Guía Informe final de proyecto de Desarrollo de SIAs

# Integrantes:

1. Heidy Calizaya
2. Rodrigo Diaz
3. Nicole García
4. Ricardo Zepeda

# Introducción

En el presente informe se desglosa la información correspondiente a nuestro proyecto el cual consiste en un desarrollo de software para un sistema de gestión de biblioteca virtual que estará disponible para la Universidad Católica del Norte.

# Etapa 1: Visión y Alcance del Proyecto

## 1.1 Visión (máx de 3 frases)

Construir un sistema de reservas de cubículos y libros para la biblioteca de la UCN, que permita registrar entradas y salidas en tiempo real de las reservas, reduciendo la saturación del sistema.

## 1.2 Problema y contexto

Mal funcionamiento del sistema de reservas de cubículos, evidenciando de forma presencial la poca eficiencia del sistema actual, aquejando a diversos estudiantes de la universidad. Aprovechando para mejorar el sistema de reservas de libros.

## 1.3 Usuarios y stakeholders

**Usuarios:** Alumnos y personal de la biblioteca

**Stakeholders:** Estudiantes, funcionarios de la universidad y personas que inviertan en la universidad.

## 1.4 Propuesta de valor

* Comodidad
* Accesibilidad rápida y simple
* Mejora eficiencia operativa que reduce la carga al personal de la biblioteca

# Etapa 2: Organización y Planificación del Proyecto

## 2.1 Definición de Roles y Responsabilidades.

**Jefe de proyecto:** Nicole Garcia

**Analista de negocios:** Ricardo Zepeda

**Arquitecto:** Rodrigo Diaz

**Equipo de desarrollo:** Todos

**Tester:** Ricardo Zepeda

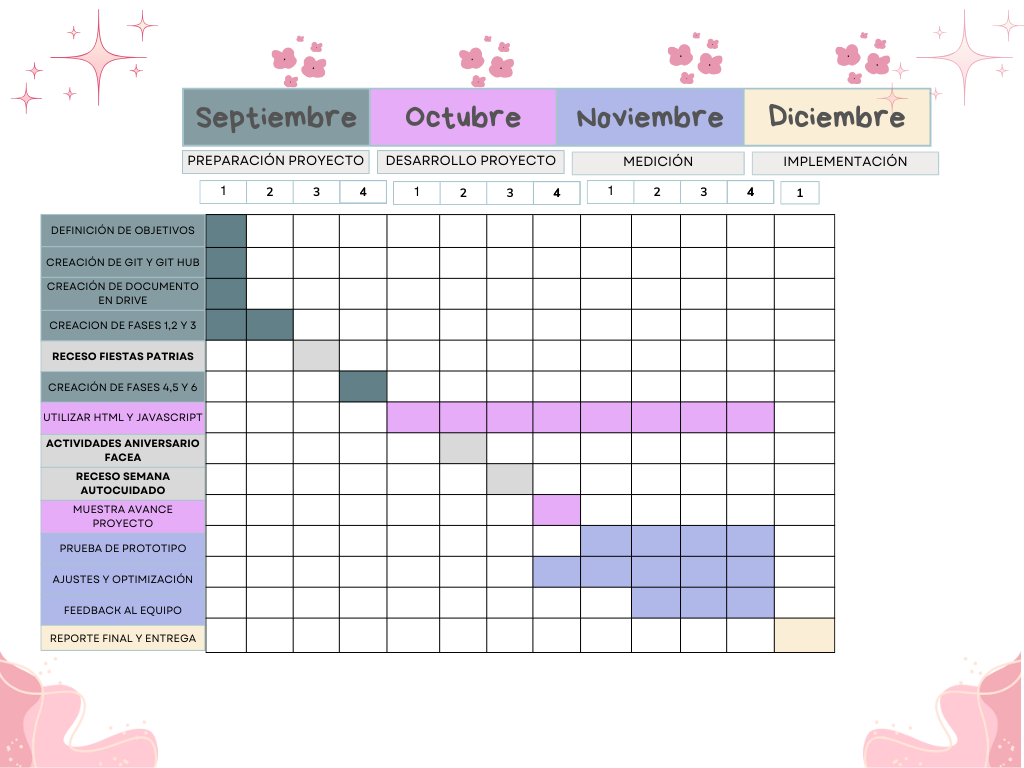
**Administrador Base de datos:** Nicole Garcia

**Administrador de sistema:** Heidy Calizaya

**Encargado de documentación:** Rodrigo Diaz

**Soporte:** Todos

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



## 2.3 Herramientas de Organización y Gestión del Proyecto.

En este proyecto se utilizará Carta Gantt y Kanban, para que el seguimiento de los avances del proyecto se cumplan dentro del plazo establecido de manera ordenada.

