

Documento de formulación del proyecto

Luis Fernando Martínez Niño

Área:

Análisis y Diseño de Sistemas

Presentado a:

Tatiana Cabrera

Corporación Universitaria Iberoamericana

Colombia

2025



TABLA DE CONTENIDOS:

INTRODUCCIÓN	4
ALCANCE DEL PROYECTO	6
PLANTEAMIENTO DEL PROBLEMA	8
CONTEXTUALIZACIÓN	9
OBJETIVO GENERAL	10
OBJETIVOS ESPECÍFICOS	10
METODOLOGÍA.....	11
MATRIZ DE RIESGOS.....	14
Fase de Análisis	15
LEVANTAMIENTO DE INFORMACIÓN:	15
MAPA DE STAKEHOLDERS.....	17
DIAGRAMA DE FLUJO DE SOLUCIÓN	18
REQUISITOS FUNCIONALES Y NO FUNCIONALES	19
REQUISITOS NO FUNCIONALES:	21
ANEXOS:	24
CONCLUSIONES	25
BIBLIOGRAFÍA	26

TABLA DE ILUSTRACIONES

Ilustración 1 Diagrama Causa Efecto	8
Ilustración 2. Mapa de empatía	9
Ilustración 3 Tablero Trello.....	13
Ilustración 4 Mapa de Stakeholders	17
Ilustración 5 Diagrama de flujo.....	18
Ilustración 6 HU-1: Crear cronogramas.....	23



TABLA DE TABLAS

Tabla 1 Matriz de riesgos.....	14
Tabla 2 RF-001	19
Tabla 3 RF-002	19
Tabla 4 RF-003	20
Tabla 5 RF-004	20
Tabla 6 RF-005	21
Tabla 7 RNF-001	21
Tabla 8 RNF-002	21
Tabla 9 RNF-003	21
Tabla 10 RNF-004.....	22
Tabla 11 RNF-005.....	22
Tabla 12 RNF-006.....	22
Tabla 13 RNF-007.....	22
Tabla 14 RNF-008.....	22

ANEXOS

Anexo A Workshop con líderes.....	24
Anexo B Encuesta para crear mapa de empatía.....	24



INTRODUCCIÓN

Link GitHub: <https://github.com/Lufemani02/ProyectoCronogramaIPUC>

El presente proyecto académico tiene como objetivo el diseño e implementación de un sistema de software para apoyar a los líderes de los comités de la IPUC Alfonso López en la gestión de cronogramas de servicio. Actualmente, los líderes enfrentan serias dificultades al momento de organizar turnos, ya que los servidores participan en varios comités y no existe una herramienta que muestre su disponibilidad de manera centralizada. Esto ocasiona choques de horarios, desorganización y pérdida de tiempo en la planeación manual. La solución propuesta consiste en un sistema que permita crear, visualizar y exportar cronogramas, con validación de disponibilidad de servidores y alertas en caso de conflictos. Además, contempla la posibilidad de gestionar usuarios (CRUD de servidores) y consultar su disponibilidad en tiempo real, asegurando que la información esté siempre actualizada. Para brindar mayor flexibilidad, los líderes podrán exportar los cronogramas en Excel o imagen, facilitando la socialización con otros miembros de la iglesia. El desarrollo del sistema se abordará con un enfoque ágil, utilizando la metodología Kanban, lo que permitirá mantener un flujo de trabajo visual, claro y adaptable. Asimismo, se aplicaron técnicas de Design Thinking en las primeras fases, con el fin de empatizar con los usuarios, definir el problema y generar ideas de solución centradas en sus necesidades reales.

El sistema contará con tres componentes principales:

- Front-end: diseño de interfaces gráficas y prototipos navegables para la gestión de cronogramas.
- Back-end: implementación de procesos y funcionalidades organizadas en módulos, incluyendo autenticación, validación de conflictos y generación de alertas.



- Persistencia: almacenamiento local y persistente de la información, con posibilidad de escalar a bases de datos relacionales en un entorno real.

De esta manera, el proyecto no solo busca cumplir con los objetivos académicos, sino también entregar un producto funcional que mejore la organización de los cronogramas y brinde tranquilidad a los líderes de comité al momento de coordinar los turnos de los servidores.

