Guía Informe final de proyecto de Desarrollo de SIAs

# Introducción

Este documento guía está diseñado para ayudarte a través de todas las fases del ciclo de vida del desarrollo de software utilizando principios de ingeniería de software. Completa cada sección según corresponda al proyecto que te haya tocado. Recuerda que la calidad y la claridad en cada una de estas fases son cruciales para el éxito de tu proyecto.

# Etapa 1: Visión y Alcance del Proyecto

En esta fase se debe establecer con claridadel **para qué**, **para quién**, **qué incluirá** y **qué no** incluirá el sistema a construir. Esta sección guía todas las decisiones posteriores (modelo de desarrollo, requerimientos, diseño, pruebas y operación).

1.1 Visión (máx de 3 frases)

*Describa el propósito del SIA, a quién sirve y el valor que entrega.*

*“Construir un sistema inteligente de gestión de inventario para tiendas minoristas que ofrezca autonomía operativa y visibilidad en tiempo real para toma de decisiones basada en datos, optimizando su capital y minimizando las ineficiencias en stock.”*

1.2 Problema y contexto

*Problema/ necesidad que se intenta resolver (1 párrafo).*

*Evidencia breve (dato, observación, requerimiento institucional).*

*El problema central que se resolverá es la inadecuada gestión del inventario que se genera por faltas críticas de stock de productos altamente solicitados por los clientes y la incapacidad de gestionar efectivamente las fechas de caducidad de productos perecederos, lo que provoca pérdidas en ventas y mermas económicas en productos vencidos, demostrando una ineficiencia operativa en la falta de automatización para aprovisionamiento, la recopilación de información y análisis.*

1.3 Usuarios y stakeholders

*Lista de usuarios y stakeholders (internos/externos).*

*Dolencias/objetivos de cada uno (1–2 bullets).*

*Vendedor:*

* *Requieren acceso rápido al stock disponible.*
* *Pérdida de ventas y deterioro del servicio al cliente por la indisponibilidad de productos en el momento de la venta.*

*Supervisor:*

* *Falta de datos en tiempo real para optimizar la reposición y reportes.*
* *Requiere dashboard interactivos que centralizan la información crítica del inventario.*

*Administrador (Dueño):*

* *Pérdida de rentabilidad por ineficiencias operativas.*
* *Necesidad de reportes finales, alertas tempranas de caducidad y stock crítico.*

*Cliente:*

* *Experiencia negativa al no estar disponible el producto.*
* *Expectativas de obtener disponibilidad inmediata en productos solicitados.*

1.4 Propuesta de valor

*Beneficios esperados para usuarios/organización (3–5 bullets).*

*Los beneficios que se esperan entregar a los usuarios que manejan el sistema de gestión de inventario y a la tienda minorista es:*

* *Reducir problemas de falta de stock con alertas tempranas y análisis de demandas precisos.*
* *Visibilidad en tiempo real de inventario permitiendo un control preciso e inmediato de productos.*
* *Reducir pérdidas financieras al disminuir productos pérdidos por caducidad.*
* *Generar automáticamente reportes de inventario, eficiencia operativa y rotación de inventario, eliminando la dependencia de procesos manuales y mejorando rentabilidad.*
* *Mejorar la experiencia del cliente con constante disponibilidad de productos claves.*

# Etapa 2: Organización y Planificación del Proyecto

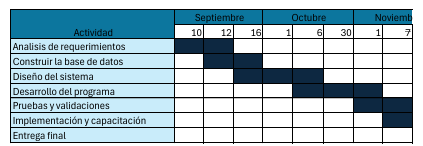
En esta fase, se debe realizar toda la organización administrativa del proyecto. Esto incluye la planificación de tareas, asignación de roles, creación de una carta Gantt, y selección de herramientas de gestión de proyectos. Este paso es crucial para asegurar que el proyecto avance de manera ordenada y eficiente.

