Guía Informe final de proyecto de Desarrollo de SIAs

# Introducción

El sistema propuesto busca optimizar la gestión de inventario para una tienda minorista, brindando a los usuarios un control en tiempo real de la disponibilidad de productos, reduciendo pérdidas y mejorando la satisfacción tanto de los vendedores como de los clientes. Con esta solución se espera generar un impacto positivo en la eficiencia y en el crecimiento de la tienda minorista.

# Etapa 1: Visión y Alcance del Proyecto

1. Visión

"Ser el sistema de inventario más rápido y eficiente del mercado, permitiendo a las empresas optimizar sus flujos de trabajo, reducir tiempos y disminuir las pérdidas por obsolescencia".

1.2 Problema y contexto

La tienda actualmente gestiona su inventario de forma manual, lo que genera varios impactos negativos:

* Desconocimiento del stock real, dificultando la planificación y la toma de decisiones.
* Pérdida de ventas debido a la falta de productos disponibles.
* Mayor tiempo y esfuerzo en la gestión y actualización de registros de inventario.
* Mayor riesgo de errores operativos y desajustes en el stock.

Estos problemas evidencian la necesidad de implementar un sistema automatizado de gestión de inventarios que permita mejorar la eficiencia, reducir pérdidas y garantizar la disponibilidad de productos para los clientes.

1.3 Usuarios y stakeholders

* **Dueño del almacén**

Dolencias: desconocimiento del stock real, incrementa pérdidas de ventas debido a la falta de productos disponibles y genera un mayor consumo de tiempo y actualización del inventario

Objetivo: Tener información clara y en tiempo real del inventario, optimizar la organización de las compras, Reducir pérdidas, maximizar la rentabilidad y anticiparse a los quiebres de stock para asegurar la continuidad en las ventas.

* **Vendedor de tienda**

Dolencia: Dificultad para saber que productos están disponibles, pérdida de tiempo buscando o consultando stock.

Objetivo: verificar rápidamente el inventario, anotar ingresos y salidas sin equivocaciones, disminuyendo el tiempo de gestión.

* **Clientes**

Dolencias: Largas esperas en caja o atención y dificultad para encontrar productos

Objetivo: Encontrar siempre el producto disponible, Obtener una experiencia de compra más eficiente y satisfactoria, comprar de manera rápida y sin complicaciones

* **Proveedor**

Dolencias: Recepción de pedidos desordenados o fuera de tiempo y falta de comunicación clara sobre las necesidades reales de la tienda

Objetivo: Tener pedidos claros y oportunos, mantener una relación estable con el almacén, contar con información sobre las necesidades del almacén, establecer una relación más eficiente y fluida con el negocio.

1.4 Propuesta de valor

La solución propuesta permitirá a la tienda gestionar su inventario de manera efectiva en aspectos como:

* Gestión efectiva del stock en tiempo real.
* Disminución de pérdidas por falta de productos.
* Ahorro de tiempo en la gestión del inventario.
* Aumento en la satisfacción del cliente debido a la disponibilidad de artículos.
* Planificación de compras más eficiente.

Con todo lo anterior se mejorará tanto los procesos internos como los resultados comerciales de la tienda.

# Etapa 2: Organización y Planificación del Proyecto

* 1. Definición de Roles y Responsabilidades.

1. Cliente/usuario final- Yahima Hadfeg, Valentina Duran, Valentina Fuentes, Ginna Cespedes, Maria Quezada:

* **Rol:** Persona que realiza las compras
* **Responsabilidades:**
  + Identificar los requerimientos del sistema.
  + Determinar las metas del negocio.
  + Revisar y aprobar los resultados del proyecto.

1. Jefe de Proyecto/Proyect Manager - Valentina Duran:

* **Rol:** Encargada de la gestión integral del proyecto
* **Responsabilidades:**
  + Organizar las actividades y etapas del proyecto.
  + Dirigir al grupo de trabajo.
  + Supervisar el calendario, los posibles problemas y la calidad del proyecto.
  + Actuar como intermediario entre el cliente y el equipo técnico.

1. Arquitecto del Software/Diseñador - Valentina Fuentes:

* **Rol:** Encargada de las decisiones técnicas de alto nivel
* **Responsabilidades:**
  + Crear el diseño del sistema.
  + Establecer el formato de la base de datos.
  + Elegir las herramientas, modelos de diseño y conexiones.
  + Garantizar la uniformidad técnica en todo el proyecto

1. Equipo de desarrollo (Programadores) - María Quezada:

* **Rol:** Desarrolladora principal del sistema
* **Responsabilidades:**
  + Estructurar el sistema de acuerdo con los planos y especificaciones.
  + Ejecutar las funciones esenciales.
  + Llevar a cabo pruebas unitarias del código.
  + solucionar problemas encontrados a lo largo del desarrollo.