Fase de planificación

Justificación

La necesidad de este proyecto surge de las dificultades que enfrentan los líderes de los comités de la IPUC Alfonso López al organizar cronogramas de servicio. La ausencia de una herramienta que centralice la disponibilidad de los servidores ocasiona choques de horarios, duplicidad en la asignación de tareas y pérdida de tiempo en la planeación manual. Un sistema de software que permita visualizar, organizar y exportar cronogramas representa una solución práctica y eficiente a esta problemática. Al ofrecer funcionalidades como validación de conflictos de horarios, gestión de usuarios y consultas rápidas de disponibilidad, se logra una herramienta adaptada a las necesidades reales de los líderes. La elección de la metodología ágil Kanban responde al carácter académico y al tamaño reducido del equipo de trabajo. Esta metodología facilita un flujo visual y flexible de tareas, lo cual resulta adecuado para un proyecto que evoluciona progresivamente hasta alcanzar su implementación. Por otro lado, el uso de técnicas de Design Thinking en las fases iniciales garantiza un enfoque centrado en los usuarios, priorizando la empatía, la definición clara del problema y la generación de ideas aplicables. Asimismo, se utilizarán diagramas de clases, componentes, secuencias y flujo como instrumentos fundamentales para modelar la solución, aportando claridad a la arquitectura del sistema y facilitando la comunicación de la propuesta. En un



entorno académico, esta práctica fortalece la capacidad de planear, estructurar y documentar el desarrollo de software, elementos esenciales en el ámbito profesional. En conclusión, el proyecto no solo constituye una respuesta tecnológica a una necesidad real, sino que también es una oportunidad académica para integrar conocimientos de ingeniería de software, metodologías ágiles y diseño centrado en el usuario, con el propósito de entregar un producto funcional, escalable y con potencial de aplicarse en escenarios similares.

ALCANCE DEL PROYECTO

El proyecto consiste en el desarrollo de un sistema de software para la IPUC Alfonso López, cuyo propósito es apoyar a los líderes de los comités en la organización de cronogramas de servicio, mostrando la disponibilidad de cada servidor y evitando cruces de horarios. El alcance incluye el diseño de interfaces gráficas (Front-End), la implementación de la lógica de negocio modular (Back-End) y la construcción de una base de datos relacional local (Persistencia) que asegure almacenamiento confiable. También se contemplan prototipos, pruebas de usabilidad, documentación, despliegue en un entorno controlado y la posibilidad de exportar los cronogramas en formato Excel o imagen. El alcance no incluye la integración con sistemas externos, aplicaciones móviles ni servicios en la nube.

En un entorno real, el desarrollo requeriría analistas, diseñadores UX/UI, desarrolladores, DBA, testers y un project manager. Sin embargo, en este proyecto académico, todas estas funciones las asumiré personalmente. El sistema se implementará mediante módulos principales:

- **Gestión de usuarios/servidores:** registro y asignación a comités, **Gestión de comités y cronogramas:** creación, edición y consulta de horarios, **Visualización de disponibilidad:** identificación de cruces y tiempos libres. **Reportes y exportación:** descarga del cronograma en Excel o imagen.



Se utilizarán herramientas de software libre: React para la interfaz, Node.js para el backend, MySQL/SQLite para la base de datos, GitHub para control de versiones y Figma para prototipado. El tiempo estimado es de 16 semanas, considerando análisis, diseño, desarrollo, pruebas y despliegue. Los entregables incluyen: prototipos de baja y alta fidelidad, diagramas de análisis y diseño (casos de uso, clases, arquitectura y flujo), base de datos relacional implementada, aplicación funcional con al menos tres módulos, resultados de pruebas y un video demostrativo de la solución. En cuanto a requerimientos funcionales, el sistema deberá gestionar usuarios, comités y cronogramas, mostrar la disponibilidad de los servidores, generar reportes y permitir exportar los cronogramas. Los requerimientos no funcionales incluyen facilidad de uso, persistencia local de datos, integridad de la información y tiempos de respuesta adecuados.

Finalmente, el proyecto se desarrollará con las restricciones propias de un entorno académico: recursos humanos limitados (un solo desarrollador), infraestructura local y uso exclusivo de tecnologías gratuitas.



PLANTEAMIENTO DEL PROBLEMA

Falta de gestión en cronogramas vs disponibilidad de usuarios(servidores)

¿Cómo se gestiona una aplicación que centraliza y valida la disponibilidad de cada miembro?

