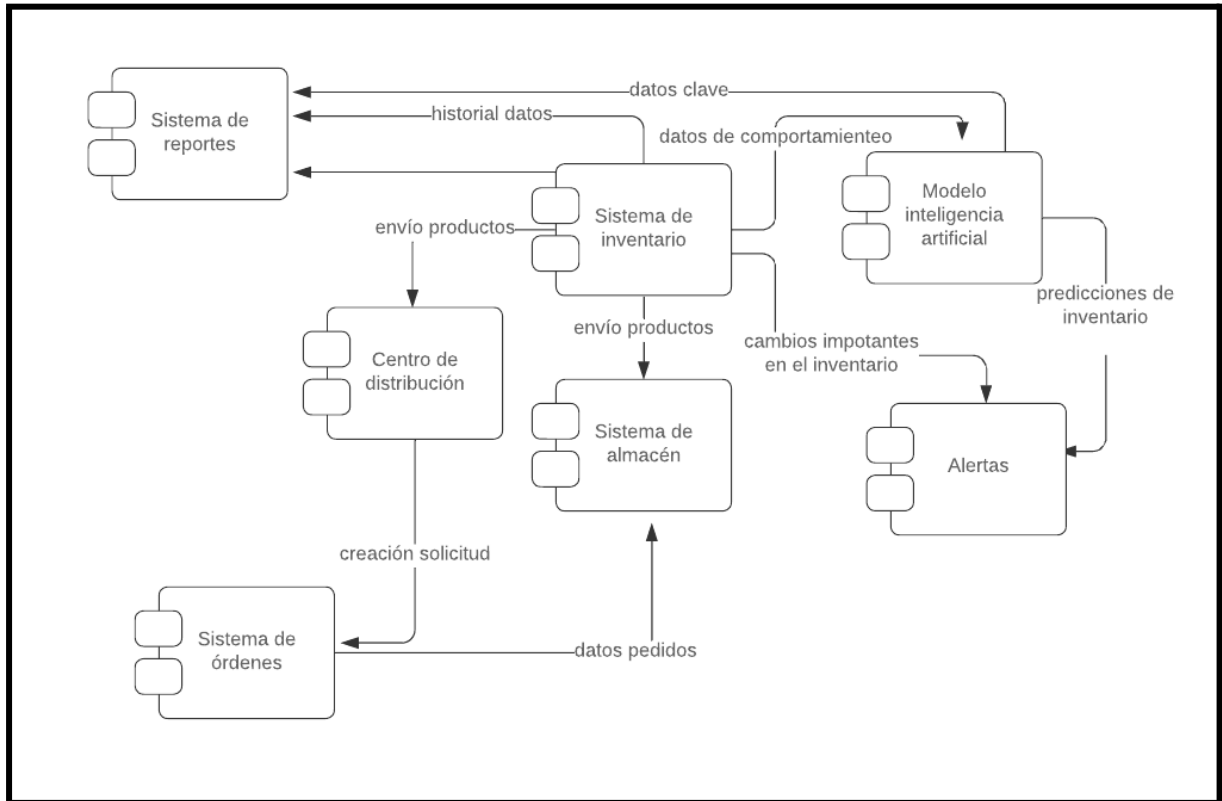
7. IA en nuestra solución

Primero se necesita alinear los datos de inventario, ventas y compras por día para identificar patrones y realizar predicciones de reaprovisionamiento de manera más precisa, al igual que las fechas de pedidos y entregas para calcular la tardanza (lead time) de las entregas.

Después se necesitará cargar los datos, y ligarlos con SAP AI para entrenar el modelo predictivo. Se necesitará especificar la estrategia de reaprovisionamiento; por ejemplo, cantidades mínimas de pedidos, porcentaje bajo el stock máximo, días de anticipación, etc. Además, se necesita probar el modelo para simular la compra venta de productos. Finalmente se despliega el modelo, y se conecta con el resto del software.

8. Anexos

Diagrama de Componentes:



Github: <https://github.com/gmonmarr/WuSAP>

Acceso a jira

Fue enviado vía correo electrónico el acceso al jira.

Link al Jira