1. Administrador de sistemas/ Infraestructura – Ginna Cespedes:

* **Rol:** Responsable de la infraestructura tecnológica
* **Responsabilidades:**
  + Establecer el entorno local ya sea servidores, base de datos MySQL.
  + Asegurar la red, la protección y la operatividad del sistema.
  + Hacer copias de seguridad y restaurar datos.
  + Supervisar el funcionamiento de los recursos del sistema.

1. Encargado de documentación – Ginna Cespedes:

* **Rol:** Responsable de la documentación del proyecto
* **Responsabilidades:**
  + Asegurar que el equipo tenga documentación actualizada y clara
  + Escribir y estructurar los documentos del sistema.

2.2 Planificación de Tareas (Carta Gantt/ Kanban.).

La planificación del proyecto se representará a través de una carta Gantt, la cual permitirá visualizar de manera clara las actividades, duración y la relación entre ellas. Este recurso facilita la organización del trabajo en equipo, los plazos establecidos y las responsabilidades. De este modo, se garantiza un seguimiento eficiente del progreso y una adecuada gestión del tiempo, contribuyendo al cumplimiento de los objetivos definidos.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Tareas** | **Agosto** | | | | **Septiembre** | | | | **Octubre** | | | | **Noviembre** | | | |
| **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** |
| Realizar reunión inicial |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Identificar necesidades |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Definir funcionalidades principales |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Identificar usuarios y roles |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Crear modelo identidad -relación |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Crear base de datos |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Diseñar interfaces graficas |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Definir arquitectura MVC |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Implementar conexión a MySQL |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Desarrollar módulo de productos |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Desarrollar modulo de movimientos de inventario |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Desarrollo modulo de consultas de stock |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Realizar pruebas unitarias |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Realizar pruebas de integración con MySQL |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Realizar pruebas de seguridad |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Revisar documentación y correcciones |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Entregar y presentar |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

2.3 Herramientas de Organización y Gestión del Proyecto.

Para garantizar una adecuada colaboración entre los miembros del equipo y el cumplimiento de los plazos fijados, se emplearán las siguientes herramientas para ayudar en la gestión del proyecto:

1. **Word compartido**

* ***Finalidad:*** Centralizar la edición y almacenamiento de documentos relacionados con el proyecto.
* **Justificación:** Permite que los miembros del equipo trabajen de manera simultánea en informes, requisitos y documentación, asegurando consistencia, accesibilidad desde distintos dispositivos y resguardo de la información.

1. **GitHub**

* *Finalidad:* Plataforma en la nube para almacenar, gestionar y colaborar en proyectos de código.
* *Justificación:* Facilita el trabajo colaborativo de los programadores mediante control de versiones, lo que permite llevar un historial de cambios y evitar la pérdida de avances.

1. **NetBeans IDE**

* ***Finalidad:*** Entorno de desarrollo integrado para la construcción de la aplicación en Java, incluyendo interfaz gráfica y conexión a la base de datos.
* ***Justificación:*** Su estructura permite organizar de manera clara los paquetes, clases y controladores bajo el modelo MVC, garantizando orden y escalabilidad del software. Además, simplifica la depuración y pruebas iniciales de la aplicación.

1. **MySQL Workbench**

* *Finalidad:* diseño, creación y gestión de la base de datos para el sistema de inventario.
* *Justificación:* facilita la estructuración de tablas, relaciones y pruebas de consultas. Su integración con el entorno de desarrollo asegura que la lógica de datos sea confiable, optimizando el rendimiento del sistema.