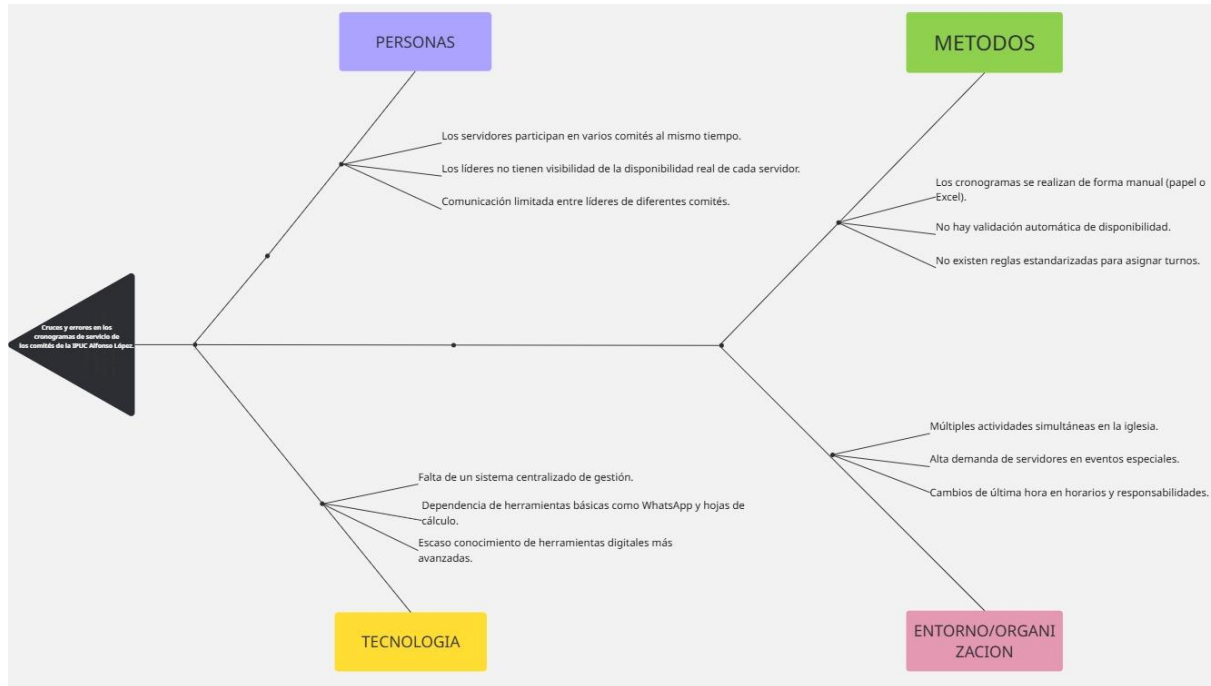


Ilustración 1 Diagrama Causa Efecto

[VER DIAGRAMA](#)

CONTEXTUALIZACIÓN

Para poder contextualizar la necesidad, usamos metodologías como Design Thinking y un mapa de empatía vea ilustración I:



Ilustración 2. Mapa de empatía

VER MAPA

Con ayuda del mapa anterior, pudimos contextualizar la necesidad: Los líderes de los comités de la Iglesia Pentecostal Unida de Colombia en la sede de Alfonso López Bogotá, en la localidad de Usme; tienen dificultades para organizar cronogramas de servicio ya que los servidores o miembros participan en diferentes comités o departamentos y no existe un sistema que muestre su disponibilidad. Por ende, necesitan una forma de visualizar la ocupación de cada servidor para evitar cruces de horarios.

OBJETIVO GENERAL

Desarrollar e implementar una solución de software que le permita a los líderes de los comités de la IPUC Alfonso López organizar cronogramas de actividades de manera eficiente, mostrando la disponibilidad de cada servidor para evitar cruces de horarios y mejorar la coordinación, además que permita exportar en imagen o en Excel dicho cronograma para compartirlo.

OBJETIVOS ESPECÍFICOS

- Realizar la contextualización de la necesidad, planteamiento del problema, definición de objetivos, alcance, metodología y matriz de riesgos.
- Ejecutar el levantamiento de información, identificación de stakeholders y definición de requerimientos funcionales y no funcionales (RQF–RQNF) mediante historias de usuario.
- Elaborar diagramas de flujo de solución, casos de uso, diagramas de clases, componentes/arquitectura y secuencias que representen el funcionamiento y estructura del sistema.
- Diseñar prototipos de baja y alta fidelidad, el sitemap y validar la navegabilidad del MVP mediante pruebas de usabilidad.
- Implementar la solución de software en sus tres componentes principales:
 - **Front-End:** interfaces gráficas que muestren la disponibilidad de los servidores y permitan organizar cronogramas.
 - **Back-End:** procesos y funcionalidades modulares que gestionen horarios, validen cruces y generen reportes.
 - **Persistencia:** almacenamiento de la información en una base de datos local (SQLite, MySQL local) que asegure integridad y consistencia.