2.1 Definición de Roles y Responsabilidades.

1. Jefe de proyecto
2. Analista de negocio/ Analista de requerimientos
3. Arquitecto/ Diseñador
4. Equipo de desarrollo
5. Encargado de Documentación/ Librería Técnica
6. Soporte/ mesa de ayuda

| Encargado | Roles | Responsabilidades |
| --- | --- | --- |
| Estephania Valencia | Arquitecto | Encargado del diseño del sistema, creación de diagramas y modelos para guiar el desarrollo, cumpliendo requisitos de escalabilidad, seguridad y mantenimiento. |
|  | Soporte/mesa de ayuda | Resuelve incidencias y consultas de usuarios brindando asistencia técnica post-implementación, además de realizar reportes de problemas recurrentes con el fin de entregar una mejora continua. |
| Francisca Zeballos | Equipo de desarrollo | Realiza pruebas unitarias y de integración, corrige errores y mejora el rendimiento, además de colaborar en las revisiones técnicas. |
|  | Encargado de documentación | Organiza manuales, guías y procedimientos asegurándose de que la documentación esté organizada y actualizada para que sea accesible para todos y apoyar la formación personal. |
| Katherine Segovia | Analista de negocios | Recopila y analiza las necesidades del negocio y los usuarios, también documenta los requerimientos del negocio funcionales y no funcionales. |
| Natalia Ortega | Jefe de proyecto | Planifica y coordina todo el proyecto, define objetivos, cronogramas y recursos, asegurándose además del cumplimiento de plazos y calidad. |

* 1. Planificación de Tareas (Carta Gantt/ Kanban.).



2.3 Herramientas de Organización y Gestión del Proyecto.

Para la organización y gestión del proyecto se proponen los siguientes esenciales para coordinar, monitorear y planificar ideas

1. Gestión de Riesgos (Identificación de riesgos):

·Retraso en desarrollo por problemas técnicos.

·Cambios de requerimientos por parte del cliente.

· Falta de capacitación del personal del kiosco.

Plan de mitigación:

·Reuniones de seguimiento semanales.

·Documentación clara de requerimientos.

·Plan de capacitación previo a la implementación

2. Registro de Avances / Actas de Reunión

Documento breve después de cada reunión de control que contiene tareas completadas, problemas detectados, acuerdos y próximos pasos.

3. Herramientas de Software: Donde usaremos GitHub y como lenguaje de programación Python y MySQL.

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*** | | *Acceso no autorizado a la BD* | *Seguridad* | *Media* | *Alto* | *Implementar autenticación robusta, aplicar y definir políticas de roles y permisos, además de encriptar datos sensibles* | *Arquitecto/Diseñador* | | *Pérdida de información crítica (ej: inventario, transacciones)* | *Datos/continuidad* | *Baja* | *Alto* | *Establecer protocolos periódicos. realizar pruebas de restauración.* | *Equipo de desarrollo* | | *Caída del servidor durante clase/demo* | *Continuidad* | *media* | *Alto* | *Tener un plan de contingencia con servidor de respaldo listo.* | *Equipo de desarrollo/ soporte* | | *Fuga de datos sensibles de clientes (proveedores, precios)* | *Seguridad/datos* | *media* | *Alto* | *Se protegen los datos usando información falsa para pruebas, monitoreando quién accede a la información confidencial.* | *Arquitecto/Diseñador* | | *Error humano al borrar registros (ej: stock, flujos)* | *Datos* | *Alta* | *Media* | *Usar borrador lógico, pedir confirmación en acciones riesgosas y guardar registro de todos los cambios realizados.* | *Equipo de desarrollo* | | *Fallas en la integración con otros sistemas (ej: ERP, punto de venta)* | *Continuidad* | *Media* | *Alta* | *Crear y probar interfaces usando datos de simulación primero, e integrar los sistemas de forma progresiva y planificada.* | *Arquitecto/ Equipo de soporte.* | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

