

Sistema de información para la gestión de horarios académicos en la escuela de ingeniería de sistemas e informática – Universidad Industrial de Santander

Daniel Fernando La Rotta Peña,

Juan David Morantes Vergara

Trabajo de Grado para Optar al Título de Ingeniero de Sistemas

Director

Juan Ramon Pernalete Maldonado

Ingeniero de Sistemas

Tutor

Luis Carlos Gomez Florez

Ingeniero de Sistemas

Universidad Industrial de Santander

Facultad de Ingenierías Fisicomecánicas

Escuela de Ingeniería de Sistemas e Informática

Bucaramanga

2026

Dedicatoria

Daniel Fernando La Rotta Peña

Dedico este proyecto, con profundo agradecimiento y cariño a mi madre Belsy Peña Ortiz, abuela Odolonida Ortiz. Gracias por apoyarme en todo el ciclo de mi formación académica y profesional. Gracias por enseñarme el amor de una madre y una abuela que luchan día a día por querer verme profesional y grande. Por sacrificarte desde que yo he nacido, para darme bienestar y darme siempre lo mejor. Este logro es tuyo Madre ya que siempre has querido ser UIS y este es tu momento de dejar de sacrificarse tanto por ti, ahora yo lo haré por ti. También le dedico este logro a mi novia Natalia Gualdron ya que me vio crecer en la UIS desde los inicios hasta ya poder graduarme, Gracias por acompañarme siempre en los peores momentos y los mejores. Este logro también lo comparto contigo por estar en todo este ciclo.

Juan David Morantes Vergara

Dedico este proyecto, con profundo amor y gratitud, a mi padre Francisco José Morantes Valderrama y a mi madre Ruth Clemencia Vergara, por su apoyo incondicional a lo largo de toda mi vida. Gracias por hacer siempre todo lo posible para que pudiera alcanzar este logro, por creer en mí y por inculcarme valores y principios desde mi infancia que hoy me permiten formarme como profesional.

A mi hermano Andrés Felipe Morantes Vergara, por apoyarme sin condiciones durante los momentos difíciles de mi vida universitaria, siendo un pilar fundamental en este proceso.

A mi hermano Iván Adolfo Morantes, a Eliana Morantes y a su esposo Jaime Alberto Herrera Upegui, quienes siempre han estado presentes cuando los he necesitado, brindándome su apoyo incondicional y acompañamiento constante.

Finalmente, a mi abuelo Elías Vergara (Q.E.P.D.), quien me vio iniciar este camino. Aunque hoy no se encuentra físicamente, sé que desde el cielo me acompaña y se siente orgulloso de este logro. Este triunfo también es para él.

Agradecimientos

Al ingeniero Luis Carlos el cual nos brindo su mano para poder realizar este proyecto. Al ingeniero Pernalette el cual nos conoce desde que estábamos en 5to semestre y siempre ha creído en nosotros como profesionales a futuro, gracias por ser una persona tan agradable, grata y con mucho compromiso, por profesores como usted siempre van a ser recordados en nuestros corazones y ser un ejemplo en nuestra vida profesional. a mi compañero de trabajo los cuales aportaron en la realización de este proyecto.

A cada uno de los profesores que me orientaron y brindaron el conocimiento para forjar las bases de mi carrera profesional.

Contenido

Introducción	12
1. Planteamiento y justificación del problema	16
2. Objetivos	20
2.1 Objetivo general	20
2.2 Objetivos específicos	20
3. Marco de referencia	21
3.1 Marco Teórico	21
3.1.1 Sistemas de Información	21
3.1.2 Gestión de Recursos Académicos	22
3.1.3 Optimización y Programación de Tareas	22
3.2 Marco conceptual	23
3.3 Estado del arte	24
3.3.1 Antecedentes	24
4. Metodología	26
4.1 Fase de Inicio	28
4.2 Fase de elaboración	29
4.3 Fase de construcción	30
4.4 Fase de transición	31
5. Resultados	32
5.1 Fase de Inicio	32
5.1.1 Contexto y gestión del proyecto	32
5.1.1.1 Vista general del proyecto.	33
5.1.1.2 Organización del proyecto.	36
5.1.2 Análisis de la organización y del proceso actual	41
5.1.2.1 Descripción del proceso anterior.	41
5.1.4 Entregables de la fase de inicio	43
5.1.5 Identificación y caracterización de actores	44
5.1.6 Modelamiento del proceso actual	45
5.1.6.1 Recolección de disponibilidad - proceso antiguo.	46
5.1.6.2 Asignación de horarios – proceso antiguo	50
5.1.7 Problemáticas identificadas	52
5.1.8 Delimitación y estructuración funcional del sistema	55
5.1.8.1 Gestión de áreas y asignaturas.	56
5.1.8.2 Configuración y acceso al sistema	58
5.1.8.3 Disponibilidad y horarios	60
5.1.8.4 Gestión de usuarios	62