- Desarrollar al menos tres módulos funcionales de acuerdo con los RQF (ejemplo: gestión de usuarios/servidores, gestión de comités y cronogramas, visualización de disponibilidad).
- Construir la base de datos relacional y su arquitectura para garantizar la persistencia y trazabilidad de la información.
- Realizar pruebas unitarias, de integración, de caja blanca y caja negra para garantizar la calidad del software.
- Documentar los resultados de pruebas de usabilidad, validando que la solución responde a los requerimientos funcionales definidos.
- Desplegar la aplicación en un entorno de producción controlado, asegurando la validación de los RQF y la satisfacción de los objetivos planteados.

METODOLOGÍA

Considero que con Kanban puedo tener todo en orden en un tablero —lo que falta, lo que está en proceso y lo que ya quedó listo— y eso me da claridad total. Además, uso Trello para llevarlo organizado, pero lo esencial está en asegurar que cada tarjeta este bien definida en cada una de las tarjetitas y ver cómo fluye el trabajo.

Un enfoque centrado en las personas

Con los fundamentos que nos han dado en la universidad, he aprendido que no se puede empezar programando de buenas a primeras. Por eso, antes de escribir código me apoyé en la metodología Design Thinking y al igual que los demás apliqué tres fases clave:

1. Empatizar → me permitió entender qué sienten y necesitan los líderes de comité cuando intentan cuadrar cronogramas.
2. Definir → ayudó a poner en palabras claras el problema: la desorganización y los choques de horarios.



3. Idear → fue el espacio para soltar propuestas sin miedo, desde papelitos hasta diagramas y posibles soluciones.

Este proyecto lo veo como armar un rompecabezas: primero entender las piezas, luego mirar la imagen de la caja, y al final ensamblarlo con calma.

Por otra parte, el proyecto lo dividiremos en etapas simples pero conectadas:

- Análisis y levantamiento de información → entrevistas, mapas de empatía y recolección de documentos (fotos de cronogramas viejos, notas de los líderes, etc.).
- Diseño → prototipos en Figma, diagramas en Miro, y varias pruebas de “¿qué pasaría si...?”.
- Desarrollo → un módulo funcional, programado en HTML, CSS y JavaScript, con almacenamiento en LocalStorage para simular cómo funcionaría en la vida real.
- Implementación → pruebas directas con los usuarios, ajustes sobre la marcha y entrega del producto final.

Cabe resaltar, que como estoy solo en el proyecto, básicamente me pongo varios sombreros:

- Analista → recojo la información y defino el problema.
- Diseñador → creo prototipos y pienso en la experiencia del usuario.
- Desarrollador → hago que la idea cobre vida en código.
- Tester → pruebo, ajusto y vuelvo a probar hasta que funcione bien.

En cuanto a las herramientas que utilizare en este proyecto se incluirán:

- Trello/Jira → tablero Kanban para organizar tareas.
- Figma y Miro → para diseñar y visualizar ideas.



- VSCode, HTML, CSS, JS y LocalStorage → para el desarrollo.
- WhatsApp/Teams → comunicación con usuarios o revisores.

La forma en que desarrollare esto, será cada semana haciendo un corte de avances. La idea no es esperar al final, sino entregar pedacitos que ya se puedan mostrar. Durante las 16 semanas de trabajo, voy revisando: “esto se hizo, esto sigue, esto falta”.

Control y seguimiento

El tablero Kanban será mi brújula. Cada tarea se mueve y eso me da la foto clara del progreso. Además, cada tanto pondré el prototipo frente a los usuarios para recoger reacciones. Si algo no cuadra, se ajusta de inmediato.

Porque al final, este proyecto no se trata solo de código o diseño, sino de darle tranquilidad a los líderes de comité que hoy se enredan cuadrando horarios.