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: Un sistema de inventario SaaS es accesible, reduce costos y permite la accesibilidad. El proveedor se encarga del mantenimiento y actualizaciones, que es ideal para la escalabilidad. 2. **Infraestructura: Nube vs On-Premise** On-Premise**.**  **Nube (IaaS/PaaS/DBaaS)** Justificación:Ofrece escalabilidad automática, alta disponibilidad y ahorro de costos. Incluye servicios administrados que simplifican el mantenimiento y garantizan seguridad integrada. 3. **Arquitectura: Monolito vs Microservicios/Servicios** Monolito modular  **Microservicios / SOA** Justificación: Permite desarrollar y escalar módulos claves ( cómo stock, órdenes o reportes) de forma independiente. Facilita la integración con otros sistemas y acelera la implementación de nuevas funciones sin registro global. 4. **Proceso de desarrollo: Plan-Dirigido vs Ágil/DevOps** Plan-dirigido (Cascada / V-Model)  **Ágil / DevOps (Scrum, Kanban, XP)** Justificación: Permite adaptarse a cambios, entregar funcionalidades rápido y mejorar continuamente el sistema. Ideal para proyectos iterativos. |
| --- |

# Etapa 3: Selección del Modelo de Desarrollo

En esta fase, deberás seleccionar el modelo de desarrollo de software que utilizarás para guiar el proyecto. Justifica tu elección basándote en las características del proyecto y las características de cada modelo.

3.1 Descripción del Proyecto

El proyecto consiste en desarrollar un sistema de gestión de inventario para una tienda minorista, restringiendo el acceso según el rol del usuario que usará el software, permitiendo optimizar y gestionar el control de entrada y salida de productos, evitando los quiebres de stock y la caducidad de productos perecibles, también permitirá registrar los productos de acuerdo a su categoría, lo que contribuye a una administración más eficiente. Este software, además, generará reportes básicos basados en los datos recolectados según el tiempo seleccionado con información como movimientos diarios, valores totales (ingresos) y gráficos con productos con mayor y menor rotación gracias a la información entregada en tiempo real, logrando facilitar la toma de decisiones informada para el negocio, lo que mejora la eficiencia de la tienda y contribuye a la mejora en la atención al cliente.

3.2 Modelos de Desarrollo

Considerados (cascada, incremental, iterativo, ágil)

Para el desarrollo de este sistema de gestión de inventario se considera el modelo de desarrollo iterativo e incremental.

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

El modelo iterativo-incremental es el más adecuado para el desarrollo del sistema de gestión de inventario, ya que permitirá construir un software en etapas sucesivas, entregando versiones parciales comprendiendo los distintos módulos que estarán interconectados, y así mejorar cada segmento del sistema, de forma que se va desarrollando cada componente más complejo de manera iterativa e incremental. Además, se podrá tener mayor consideración de los requisitos que pueden ir cambiando y adaptándose de acuerdo al constante cambio del entorno que tiene una tienda minorista, lo que le dará mayor valor agregado en las re-evaluaciones y ajustes a las necesidades reales que pida el cliente durante el desarrollo, asegurando que el software entregado se ajuste exactamente a las necesidades dinámicas del negocio; también, se toma en cuenta que cada fase (desarrollo de cada módulo) tendrá su propio diseño, implementación y prueba antes de ser unificado en el sistema central, por lo que permite identificar y corregir de forma temprana los errores o problemas detectados.

# 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.

4.2 Requisitos Funcionales y No funcionales.

4.3 Requisitos de seguridad y privacidad (ejemplo: cifrado, roles, logs).

4.4 Otros Requisitos (de actores, funcionales, interacción).

4.5 Priorización de Requerimientos.(Alta, Medio, Baja)

# Etapa 5: Diseño del Sistema

En esta fase, debes diseñar la arquitectura general del sistema. Define los componentes principales y sus interacciones, así como la base de datos que soportará al sistema.