	6
5.1.8.5 Diagramas de Casos de Uso.	63
5.1.8.5.1 Asignación de profesores	63
5.1.8.5.2 Creación de Áreas	65
5.1.8.5.3 Creación de Asignaturas	67
5.1.8.5.4 Creación de Grupos	68
5.1.8.5.5 Creación de Semestres	69
5.1.8.5.6 Creación de Usuarios	70
5.1.8.5.7 Disponibilidad	73
5.1.9 Modelamiento del sistema propuesto	74
5.1.9.1 Solicitud y gestión de disponibilidad – proceso rediseñado	76
5.1.9.2 Asignación de horarios – proceso rediseñado	79
5.1.10 Análisis comparativo del proceso	81
5.1.11 Elicitación inicial de requerimientos	82
5.1.11.1 Requerimientos No Funcionales.	85
5.1.11.2 Requerimientos Complementarios.	86
5.2 Fase de Elaboración	86
5.2.1 Consolidación de requerimientos	86
5.2.1.1 Requerimientos funcionales	87
5.2.1.2 Requerimientos no funcionales	87
5.2.1.3 Requerimientos complementarios	88
5.2.2 Diseño del sistema	88
5.2.2.1 Modelo de datos	88
5.2.2.2 Diseño arquitectónico del sistema	89
5.2.2.3 Prototipos visuales	89
5.2.3 Análisis de riesgos	90
5.2.4 Entregables de la fase de elaboración	90
5.2.5 Consolidación y Especificación de Requerimientos	92
5.2.5.1 Requerimientos funcionales.	93
5.2.5.2 Historias de Usuario.	98
5.2.5.3 Requerimientos No Funcionales.	98
5.2.6 Diseño Arquitectónico del Sistema SIGHA	101
5.2.6.1 Diseño Arquitectónico del Sistema SIGHA	101
5.2.6.2 Diagrama de Componentes	104
5.2.6.3 Diagrama de Capas del Backend	107
5.2.7 Modelo de Datos del Sistema	109
5.2.7.1 Modelo de Datos Conceptual y Lógico.	111
5.2.7.2 Modelo de Datos del Sistema SIGHA.	114
5.2.7.2.1 Relaciones entre tablas.	118

	7
5.2.8 Análisis y Gestión de Riesgos Técnicos	124
5.2.9 Prototipos visuales	128
5.3 Fase de construcción	132
5.3.1 Organización del desarrollo	133
5.3.2 Implementación de módulos del sistema	134
5.3.2.1 Módulo de recolección de disponibilidad docente.	135
5.3.2.2 Módulo de gestión de horarios académicos.	135
5.3.2.3 Validaciones automáticas y reglas de negocio.	136
5.3.3 Pruebas técnicas del sistema	137
5.3.3.1 Pruebas unitarias	137
5.3.3.2 Pruebas de integración	137
5.3.4 Entregables de la fase de construcción	138
5.3.5 Entregables de la fase de construcción	138
5.3.6 Organización del desarrollo	140
5.3.7 Implementación de módulos del sistema	143
5.3.8 Validaciones automáticas implementadas	145
5.3.9 Pruebas técnicas del sistema	146
5.3.9.1 Pruebas unitarias.	146
5.3.9.2 Pruebas de integración.	148
5.3.10 Resumen del producto de software	149
5.4 Resultados de la Fase de Transición	153
5.4.1 Preparación para la puesta en marcha	153
5.4.2 Capacitación de usuarios	154
5.4.3 Pruebas de aceptación del sistema	154
5.4.4 Documentación del sistema	155
5.4.5 Entregables de la fase de transición	156
5.4.6 Despliegue del sistema	157
5.4.7 Capacitación de usuarios	161
5.4.8 Pruebas de aceptación del sistema	162
5.4.9 Análisis de costos y beneficios del sistema	165
5.4.9.1 Costos reales del proyecto.	165
5.4.9.2 Comparativa de tiempos.	166
5.4.9.3 Beneficios de la implantación del sistema.	167
6. Conclusiones	169
7. Recomendaciones	171
7.1 Ampliación del sistema a otras escuelas y sistemas institucionales	171
7.2 Obtención de permisos para acceso a información en tiempo real	171
7.3 Carga automática y gestión inteligente de información de los estudiantes	172