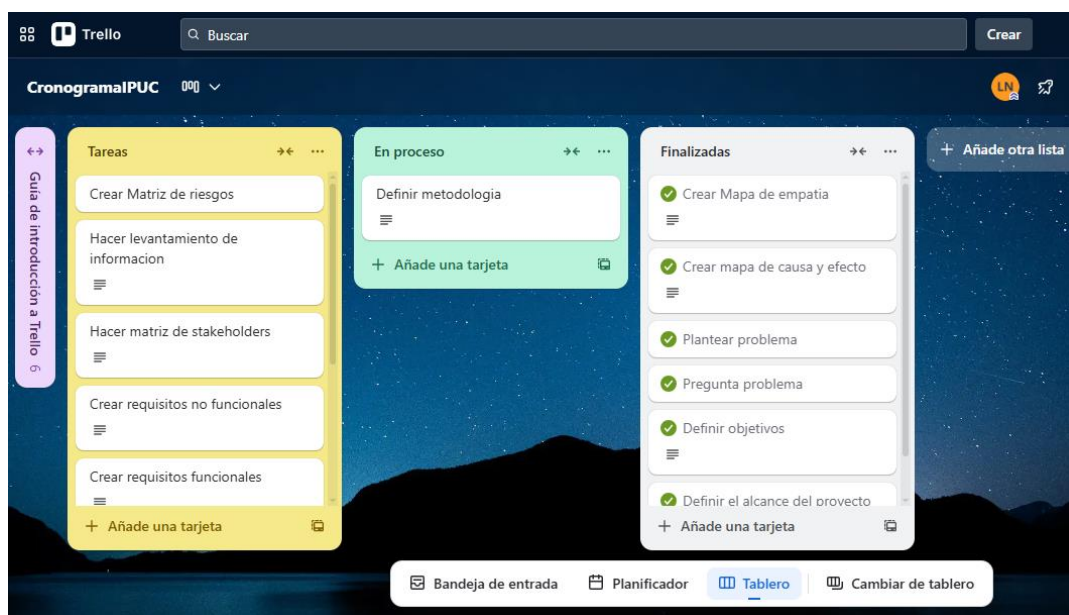


Ilustración 3 Tablero Trello

[Ver tablero](#)

MATRIZ DE RIESGOS

Tabla 1 Matriz de riesgos

Riesgo	Descripción	Probabilidad	Impacto	Nivel	Plan de mitigación
Fallas en la autenticación de usuarios (acceso no autorizado)	Vulnerabilidades en login que permitan accesos indebidos.	Medio	Alto	Alto	Cifrar contraseñas, controles de sesión, políticas de contraseña, pruebas de seguridad (penetration testing básico).
Errores en la visualización de disponibilidad (datos inconsistentes)	Cronogramas muestran asignaciones erróneas o desincronizadas.	Medio	Alto	Alto	Validaciones en backend, pruebas unitarias/integración, transacciones en BD y locks lógicos para asignaciones.
Pérdida de datos en la persistencia (corrupción BD/local)	Corrupción o pérdida de registros por fallo físico o lógico.	Bajo	Alto	Medio	Backups automáticos regulares, copia de seguridad antes de operaciones críticas, pruebas de restauración.
Baja escalabilidad al aumentar usuarios/eventos	Rendimiento degradado cuando crecen registros o simultaneidad.	Bajo	Medio	Bajo	Consultas optimizadas, índices en BD, paginación y límites en consultas; plan de migración a servidor si hace falta.
Exportación con errores (Excel/imagen incompleta)	Archivos exportados no reflejan lo mostrado en la pantalla.	Medio	Medio	Medio	Tests de exportación con casos reales, plantillas controladas, validación pos-exportación.
Incompatibilidad en navegadores/dispositivos	Interfaz rota o pérdida de funcionalidad en ciertos navegadores/dispositivos.	Medio	Medio	Medio	Pruebas cross-browser, polyfills si es necesario, diseño responsivo y pruebas en móviles.

Caídas por sobrecarga de consultas simultáneas	Servicio no responde ante picos de uso concurrente.	Bajo	Alto	Medio	Limitar accesos concurrentes, optimizar queries, caché para vistas no críticas, monitoreo de uso.
Interfaz poco intuitiva (baja adopción)	Usuarios no logran completar tareas sin apoyo.	Medio	Medio	Medio	Tests de usabilidad tempranos, ajustar flujos, documentación rápida y guía de uso.
Errores en sincronización de horarios (servidor en varios comités)	Solapamientos no detectados o reglas de asignación equivocadas.	Alto	Alto	Alto	Reglas de validación en creación/edición, bloqueo/alerta inmediata de conflictos, simulador de asignaciones antes de confirmar.

Fase de Análisis

LEVANTAMIENTO DE INFORMACIÓN:

Para la construcción del sistema de gestión de cronogramas de la IPUC Alfonso López se realizó un levantamiento de información basado en métodos interactivos y participativos, con el fin de comprender de manera precisa las necesidades de los líderes de los comités y la dinámica de los servidores.