2.4 Matriz de riesgos inicial (seguridad, datos, continuidad).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| | ***Riesgo identificado*** | ***Categoría (Seguridad/Datos/Continuidad)*** | ***Probabilidad (Alta/Media/Baja)*** | ***Impacto (Alto/Medio/Bajo)*** | ***Mitigación inicial*** | ***Responsable*** | | --- | --- | --- | --- | --- | --- | | *Fallo en el registro de entradas/ salidas de los productos* | *Datos* | *Media* | *Alto* | Llevar un registro manual o en planilla mientras se arregla el sistema, revisando que las entradas y salidas coincidan*.* | *Programador* | | *Acceso no autorizado al sistema de inventario* | *Seguridad* | *Media* | *Alto* | *Autentificación con usuario/contraseña segura y roles diferenciados* | *Administrador de sistemas* | | *Pérdida de datos por corte de luz o falla técnica* | *Continuidad* | *Bajo* | *Alta* | *Respaldos diarios y UPS para proteger servidores* | *Administrador de sistemas* | | *Manipulación incorrecta de stock por error humano* | *Datos* | *Alta* | *Medio* | *Capacitación a usuarios y registros con logs de cambios* | *Dueño/vendedor* | | *Falla en la conexión a internet que impida usar el sistema (si está en la nube)* | *Continuidad* | *Media* | *Medio* | *Habilitar modo offline temporal o exportar reportes en Excel* | *Jefe de proyecto* | | *Errores en la base de datos (corrupción, mala normalización)* | *Datos* | *Media* | *Alto* | *Revisiones periódicas de integridad, normalización adecuada y respaldos incrementales.* | *Programador* | | *Resistencia al cambio por parte de los usuarios* | *Continuidad* | *Alta* | *Alto* | |  | | --- | |  |   *Plan de gestión del cambio como capacitaciones, apoyo continuo y comunicación clara de beneficios* | *Jefe de proyecto* | | *Uso indebido de credenciales compartidas* | *Seguridad* | *Alta* | *Alto* | *Políticas de credenciales únicas y monitoreo de accesos.* | *Administrador de sistemas* | | *Ataques externos (malware, ransomware)* | *Seguridad* | *Media* | *Alto* | *Copias de seguridad y monitoreo de amenazas.* | *Administrador de sistemas* | |

2.5 Checklist de decisión rápida (producto vs SaaS, nube vs on-premise, etc.).

|  |
| --- |
| 1. ****Producto instalado vs SaaS (Software como servicio)**** **Producto instalado (On-Premise)**  **SaaS (Servicio en la nube)**  **Justificación:**  **El sistema será creado usando Java con NetBeans y MySQL, operando de manera local en la computadora del almacén, sin necesidad de internet o servicios externos.** 2. ****Infraestructura: Nube vs On-Premise**** **On-Premise.**  **Nube (IaaS/PaaS/DBaaS)**  **Justificación:**  Se utilizará un ambiente local (MySQL instalado en la computadora), lo que permitirá evitar gastos en servidores en la nube y hará más sencillo el desarrollo académico. 3. ****Arquitectura: Monolito vs Microservicios/Servicios**** **Monolito modular**  **Microservicios / SOA**  **Justificación:**  El proyecto es académico y de pequeña escala, un monolito facilita el desarrollo y mantenimiento sin la complejidad de microservicios. 4. ****Proceso de desarrollo: Plan-Dirigido vs Ágil/DevOps**** **Plan-dirigido (Cascada / V-Model)**  **Ágil / DevOps (Scrum, Kanban, XP)**  **Justificación:**  permite planificar y controlar cada fase de manera secuencial, asegurando mayor control de calidad, documentación completa y seguimiento estructurado. |

# Etapa 3: Selección del Modelo de Desarrollo

3.1 Descripción del Proyecto

El proyecto implica crear un sistema de gestión de inventario para una tienda minorista (almacén), ayudando a registrar la entrada y salida de los productos, verificar la cantidad de stock en tiempo real y a optimizar la forma en que se gestionan las actividades relacionadas con el control del inventario, logrando disminuir quiebres y perdidas de stock, para así mejorar las ventas, los tiempos de control de inventario, una mayor satisfacción al cliente y una mejor planificación de compras.

3.2 Modelos de Desarrollo Considerados (cascada, incremental, iterativo, ágil)

* **Modelo** **iterativo:** se basa en desarrollar un producto a través de versiones sucesivas. Es decir, se comienza con una versión inicial del producto y a través de varias iteraciones, se va mejorando progresivamente. Cada nueva versión incorpora mejoras, correcciones o nuevas funciones basadas en lo aprendido en la versión anterior. El objetivo principal es mejorar el producto hasta que cumpla con los requisitos esperados.
* **Modelo Incremental:** se centra en construir el sistema de manera gradual, dividiéndolo en partes o módulos funcionales que se entregan y ponen en uso progresivamente. Cada incremento añade nuevas funcionalidades completas al producto, de manera que al final de cada etapa se cuenta con un sistema operativo más robusto que el anterior. El valor clave aquí es que el cliente recibe partes utilizables del software antes de que el proyecto esté terminado en su totalidad.