3.1 Diagrama de Flujo de Datos.

3.2 Modelo de Datos Semánticos.

3.3 Modelo de Dominio.

3.3.1 Diagrama de Clases.

3.4 Modelo de Objetos.

3.4.1 Diagrama de Objetos.

3.4.2 Diagrama de Secuencia.

3.4 Diagrama de Arquitectura del Sistema (monolítico modular / servicio).

3.5 Diseño de la Base de Datos (Modelo Relacional).

3.6 Diseño de la Interfaz de Usuario.

3.7 Esquema de seguridad: autenticación/autorización, cifrado, segregación de datos.

3.8 Política de acceso y roles.

# Etapa 4: Desarrollo e Implementación

En esta fase, el diseño del sistema se convierte en código ejecutable. Documenta las decisiones importantes que tomaste durante la implementación y describe cómo implementaste los principales módulos del sistema.

4.1 Lenguajes y Herramientas Utilizados.

4.2 Implementation de Módulos Principales.

* 1. Problemas y Soluciones Durante la Implementación.

4.4 Gestión de versiones (Git/CI/CD).

4.5 Políticas de backup en desarrollo (ej.: dumps automáticos de BD).

# Etapa 5: Pruebas y corrección de errores

En esta fase, debes probar el sistema para asegurarte de que cumple con los requisitos especificados. Esto incluye pruebas unitarias, pruebas de integración, pruebas del sistema completo y pruebas de aceptación.

5.1 Plan de Pruebas (unitarias, integración, siatema, aceptación)

5.2 Casos de prueba documentados.

5.3 Pruebas de seguridad: inyección SQL, manejo de errores, validación de datos.

5.4 Pruebas de recuperación: restauración de BD desde backup, medición de RTO/RPO.

5.4 Reporte y corrección de errores (Formato para registrar fallos y tiempo de reparación).

| Información General del Proyecto   * Nombre del Proyecto: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ * Nombre del Software: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ * Fecha de Inicio del Registro: //\_\_\_\_ * Fecha de Fin del Registro: //\_\_\_\_ * Nombre del Estudiante/Grupo: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ * Nombre del Responsable: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  | **ID de Fallo** | **Fecha y Hora del Fallo** | **Descripción del Fallo** | **Condiciones Previas** | **Impacto del Fallo** | **Tiempo de Detección (minutos)** | **Tiempo de Reparación (minutos)** | **Estado de la Reparación** | **Comentarios Adicionales** | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 1 | **/**/\_\_\_\_ hh | Breve descripción del fallo | Ejemplo: Al intentar dividir por cero, el software se cierra inesperadamente | Alto | 5 | 15 | Reparado / Pendiente | Detalles adicionales si es necesario | | 2 | **/**/\_\_\_\_ hh | Breve descripción del fallo | Ejemplo: Error al marcar tarea como completada | Medio | 3 | 10 | Reparado / Pendiente | Detalles adicionales si es necesario | | 3 | **/**/\_\_\_\_ hh | Breve descripción del fallo | Ejemplo: Resultados incorrectos en la división de números | Alto | 2 | 5 | Reparado / Pendiente | Detalles adicionales si es necesario | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

# Etapa 6: Seguridad, Respaldo y Recuperación de Datos

En esta etapa se definen, implementan y documentan las medidas que garanticen la integridad, disponibilidad y confidencialidad de la información gestionada por el sistema. Se deben aplicar controles de seguridad como políticas de acceso, gestión de usuarios y cifrado de datos. Asimismo, se establecen y prueban las políticas de respaldo periódico de la base de datos y de los archivos críticos del sistema, definiendo su frecuencia, responsables y herramientas utilizadas. Finalmente, se deben diseñar y ejecutar mecanismos de recuperación que permitan restaurar la información en caso de fallos, incidentes de seguridad o pérdida de datos, midiendo los tiempos de recuperación (RTO) y los puntos de restauración alcanzados (RPO).

6.2 Plan de respaldo: frecuencia, herramientas utilizadas, validación de integridad.

6.3 Plan de recuperación: pasos para restaurar, responsables, tiempos medidos.

6.4 Evidencia de simulacros de restore.

### Conclusiones y Lecciones Aprendidas

* Cómo se cumplieron los RA del curso.
* Valor del sistema desarrollado para la organización.
* Próximos pasos de mejora.