Se aplicaron las siguientes técnicas:

- **Workshops de Historias de Usuario (HU):** Se llevaron a cabo sesiones colaborativas con los líderes de comité en las que, mediante dinámicas participativas, se definieron las principales historias de usuario que guiarían el desarrollo del sistema. Estos talleres permitieron identificar de manera práctica las funcionalidades



esenciales, como la visualización de disponibilidad, la gestión de servidores y la exportación de cronogramas. **Vea imagen 1** en sección de anexos.

- **Entrevistas semiestructuradas:** Se realizaron conversaciones directas con líderes de la iglesia, lo que permitió recopilar percepciones, problemas actuales y expectativas sobre el sistema.

Se emplearon además los siguientes instrumentos de recolección:

- **Fotografías:** Se documentaron los cronogramas manuales existentes (en papel y tableros físicos), lo que permitió comprender la forma tradicional de organización y los puntos críticos de confusión o cruce de horarios. Vea sección de anexos imagen 2,3.
- **Documentos:** Se analizaron cronogramas previos, listados de servidores y reportes de participación en comités, los cuales sirvieron como insumo para modelar la base de datos y validar requerimientos funcionales. Vea sección de anexos imagen 4.

De esta manera, la información recolectada se integró en un repositorio central, que sirvió de insumo para la definición de Requisitos Funcionales y No Funcionales, así como para la creación de los primeros prototipos.



MAPA DE STAKEHOLDERS



Ilustración 4 Mapa de Stakeholders

[VER MAPA](#)

DIAGRAMA DE FLUJO DE SOLUCIÓN

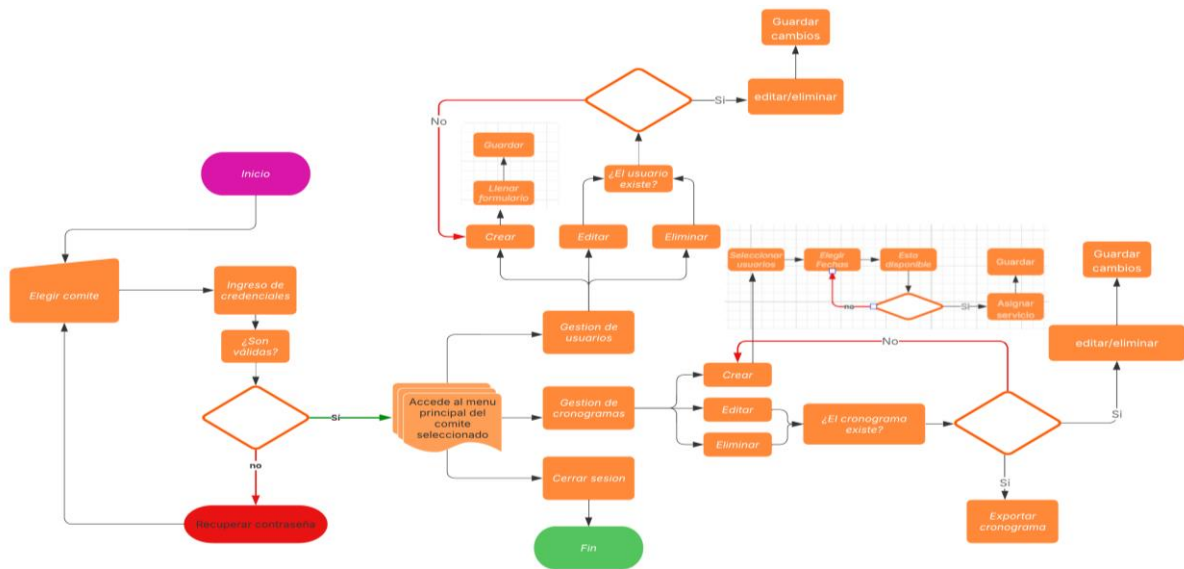


Ilustración 5 Diagrama de flujo

[Ver Diagrama](#)