La diferencia entre estos modelos es que iterativo se centra en refinar continuamente versiones del mismo producto hasta alcanzar el resultado esperado, mientras que el modelo incremental se enfoca en entregar funcionalidades completas por etapas, agregando nuevas piezas al sistema hasta completarlo. En la práctica, ambos enfoques suelen combinarse: se puede avanzar de manera incremental en las funcionalidades, y dentro de cada incremento aplicar ciclos iterativos de mejora.

3.3 Justificación del Modelo Seleccionado (Cómo este modelo se adapta al ciclo de vida del proyecto elegido)

Se eligió la mezcla entre los modelos iterativo e incremental permite aprovechar las ventajas de ambos enfoques y adaptarse mejor a las necesidades del proyecto, ya que por un lado, el modelo incremental asegura que el sistema se vaya construyendo por etapas, entregando funcionalidades completas y utilizables en plazos más cortos, lo que facilita al cliente obtener valor temprano y comprobar que el desarrollo responde a sus requerimientos. Esto reduce la incertidumbre y mejora la percepción de avance del trabajo, mientras que, el modelo iterativo permite que cada uno de esos incrementos se refine en ciclos sucesivos, corrigiendo errores, ajustando detalles y perfeccionando la solución antes de avanzar a nuevas funcionalidades. De esta manera, no solo se asegura la entrega gradual del producto, sino también la mejora continua de la calidad y la capacidad de adaptarse a cambios en los requerimientos o condiciones del entorno.

# Etapa 4: Recolección y Análisis de Requerimientos

En esta fase, deberás identificar y documentar los requisitos del sistema. Estos pueden dividirse en requisitos funcionales (qué debe hacer el sistema) y requisitos no funcionales (cómo debe comportarse el sistema). Identificar y justificar la técnica que ocuparon para identificar los requisitos (Como documento Anexo deben entregar el [DER](about:blank))

4.1 Identificación de la o las técnicas para la recolección de los requerimientos (Mostrar la evidencia).

Para la recolección de requerimientos se optó por utilizar dos técnicas complementarias, entrevista no estructurada y encuesta mediante Google Forms. La entrevista no estructurada se aplicó con el objetivo de obtener información más abierta, detallada y flexible, permitiendo que el cliente exprese sus necesidades y expectativas sin limitaciones rígidas en las preguntas. En complemento, la encuesta vía Google Forms se eligió para recopilar datos de manera rápida y sistemática. La combinación de ambas técnicas nos permitió obtener una visión integral tanto de los requerimientos como de las necesidades del cliente para el sistema de gestión de inventarios de la tienda minorista

4.2 Requisitos Funcionales y No funcionales (listado).

En este apartado se presenta un listado de los requerimientos del sistema, tanto funcionales como no funcionales, donde el primero corresponde al de mayor importancia y el último al de menor relevancia, también, se clasificarán los tipos de requerimientos no funcionales que permiten cumplir los objetivos del usuario asegurando un funcionamiento eficiente y confiable.

* Requisitos funcionales

01 – Registro de usuarios con roles  
02 – Restricción de funcionalidades por rol  
03 – Registro de proveedores  
04 – Búsqueda de proveedores por celular  
05 – Modificación de proveedores  
06 – Eliminación de proveedores  
07 – Registro de productos  
08 – Búsqueda de productos por código  
09 – Modificación de productos  
10 – Eliminación de productos  
11 – Registro de entradas de productos  
12 – Consulta de entradas  
13 – Modificación de notas en entradas  
14 – Eliminación de entradas  
15 – Actualización automática de stock en entradas  
16 – Registro de salidas de productos  
17 – Actualización automática de stock en salidas  
18 – Consulta de salidas  
19– Modificación de notas en salidas  
20– Eliminación de salidas  
21– Generación de reportes de inventario  
22– Notificación de stock mínimo  
23– Historial de operaciones  
24– Respaldo de la base de datos  
25– Restauración de la base de datos

* Requisitos No Funcionales

01 – Cifrado de contraseñas (seguridad)  
02 – Tiempo de respuesta en búsquedas (Rendimiento/Eficiencia)  
03 – Conexiones simultáneas soportadas (Escalabilidad)  
04 – Escalabilidad del sistema (Escalabilidad)  
05 – Compatibilidad en instalación (Portabilidad/Compatibilidad/interacción)  
06 – Interfaz intuitiva y validaciones (Usabilidad/Interfaz de usuario)  
07 – Limpieza automática de formularios (Usabilidad/ Interfaz de usuario)  
08 – Mensajes claros de éxito y error (Usabilidad/ Interfaz de usuario)  
09 – Arquitectura MVC (Mantenibilidad/Interacción)