7.4 Sistema de recomendación de horarios y asignaturas con IA	8
Apéndices	172
Referencias bibliográficas	175

Lista de figuras

1.	Diagrama de Casos de Uso - proceso antiguo	33
2.	Recolección de disponibilidad - proceso antiguo	46
3.	Asignación de horarios - Proceso antiguo	49
4.	Diagrama de contexto - Gestion de areas y asignaturas	55
5.	Diagrama de contexto - Configuración y Acceso	57
6.	Diagrama de contexto - Disponibilidad y horarios	59
7.	Diagrama de contexto - Gestión de usuarios	61
8.	Diagrama de caso de uso – Asignación de profesores	63
9.	Diagrama de caso de uso – Creación de áreas	64
10.	Diagrama de caso de uso – Creación de Asignaturas	66
11.	Diagrama de caso de uso – Creación de Grupos	67
12.	Diagrama de caso de uso – Creación de Semestres	69
13.	Diagrama de caso de uso – Creación de Usuarios	70
14.	Diagrama de caso de uso – Disponibilidad	72
15.	Solicitud de disponibilidad – proceso propuesto	75
16.	Asignación de grupos y horarios – proceso propuesto	78
17.	Formato de requerimientos funcionales - Historias de usuario	98
18.	Diagrama general cliente–servidor	109
19.	Diagrama de Componentes	
	111	
20.	Diagrama de Capas del Backend – SIGHA	114
21.	Modelo de Datos Conceptual y Lógico del Sistema SIGHA	113
22.	Modelo de Datos del Sistema SIGHA	115
23.	Prototipo – Inicio de sesión	128
24.	Prototipo – Dashboard	129
25.	Prototipo – Registro de disponibilidad docente	130
26.	Prototipo – Gestión de horarios	130
27.	Prototipo – Gestión de grupos	131
28.	Mapa de navegación del sistema	131
29.	Tablero Kanban para la asignación de tareas	134
30.	Cronograma	142
31.	Diagrama de arquitectura Frontend	150
32.	Diagrama de arquitectura Backend	152
33.	Arquitectura de implantación	160

Lista de tablas

Definición de conceptos clave del proyecto	25
Actores involucrados	34
Roles y funciones dentro del equipo de trabajo	39
Estimación de costos	41
Entregables de la fase de inicio	44
Actores involucrados en el proyecto	46
Problemáticas identificadas	53
Requerimientos funcionales del sistema	84
Requerimientos no funcionales del sistema	85
Entregables: fase de elaboración	90
Requerimientos Funcionales - Descripción	94
Diccionario de datos. esquema: core, tabla: user	123
Diccionario de datos. esquema: core, tabla: roles	123
Diccionario de datos. esquema: core, tabla: programs	124
Diccionario de datos. esquema: core, tabla: area	124
Diccionario de datos. esquema: core, tabla: semester	125
Diccionario de datos. esquema: core, tabla: availability	125
Diccionario de datos. esquema: core, tabla: user_area	126
Diccionario de datos. esquema: core, tabla: user_rol_program	126
Diccionario de datos. esquema: ing_sistemas, tabla: subject	127
Diccionario de datos. esquema: ing_sistemas, tabla: groups	127
Diccionario de datos. esquema: ing_sistemas, tabla: schedule	128
Plan de Gestión de Riesgos Técnicos del Sistema SIGHA	131
Entregable: Fase de Construcción	142
Iteraciones de desarrollo	143
Descripción de Historias de usuarios	146
Validaciones automáticas implementadas	148
Pruebas unitarias	149
Pruebas de integración	151
Entregables: fase de transición	158
Contenido de manual de usuario (videotutoriales)	163
Pruebas de aceptación con usuarios finales	165
Costos del proyecto.	167
Comparativa de tiempos – Proceso manual vs. SIGHA	168

Lista de Apéndices

A. Requerimientos Funcionales de SIGHA.	177
B. Video tutoriales por tema particular	178

Glosario

- Elicitación: proceso sistemático de investigar, descubrir y obtener requisitos, necesidades o información de los interesados en un proyecto.

Resumen

Título: Sistema de información para la gestión de horarios académicos en la Escuela de Ingeniería de Sistemas e Informática de la Universidad Industrial de Santander.

Autores: Daniel Fernando La Rotta Peña, Juan David Morantes Vergara

Palabras Clave: Disponibilidad, Sistema de información, Gestión de horarios

Descripción: El presente trabajo se ubica en el contexto de un proyecto de ingeniería orientado al desarrollo de un Sistema de Información para la Gestión de Horarios Académicos en la Escuela de Ingeniería de Sistemas e Informática de la Universidad Industrial de Santander (UIS). Su objetivo es mejorar la gestión de horarios mediante la automatización de tareas que tradicionalmente se realizaban con hojas de cálculo, trasladándose a un entorno web que reduce imprecisiones administrativas, disminuye errores de digitación y mejora la experiencia de docentes y personal administrativo.

Para el desarrollo se aplicó la metodología RUP (Rational Unified Process) junto con Kanban, lo cual permitió analizar el proceso actual, definir requerimientos funcionales, no funcionales y complementarios, establecer la arquitectura del sistema y construir una solución acorde a las necesidades identificadas. Este enfoque facilitó el seguimiento del avance, la organización de actividades y la entrega incremental de funcionalidades.

