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)

*Buscamos facilitar la eficiencia dentro de talleres automotrices, mediante organización y rapidez de localización y uso de insumos*

1.2 Problema y contexto

*Actualmente, muchos talleres automotrices registran el inventario de manera manual en cuadernos o planillas, lo que genera errores humanos, pérdidas de piezas, desabastecimiento y exceso de stock en otros casos. La ausencia de un sistema digital centralizado afecta la eficiencia y a calidad de los servicios ofrecidos por los talleres mecánicos*

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

1.3 Usuarios y stakeholders

***Administrador del taller(Internos):*** *Requieren gestionar pedidos de clientes de y compras a proveedores de manera continua.*

*Una Mejor trazabilidad sobre el stock es igual a una Mejor fiabilidad de reportes por tanto una Mejora en la toma de decisiones sobre el taller.*

***Operarios Mecánicos(Internos):*** *Requieren rapidez para el registro de repuestos para no perder tiempo que pueden usar en realizar servicios de revisión y reparación.*

***Propietarios del Taller(Interno):*** *Busca aumentar rentabilidad del negocio, mediante el control claro de sus insumos*

***Proveedores(Externos):*** *Necesitan recibir pedidos claros y oportunos para asegurar el suministro*

***Clientes(Externos):*** *Necesitan un servicio que ofrezca plazos razonables para la realización del servicio, que no se malgaste tiempo en el servicio*

1.4 Propuesta de valor

*Control eficiente del inventario en tiempo real*

*Reducción de pérdidas y compras innecesarias*

*Alertas de stock mínimo*

*Generación de reportes automáticos de consumo y costos*

# 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.

* 1. Definición de Roles y Responsabilidades.

Lider del Proyecto: Coordinar tareas, validar avances

Diseñador UI/UX: Elaborar prototipos de interfaz en Figma

Desarrollador Backend: Implementar la lógica de negocio y conexión con la base de datos.

Desarrollador Frontend: Crear la interfaz de usuario y conectar con el backend.

Tester: Diseñar casos de prueba y validar la calidad del sistema.

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

1. Semana 1: Revisión de requisitos y necesidades del taller y redacción del Documento de especificación de requisitos; Análisis de alcance
2. Semana 2-4: Diseño del sistema (BD, diagramas, interfaz).
3. Semana 5-6(Backend): Programación de la lógica del negocio: entrada/salidas de inventario, alertas
4. Semana 7 (Frontend): Implementación de formularios, tablas y reportes en la interfaz.
5. Semana 8: Conexión de Frontend con Backend y pruebas de funcionamiento básico
6. Semana 9: Pruebas unitarias, integración y correcciones.
7. Semana 10: Documentación y entrega final.
   1. Herramientas de Organización y Gestión del Proyecto.

GitHub: control de versiones, issues, ramas.

GitHub Projects: gestión de tareas con Kanban.

Google Drive/Docs: almacenamiento colaborativo de documentos.

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 roles, autenticación y cifrado. | Dev Backend | | *Pérdida de información crítica* | *Datos* | Media | Alto | Respaldos automáticos diarios de la BD. | Líder | | *Caída del servidor durante clase/demo* | *Continuidad* | Baja | Medio | Uso de hosting confiable y pruebas previas. | Dev Backend | | *Fuga de datos sensibles de clientes* | *Seguridad* | Baja | Alto | Políticas de acceso y encriptación de datos. | Líder | | *Error humano al borrar registros* | *Datos* | Alta | Medio | Confirmación doble y logs de auditoría. | Todo el equipo | |  |  |  |  |  |  | |  |  |  |  |  |  | |  |  |  |  |  |  | |  |  |  |  |  |  | |

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:** **Reduce costos de infraestructura, permite acceso remoto y actualizaciones automáticas.** 2. ****Infraestructura: Nube vs On-Premise**** **On-Premise.**  **Nube (IaaS/PaaS/DBaaS)  Justificación: Escalabilidad, menor inversión inicial y mejor recuperación ante fallos.** 3. ****Arquitectura: Monolito vs Microservicios/Servicios**** **Monolito modular**  **Microservicios / SOA  Justificación: El proyecto es de pequeña escala, menor complejidad, fácil de implementar y mantener por un equipo reducido.** 4. ****Proceso de desarrollo: Plan-Dirigido vs Ágil/DevOps**** **Plan-dirigido (Cascada / V-Model)**  **Ágil / DevOps (Scrum, Kanban, XP)**  **Justificación: Permite iteraciones rápidas, feedback constante y adaptabilidad a cambios en requisitos** |

# 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

Se desarrollará un sistema de gestión de inventario para un taller automotriz, orientado a registrar entradas y salidas de repuestos, gestionar stock mínimo, generar reportes y enviar alertas automáticas.

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

Cascada: Demasiado rígido para las funcionalidades

Incremental: Permite entregar funcionalidades en fases, pero es tan adaptable a cambios.

Iterativo: Permite refinamiento gradual a costa de tiempo que se usa para completar cada ciclo de mejora, en vez diferentes mejoras en diferentes áreas

Ágil: Adaptable a cambios, Modular,

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

Se selecciona el modelo Ágil (Kanban), ya que el proyecto requiere flexibilidad, entregas parciales de módulos funcionales y se ajusta al trabajo en equipo con GitHub.

# 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](file:///J:\Mi%20unidad\UCN\CURSO%202024\CLASES\SEMESTRE%20II\DESARROLLO%20SIA\SEMANA%204\Documento%20de%20Especificación%20de%20Requisitos.docx))

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

4.2 Requisitos Funcionales y No funcionales (listado).

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

4.4 Otros Requisitos (de actores, de interacción, etc.)(debe colocar el pro).

4.5 Priorización de Requerimientos.

# 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.

5.1 Diagrama de Flujo de Datos.

5.2 Modelo de Datos Semánticos.

5.3 Modelo de Dominio.

3.3.1 Diagrama de Clases.

5.4 Modelo de Objetos.

3.4.1 Diagrama de Objetos.

3.4.2 Diagrama de Secuencia.

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

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

5.7 Diseño de la Interfaz de Usuario.

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

5.9 Política de acceso y roles.

# Etapa 6: 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 Implementación 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, sistema, 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.