## 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* | *Politicas de acceso y control de permisos* | *Administrador Base de Datos* | | *Pérdida de información crítica* | *Continuidad* | *Baja* | *Alto* | *Guardar Información en los repositorios* | *Tester* | | *Fallas del computador* | *Continuidad* | *Baja* | *Media* | *Tener otro computadores* | *Soporte* | | *Fallo en registro de clientes* | *Continuidad y datos* | *Media* | *Alto* | *Revisar el código* | *Arquitecto* | | *Fallas en guardar datos en la base de datos* | *Datos* | *Media* | *Media* | *Verificar los datos guardados* | *Soporte* | | *Duplicación de datos* | *Datos* | *Media* | *Alto* | *Verificar el código de la base de datos* | *Administrador de base de datos* | | *Interfaz confusa para usuarios* | *Datos* | *Media* | *Media* | *Capacitación de uso de la interfaz* | *Equipo de desarrollo* | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

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

| 1. **Producto instalado vs SaaS (Software como servicio)**  * ~~SaaS (Servicio en la nube)~~   **Justificación:** El programa será utilizado por los estudiantes, de manera online. 2. **Infraestructura: Nube vs On-Premise**  * ~~Nube (IaaS/PaaS/DBaaS)~~   **Justificación:** La información será guardada en la nube, donde se usará una base de datos 3. **Arquitectura: Monolito vs Microservicios/Servicios**  * ~~Microservicios / SOA~~   **Justificación:**  Optamos por un sistema de Micro Servicios/SOA porque en este caso se va a manejar un sistema de biblioteca en donde tendrán distintas funciones que son independientes entre sí como: gestionar la entrega de libros y entrega de cubículos para estudio, además de reportar el control de usuarios. 4. **Proceso de desarrollo: Plan-Dirigido vs Ágil/DevOps**  * ~~Ágil / DevOps (Scrum, Kanban, XP)~~   **Justificación:** El sistema requiere flexibilidad porque puede que los requisitos del programa cambien con el tiempo. Un enfoque Ágil con Scrum permite entregar los módulos funcionales en iteraciones cortas como: reservas de cubículos, préstamos de libros y de eso se puede sacar reportes. Además el uso de DevOps puede facilitar una rápida actualización del sistema web. |
| --- |

# 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

Aplicación que se encarga de la administración de una biblioteca online y sistema de reserva de cubículos para la universidad católica del norte. Este programa se encargará de que el estudiante pueda reservar libros y cubículos de manera ordenada y eficiente para aligerar el trabajo de los funcionarios de la biblioteca.

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

En este programa se utilizará el modelo de desarrollo ágil.

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

El modelo de desarrollo ágil nos permitirá evitar tener un modelo de desarrollo estático y nos permitirá a poder realizar cambios de manera eficiente al ser necesario y poder trabajar de manera iterativa e incremental.

# 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.

# 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.

**Tipos de objetivos de una clínica**

**Objetivos estratégicos:**  Mejorar el orden de la información de los pacientes de manera eficiente para que la información no sea dispersa y de este modo asegurar que cada usuario cuente con historial de exámenes, registros, etc. Fortalecer infraestructura física, tecnológica y de información para dar un servicio de calidad.

**Objetivos operacionales:** Garantizar seguridad del cliente. Optimizar la programación de citas

**Objetivos tácticos:** Mediante un sistema de ERP con una base de datos ordenada y monitoreando la atención al cliente mediante encuestas de satisfacción.

requisito no funcional para hotel:

* Los usuarios solo podrán usar la App mientras estén en la misma ciudad que el hotel.
* El usuario puede ingresar a la APP solo con su RUT, contraseña y n° de habitación

fecha: 26