Como producto final, el sistema constituye una solución robusta para la gestión de procesos académicos administrativos. Centraliza información, facilita la planeación, asignación y consulta de horarios, y disminuye la carga operativa asociada a tareas repetitivas. Además, puede integrarse con otros sistemas institucionales, lo que permite proyectar su uso hacia otros programas académicos de la universidad.

* Trabajo de Grado

** Facultad de Fisicomecánicas. Escuela de Ingeniería de Sistemas e Informática. Director: Juan Ramon Pernaleté Maldonado. Ingeniero de Sistemas. Tutor: Luis Carlos Gomez Florez. Ingeniero de Sistemas

Abstract

Title: Information System for the Management of Academic Timetables in the School of Systems and Informatics Engineering at the Universidad Industrial de Santander.

Authors: Daniel Fernando La Rotta Peña, Juan David Morantes Vergara

Keywords: Availability, Information System, Timetable Management

Description: This work is situated within the context of an engineering project aimed at developing an Information System for Academic Schedule Management at the School of Systems and Computer Engineering of the Universidad Industrial de Santander (UIS). Its objective is to improve schedule management by automating tasks that were traditionally performed using spreadsheets, transferring them to a web-based environment that reduces administrative inaccuracies, minimizes data entry errors, and enhances the experience of faculty members and administrative staff.

For development, the RUP (Rational Unified Process) methodology was applied together with the Kanban framework, which made it possible to analyze the current process, define functional, non-functional, and complementary requirements, establish the system architecture, and build a solution aligned with the identified needs. This approach facilitated progress tracking, activity organization, and incremental delivery of functionalities with quality control.

As the final product, the system constitutes a robust solution for managing academic administrative processes within the School. It centralizes information, facilitates planning, assignment, and consultation of schedules, and reduces the operational burden associated with repetitive tasks. In addition, the application can be integrated with other institutional systems, enabling its use across other academic programs at the university and contributing to the modernization of administrative processes.

* Degree Work

^[1] Faculty of Physical-Mechanical Engineering. School of Systems and Computer Engineering. Director: Juan Ramon Pernaleté Maldonado. Systems Engineer. Tutor: Luis Carlos Gomez Florez. Systems Engineer.

Introducción

El proceso de gestión de horarios académicos es una tarea esencial en las instituciones de educación superior, ya que permite planificar y coordinar la asignación de profesores y espacios de acuerdo con las necesidades de los programas académicos, constituyéndose como un componente fundamental en la administración eficiente de los recursos educativos (Cabrera, 2023). En la Escuela de Ingeniería de Sistemas e Informática de la Universidad Industrial de Santander (UIS), este proceso se desarrolla actualmente en tres fases: recolección de disponibilidad de los profesores, consolidación de la información y asignación de horarios.

En la primera fase, la disponibilidad horaria de los profesores se obtiene principalmente mediante mensajería instantánea, lo que ocasiona dificultades para registrar y actualizar la información de manera precisa. La segunda fase consiste en recopilar los datos en una hoja de cálculo, un procedimiento manual que requiere tiempo y puede generar inconsistencias en los registros. Finalmente, en la tercera fase se ajustan los horarios a partir del archivo del semestre anterior, sin validaciones que aseguren la compatibilidad entre asignaturas, profesores y grupos, lo que puede producir conflictos de horario o asignaciones incorrectas, evidenciando las limitaciones de los procesos manuales en la gestión de información académica (Laudon & Laudon, 2021).

Este método, basado en herramientas no especializadas, ha demostrado ser ineficiente y propenso a incidencias administrativas, afectando la productividad del personal administrativo y retrasando otras actividades críticas en la Escuela de Ingeniería de Sistemas e Informática. Además, la falta de automatización limita la trazabilidad y la capacidad de respuesta ante modificaciones en las disponibilidades o en la carga académica, lo cual evidencia la necesidad de implementar sistemas de información que permitan gestionar datos de forma eficiente y apoyar la toma de decisiones organizacionales (Laudon & Laudon, 2021).

Ante esta situación, el presente proyecto desarrolla un sistema de información web que optimiza la gestión de horarios académicos en la Escuela de Ingeniería de Sistemas e Informática. El sistema integra funcionalidades como la visualización de disponibilidades por profesor, la gestión centralizada de horarios, validaciones automáticas de asignaciones, un panel de control con información consolidada y la posibilidad de copiar y ajustar horarios de semestres anteriores, en concordancia con el uso de sistemas de información como herramientas clave para mejorar la gestión académica y administrativa (Laudon & Laudon, 2021).