REQUISITOS FUNCIONALES Y NO FUNCIONALES

Tabla 2 RF-001

RF-001	Gestionar de servidores	
Versión:	1.0	
Autor:	Luis Martínez	
Objetivo Asociado:	Cumplir con la funcionalidad 'Gestión de Servidores (CRUD)	
Requisito Asociado:	RF-001	
Descripción:	El sistema debe permitir la gestión completa de servidores (crear, leer, actualizar y eliminar).	
Precondición:	El líder debe estar autenticado en el sistema.	
Secuencia Normal:	Paso	Acción
	1.	El líder accede al módulo de gestión de servidores.
	2.	Selecciona la acción deseada (crear, actualizar, eliminar, consultar)
	3.	El sistema ejecuta la acción y actualiza la base de datos.
Postcondición	La información de servidores queda almacenada de manera persistente.	
Excepciones	Intentar eliminar un servidor asignado a un cronograma activo, Ingreso de datos incompletos en la creación de un servidor.	
Importancia	Alta	
Comentarios	La gestión como administrador la hará un miembro de la iglesia que comprenda el sistema	

Tabla 3 RF-002

RF-002	Exportar Cronogramas	
Versión:	1.0	
Autor:	Luis Martínez	
Objetivo Asociado:	Permitir a los líderes exportar el cronograma una vez hechos todos los cambios.	
Requisito Asociado:	RF-002	
Descripción:	El sistema debe permitir exportar el cronograma en formato Excel y como imagen una vez se haya modificado.	
Precondición:	El líder debe haber generado previamente un cronograma.	
Secuencia Normal:	Paso	Acción
	1.	El líder selecciona la opción 'Exportar cronograma'.
	2.	Elige el formato de exportación (Excel o Imagen).
	3.	El sistema genera y descarga el archivo.
Postcondición	El cronograma queda disponible en un archivo externo.	
Excepciones	El cronograma no tiene actividades registradas, error en permisos de descarga en el navegador.	
Importancia	Media	
Comentarios	Excel recomendado para edición, imagen para impresión rápida	



Tabla 4 RF-003

RF-003	Gestionar Cronogramas	
Versión:	1.0	
Autor:	Luis Martínez	
Objetivo Asociado:	Gestionar cronogramas de servicio sin conflictos.	
Requisito Asociado:	HU6, HU7, HU8, HU9	
Descripción:	El sistema debe permitir la creación, edición, eliminación y visualización de cronogramas.	
Precondición:	El líder debe estar autenticado en el sistema.	
Secuencia Normal:	Paso	Acción
	1.	El líder accede al módulo de cronogramas.
	2.	Selecciona la acción deseada (crear, modificar, eliminar, consultar).
	3.	El sistema ejecuta la acción y actualiza la base de datos.
Postcondición	Los cronogramas quedan guardados de manera persistente.	
Excepciones	Intentar asignar un servidor ya ocupado en la misma fecha/hora, error de conexión con la base de datos.	
Importancia	Alta	
Comentarios	Es el núcleo del sistema.	

Tabla 5 RF-004

RF-004	Autenticación de usuarios	
Versión:	1.0	
Autor:	Luis Martínez	
Objetivo Asociado:	Facilitar la organización de los servidores en los comités evitando cruces de horarios.	
Requisito Asociado:	HU1	
Descripción:	El sistema debe autenticar a los líderes mediante usuario y contraseña.	
Precondición:	El líder debe tener credenciales válidas.	
Secuencia Normal:	Paso	Acción
	1.	El líder ingresa al módulo de autenticación.
	2.	Ingresa sus credenciales, y el sistema valida la información.
	3.	Si es correcto, se concede acceso.
Postcondición	El líder inicia sesión en el sistema	
Excepciones	Credenciales invalidas, intento de acceso no autorizado.	
Importancia	Alta	
Comentarios	Básico para garantizar la seguridad.	



Tabla 6 RF-005

RF-005	Validar datos	
Versión:	1.0	
Autor:	Luis Martínez	
Objetivo Asociado:	Validar datos ingresados en formularios.	
Requisito Asociado:	HU2, HU6	
Descripción:	El sistema debe verificar que todos los campos obligatorios estén completos.	
Precondición:	El líder debe estar en un formulario de creación/edición	
Secuencia Normal:	Paso	Acción
	1.	El usuario llena el formulario.
	2.	El sistema valida los campos obligatorios.
	3.	Si todo está correcto, permite continuar.
Postcondición	Los datos correctos se guardan en la base de datos.	
Excepciones	Campos vacíos, formato de datos incorrecto.	
Importancia	Media	
Comentarios	Mejora la calidad de los datos.	

REQUISITOS NO FUNCIONALES:

Tabla 7 RNF-001

RNF-001	Rendimiento del sistema
Categoría	Rendimiento
Métrica	El sistema debe cargar y mostrar el cronograma en menos de 3 segundos para hasta 100 registros de servidores.