El objetivo principal de esta aplicación web es reducir los tiempos de gestión, minimizar las inconsistencias administrativas y mejorar la experiencia tanto de los profesores como del personal encargado del proceso de administración de horarios académicos. El sistema está diseñado con un enfoque centrado en el usuario, empleando tecnologías web modernas que garanticen accesibilidad, fiabilidad y eficiencia. Con su implementación, se transforma el

modelo actual en un proceso más ágil, interactivo y sostenible, sentando las bases para futuras mejoras en la gestión académica institucional, en línea con el uso de sistemas de información para optimizar procesos organizacionales y apoyar la toma de decisiones (O'Brien & Marakas, 2021).

Este trabajo se enmarca dentro de la metodología Rational Unified Process (RUP) y el marco de trabajo Kanban, siguiendo sus directrices para el desarrollo de software. RUP establece un enfoque estructurado basado en cuatro fases fundamentales, las cuales se organizan de la siguiente manera (Ortega, 2022):

Fase de inicio.

Se describen los elementos correspondientes a la Fase I (Inicio) de la metodología RUP, enfocándose en el análisis de la situación actual en la gestión de horarios académicos. En este contexto, la fase de inicio permitió establecer las bases del proyecto mediante la comprensión del problema, el análisis del proceso actual de gestión de horarios académicos y la identificación de los actores involucrados.

Durante esta etapa se definió el alcance inicial del sistema, se identificaron las principales necesidades institucionales y se documentaron las problemáticas asociadas al manejo manual de los horarios, lo cual facilitó la formulación de los requisitos iniciales y la viabilidad del sistema propuesto.

Fase de elaboración.

La fase de elaboración del proyecto tuvo como objetivo principal consolidar los requerimientos previamente identificados y establecer la arquitectura del sistema destinado a soportar la gestión de horarios académicos. Durante esta etapa se detallaron exhaustivamente las especificaciones funcionales y no funcionales necesarias para garantizar el desarrollo de un sistema que respondiera a las necesidades del usuario final, fuera técnicamente viable y estuviera alineado con los procesos académicos de la Escuela de Ingeniería de Sistemas e Informática.

Asimismo, en el marco de esta fase se llevó a cabo un análisis riguroso de los riesgos técnicos y organizacionales asociados al desarrollo del sistema. Como resultado, se elaboró un plan técnico detallado que estableció las directrices fundamentales para la fase de construcción, en concordancia con los principios del Rational Unified Process para la definición de arquitectura y gestión de riesgos (Ortega, 2022).

Fase de construcción.

En esta fase se materializó el desarrollo e implementación del sistema conforme a los requisitos y la arquitectura establecidos, utilizando ciclos iterativos para la creación y validación de funcionalidades. Se priorizaron los requerimientos funcionales, tales como la gestión de disponibilidad docente, la asignación de asignaturas y validación automatizada de horarios, asegurando que el sistema cumpliera con los objetivos planteados.

Asimismo, se realizaron pruebas continuas en cada iteración para garantizar la calidad del software y detectar oportunidades de mejora. La integración de módulos se llevó a cabo de forma progresiva, incorporando módulos clave para la visualización interactiva de horarios y la validación de conflictos de solapamiento, lo que permitió una transición fluida hacia un sistema integral y funcional.

Fase de transición.

Esta fase tiene como propósito implementar el sistema en el entorno operativo real de la Escuela. Durante este período se capacitó al personal administrativo y docente en el manejo del software, y se ejecutaron pruebas de aceptación para verificar el cumplimiento de los requerimientos del usuario final y estándares administrativos. Los entregables incluyen los resultados de las pruebas realizadas con usuarios finales y la documentación completa (manual de usuario y guía de administración) para facilitar la adopción del sistema.

1. Planteamiento y justificación del problema

La Escuela de Ingeniería de Sistemas e informática de la Universidad Industrial de Santander previo al inicio de cada semestre académico requiere organizar horarios académicos como parte fundamental para los procesos administrativos, lo que permite registrar la información en el sistema académico de la Universidad Industrial de Santander, En este proceso se definen los horarios para las asignaturas de la escuela y se crean diferentes grupos con denominaciones alfanuméricas como (A1, B2) para organizar cuántos estudiantes se tendrá en

cada clase y quién será el docente asignado. Cada materia debe estar asignada a un docente y además se deben tener en cuenta la capacidad de los salones y la cantidad de clases que cada docente puede desarrollar. Lo anterior resalta la importancia de una gestión adecuada de los recursos educativos —humanos y académicos— que permita un uso eficiente de los espacios y del talento docente, condiciones indispensables para el correcto desarrollo de la actividad académica (Cabrera, 2023).

Diversas instituciones académicas y organizaciones de alto rendimiento ya han adoptado soluciones informáticas para la programación de horarios, reconociendo que la información es uno de los activos más valiosos. En el ámbito universitario, esta visión se relaciona con la importancia de los sistemas de información como pilar de la transformación digital, donde se promueve la toma de decisiones basadas en datos fiables y la mejora de la gobernanza institucional (Briceño & Wheeler, 2024). Un sistema integrado de gestión de horarios permite centralizar datos, evitar errores por duplicación y hacer cambios fácilmente (cómo ajustar la carga de un profesor, la prioridad de un grupo o la disponibilidad de salones) sin tener que rehacer todo desde cero.

Con el objetivo de llevar a cabo este proceso, la escuela destina varios funcionarios (administrativos y docentes) a la tarea de gestionar en una hoja de cálculo la asignación de grupos, asignaturas, cupos y profesores, así como recopilar la información de los docentes para cada periodo académico y su disponibilidad horaria de clases. El personal al depender de esta

hoja de cálculo se ven forzados a realizar validaciones manuales de alto rigor para evitar solapamientos de docentes en sus distintos grupos (por ejemplo, inspeccionar celdas en busca de coincidencias de franjas), y en muchos casos volver a contactar a los profesores para ajustar o confirmar nuevas franjas horarias. Durante este proceso, cada modificación implica rehacer parte de la hoja de cálculo y validar exhaustivamente todas las asignaciones anteriores, lo cual resulta altamente problemático por el tiempo invertido, la carga repetitiva de trabajo y la alta probabilidad de cometer errores.

Partiendo de la problemática establecida anteriormente, se plantean las siguientes interrogantes del proyecto:

- ¿Qué tecnologías y modelos de desarrollo pueden emplearse para generar un sistema capaz de gestionar la asignación de horarios en la Escuela de Ingeniería de Sistemas e Informática?
- ¿Qué arquitecturas de software y patrones de diseño permitirían incorporar validaciones automáticas de solapamientos?

El correcto manejo de los horarios académicos en la Escuela de Ingeniería de Sistemas e informática es fundamental para garantizar la cobertura equilibrada de asignaturas y el cumplimiento de la carga docente establecida por los reglamentos internos de la Universidad Industrial de Santander. La asignación ineficiente de horarios no solo afecta la carga docente, sino también la adecuada utilización de recursos educativos como aulas y laboratorios, cuya gestión estratégica es vital para garantizar calidad y eficacia institucional (Cabrera, 2023).

Actualmente, la metodología basada en hojas de cálculo impone un gran esfuerzo al personal administrativo y docente, que dedica horas a recopilar disponibilidades, validar solapamientos y reconfigurar continuamente las asignaciones. El uso excesivo de estas herramientas, aunque prácticas en muchos escenarios, genera riesgos importantes como errores en la introducción de datos, problemas de control de versiones, falta de escalabilidad y pérdida de confianza en la información, lo que incrementa la probabilidad de errores humanos y ocasiona retrabajos constantes (Laudon & Laudon, 2021).

Finalmente, frente a la problemática planteada y la creciente demanda de calidad y eficiencia en la gestión académica, se hizo necesario desarrollar un software especializado para la asignación de horarios académicos, el cual aporta competitividad a la Escuela de Ingeniería de Sistemas e Informática, asegurando escalabilidad y reducción de costos operativos a medida que crecen el número de asignaturas, grupos y docentes. Además, en consonancia con las tendencias internacionales de transformación digital en el ámbito universitario, los sistemas de información se reconocen como aliados estratégicos para fortalecer la gobernanza institucional y apoyar la toma de decisiones basadas en datos fiables (Briceño & Wheeler, 2024). Incorporando validaciones automáticas, el sistema elimina parcialmente las duplicidades y los solapamientos, agiliza el proceso de recolección de datos y libera al equipo de tareas repetitivas. Esto contribuye a mejorar la experiencia de los docentes —al poder registrar y consultar su disponibilidad en tiempo real— y permite al personal administrativo enfocar sus esfuerzos en actividades estratégicas de planificación curricular.

2. Objetivos

2.1 Objetivo general

Desarrollar un sistema de información para la gestión de horarios académicos en la Escuela de Ingeniería de Sistemas e Informática de la Universidad Industrial de Santander, suficientemente flexible para adaptarse a las necesidades y situaciones imprevistas del proceso educativo, los recursos e infraestructura disponible, garantizando las intensidades horarias mínimas para los estudiantes y la jornada laboral de los docentes.