Tabla 8 RNF-002

RNF-002	Disponibilidad académica
Categoría	Disponibilidad
Métrica	El sistema debe estar disponible el 95% del tiempo en entornos de prueba.

Tabla 9 RNF-003

RNF-003	Seguridad de acceso
Categoría	Seguridad
Métrica	Solo los líderes de comité autenticados podrán acceder; se debe validar credenciales en el 100% de los intentos de inicio de sesión.



Tabla 10 RNF-004

RNF-004	Escalabilidad básica
Categoría	Escalabilidad
Métrica	El sistema debe soportar al menos 50 usuarios registrados y 200 eventos de cronograma sin pérdida de rendimiento.

Tabla 11 RNF-005

RNF-005	Usabilidad en prototipos
Categoría	Usabilidad
Métrica	En pruebas de navegabilidad, al menos el 80% de los líderes deben poder crear un cronograma sin requerir asistencia.

Tabla 12 RNF-006

RNF-006	Portabilidad
Categoría	Funcionalidad extendida / Usabilidad
Métrica	El sistema debe exportar cronogramas en Excel e imagen con un 100% de fidelidad respecto a lo visualizado en pantalla.

Tabla 13 RNF-007

RNF-007	Exportación de cronogramas
Categoría	Funcionalidad extendida / Usabilidad
Métrica	El sistema debe exportar cronogramas en Excel e imagen con un 100% de fidelidad respecto a lo visualizado en pantalla.

Tabla 14 RNF-008

RNF-008	Mantenibilidad del código
Categoría	Mantenibilidad
Métrica	El código debe estar documentado con al menos un 70% de comentarios explicativos y cumplir con estándares básicos de buenas prácticas.



Historias de usuario del sistema

☐ **HU 1 - Crear cronogramas**

+ Añadir

Como líder de comité quiero **crear cronogramas de servicio** para los servidores, de modo que pueda organizar turnos de manera clara y sin cruces de horario.

☒ **Criterios de aceptacion**

Eliminar

☐ Permite asignar fechas, horas y servidores.

☐ Valida que no existan conflictos de horario.

☐ El cronograma se guarda y queda disponible para consulta.

☒ **Escenario1. Creacion basica**

Eliminar

0%

☐ Se genera cronograma sin conflictos.

Ilustración 6 HU-1: Crear cronogramas

[VER TABLERO EN TRELLO](#)

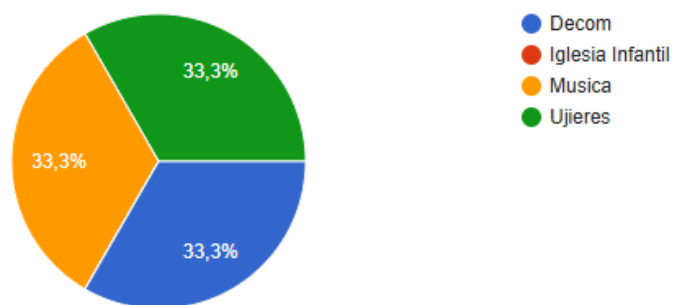
ANEXOS:



Anexo A Workshop con líderes

Selecciona el departamento que representas:

3 respuestas



Anexo B Encuesta para crear mapa de empatía

[VER ENCUESTA](#)

CONCLUSIONES

Este proyecto ayudó a entender mejor los problemas que enfrentan los líderes de comité en la IPUC Alfonso López cuando intentan organizar los turnos de los servidores. Para resolver eso, se creará un sistema que permita registrar la disponibilidad, armar cronogramas visuales y detectar conflictos de horarios de forma automática, lo que resulta mucho más práctico que hacerlo todo a mano. Se usó Design Thinking y herramientas ágiles como Kanban para planear el desarrollo de manera ordenada, siempre pensando en quienes van a usarlo. También se integraron plataformas como Jira y GitHub para manejar tareas, versiones y facilitar el trabajo en equipo. En esta solución no solo hace más fácil la gestión de turnos, sino que también deja un camino abierto para seguir mejorando la organización de los comités en el futuro.



BIBLIOGRAFÍA

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., ...

Thomas, D. (2001). *Manifesto for Agile Software Development*. Agile Alliance.

<https://agilemanifesto.org/>

Cohn, M. (2004). *User stories applied: For agile software development*. Addison-Wesley Professional.

Pressman, R. S., & Maxim, B. R. (2014). *Ingeniería del software: Un enfoque práctico* (7ª ed.). McGraw-Hill.

Sommerville, I. (2011). *Ingeniería de software* (9ª ed.). Pearson Educación.