2.2 Objetivos específicos

- **Analizar** el proceso actual en la gestión de horarios de la Escuela de Ingeniería de Sistemas e Informática, mediante el levantamiento de requerimientos funcionales, no funcionales y complementarios.
- **Establecer** una arquitectura software de acuerdo a las especificaciones de los requerimientos identificados.
- **Construir** un aplicativo en un entorno web partiendo de la arquitectura seleccionada, que incluya los módulos de recolección de disponibilidad y de generación de horarios, a partir de algoritmos de optimización y validaciones automáticas de solapamientos, garantizando la coherencia y la trazabilidad de cada asignación horaria.
- **Implementar** la puesta en marcha de la aplicación web para garantizar el buen funcionamiento del sistema de información

3. Marco de referencia

3.1 Marco Teórico

3.1.1 *Sistemas de Información*

Un **sistema de información** es un conjunto de elementos interrelacionados —personas, procesos, datos, hardware y software— que trabajan de manera conjunta para recopilar, procesar, almacenar y distribuir información, con el fin de apoyar la gestión y la toma de decisiones dentro de una organización (Laudon & Laudon, 2021).

En las instituciones de educación superior, los sistemas de información han adquirido un papel estratégico, ya que centralizan datos académicos (estudiantes, docentes, aulas, asignaturas) y permiten transformarlos en información útil para planear, administrar y evaluar la actividad académica. Su implementación contribuye a mejorar la eficiencia y reducir la carga manual en procesos complejos como la asignación de horarios.

3.1.2 *Gestión de Recursos Académicos*

La **gestión de recursos educativos** consiste en administrar de manera eficiente los recursos humanos (docentes, administrativos), físicos (aulas, laboratorios) y temporales (horas de clase, cargas laborales) para asegurar que los objetivos académicos se cumplan (Cabrera, 2023).

En el caso de los horarios académicos, este proceso implica múltiples restricciones: disponibilidad docente, capacidad de los salones, cumplimiento de reglamentos internos y ausencia de solapamientos. Por ello, la gestión manual basada en hojas de cálculo resulta poco práctica y altamente propensa a errores. Un sistema de información con algoritmos de validación y optimización permite automatizar estas tareas y garantizar mayor equidad y coherencia en las asignaciones.

3.1.3 Optimización y Programación de Tareas

El problema de la asignación de horarios académicos pertenece al campo de la optimización combinatoria, y en particular a los problemas de programación de tareas (scheduling). Estos problemas buscan encontrar la mejor manera de asignar un conjunto de tareas (clases) a recursos limitados (docentes, aulas, franjas horarias) bajo múltiples restricciones.

Los principales enfoques incluyen:

- **Métodos exactos:** como la programación lineal entera, que garantizan una solución óptima, aunque resultan costosos en escenarios con gran número de variables.
- **Métodos heurísticos:** algoritmos aproximados que generan soluciones suficientemente buenas en tiempos razonables, como los algoritmos genéticos, búsqueda tabú

o recocido simulado.

- **Métodos híbridos:** combinan técnicas exactas y heurísticas para balancear precisión y eficiencia.

Este marco teórico evidencia por qué la planificación manual mediante hojas de cálculo resulta ineficiente a medida que aumenta la cantidad de asignaturas, docentes y salones: la complejidad crece de manera exponencial y solo puede abordarse eficazmente con un sistema de información que incorpore técnicas de **optimización y programación de tareas**.

3.2 Marco conceptual

A continuación, se definen los conceptos clave empleados en el proyecto:

Tabla 1.

Definición de conceptos clave del proyecto

<i>Término</i>	<i>Definición</i>
<i>Sistema de Información</i>	<i>Conjunto de componentes interrelacionados que procesan información para apoyar la gestión y la toma de decisiones</i>
<i>Horario Académico</i>	<i>Plan estructurado que asigna docentes, aulas y franjas horarias a las asignaturas de un programa académico</i>
<i>Solapamiento</i>	<i>Conflicto en el que un mismo docente, grupo o aula es asignado simultáneamente a más de una actividad académica</i>

<i>Disponibilidad Docente</i>	<i>Conjunto de franjas horarias declaradas por un profesor durante las cuales está habilitado para dictar clases</i>
<i>Optimización</i>	<i>Proceso mediante el cual se busca la mejor solución posible a un problema bajo condiciones o restricciones específicas.</i>
<i>Scheduling</i>	<i>Rama de la optimización que estudia la programación de tareas en recursos limitados.</i>
<i>Iteración</i>	<i>Ciclo de desarrollo en el que se planifica, diseña, implementa y prueba un subconjunto del sistema.</i>
<i>Kanban</i>	<i>Método visual de gestión de tareas que permite monitorear y mejorar el flujo de trabajo.</i>
<i>RUP</i>	<i>Metodología de desarrollo de software que organiza el proceso en fases iterativas e incrementales.</i>

3.3 Estado del arte

3.3.1 Antecedentes

Rojas & Hernández (2025), quienes desarrollaron el Trabajo de Grado Titulado: “Automatización Robótica de Procesos (RPA) para el seguimiento analítico de activos y generación de reportes de excedentes en clientes AGPE de la Electrificadora de Santander (ESSA)” como requisito para obtener el título de ingeniero en sistemas en la Universidad Industrial de Santander. Este proyecto implementó soluciones de automatización mediante RPA, apoyadas en herramientas como Python y Microsoft Power Platform, logrando

transformar procedimientos manuales en flujos eficientes y reduciendo significativamente tiempos, errores humanos y costos operativos.

Lo relevante para este trabajo es la demostración de cómo la automatización de procesos manuales en entornos empresariales permite optimizar la gestión de información, lo cual respalda la necesidad de un sistema que automatice la asignación de horarios académicos.

Quispe (2021), quienes desarrollaron el Trabajo de Grado Titulado: “Sistema de información web para mejorar la gestión de procesos de adquisiciones y contrataciones del Programa de Apoyo al Desarrollo Rural Andino” como requisito para obtener el magister scientiarum en informática en la Universidad Nacional Del Altiplano. En esta tesis se diseñó e implementó un sistema de información web bajo la metodología ágil SCRUM, orientado a mejorar la gestión administrativa de procesos de adquisiciones y contrataciones. El resultado fue una mayor efectividad y satisfacción de los usuarios finales.

El aporte a este proyecto radica en la evidencia de que un sistema web, desarrollado con metodologías ágiles, puede optimizar procesos administrativos tradicionalmente manuales y mejorar la experiencia de los usuarios.

Solgorre & Renne (2023) quienes desarrollaron el Trabajo de Grado Titulado: “Sistema web basado en Scrum y XP para el proceso de gestión de docentes en el Instituto Gilda Ballivián Rosado, Universidad Autónoma del Perú” como requisito para obtener el título de ingeniero en sistemas en la Universidad Industrial de Santander. Este trabajo propuso un

sistema web que sustituye archivos físicos y hojas de cálculo en la gestión docente. Se utilizaron metodologías ágiles como Scrum y XP, logrando optimizar el registro y la elaboración de horarios, así como mejorar la disponibilidad de información para directivos y docentes.

La lección clave para este proyecto es la viabilidad de reemplazar herramientas manuales (como hojas de cálculo) por sistemas automatizados, evidenciando mejoras concretas en eficiencia y disponibilidad de datos.

4. Metodología

El desarrollo del Sistema de Información para la Gestión de Horarios Académicos (SIGHA) se enmarca dentro de una metodología formal de ingeniería de software, orientada a garantizar la calidad del producto, la trazabilidad de los requerimientos y la alineación del sistema con las necesidades institucionales de la Escuela de Ingeniería de Sistemas e Informática de la Universidad Industrial de Santander.

Para tal fin, se adopta el Rational Unified Process (RUP) como metodología base, una estructura que continúa siendo referente en la industria tecnológica y es fundamental en el ecosistema de soluciones de grandes fabricantes como IBM para el desarrollo de software robusto y escalable. Este enfoque se complementa con el marco de trabajo Kanban para la gestión operativa de las actividades. Esta combinación permite organizar el proyecto en fases

estructuradas, desarrolladas de manera iterativa e incremental, facilitando la identificación temprana de riesgos, la validación progresiva de los resultados y la entrega controlada de los artefactos del sistema bajo estándares modernos de calidad.

RUP proporciona una estructura sólida para el desarrollo de software académico, al definir claramente objetivos, actividades y entregables por fase, mientras que Kanban apoya el seguimiento del avance del proyecto mediante la visualización del flujo de trabajo, la priorización de tareas y la identificación de bloqueos. La combinación de ambos enfoques favorece la colaboración entre los participantes del proyecto y mejora la eficiencia en la ejecución de las actividades planteadas.

El desarrollo de SIGHA se organiza en las siguientes fases: Inicio, Elaboración, Construcción y Transición, las cuales se describen a continuación junto con sus objetivos, actividades y productos esperados.

4.1 Fase de Inicio

La fase de inicio tiene como propósito establecer una comprensión clara del problema a resolver, definir el alcance del sistema y analizar el contexto institucional en el cual se desarrolla el proceso de gestión de horarios académicos en la Escuela de Ingeniería de Sistemas e Informática de la Universidad Industrial de Santander.

Durante esta fase se analizaron las condiciones actuales bajo las cuales se gestionaba la asignación de horarios académicos, identificando las principales limitaciones del proceso manual y las necesidades de los actores involucrados.

Actividades:

- Se realizaron entrevistas con el personal administrativo y docente.
- Se documentaron los procesos actuales de gestión de horarios
- Se delimitaron los alcances del sistema.

Entregables:

- **Requerimientos Iniciales:** documento que recopila los principales requerimientos funcionales y no funcionales, como la recolección de disponibilidades docentes, la gestión de asignación de grupos y las validaciones automáticas de solapamientos.
- **Diagramas de Caso de Uso:** representaciones visuales de las interacciones entre los actores principales (Docente, Profesional, Coordinador Académico, Director) y el sistema.
- **Diagrama BPMN del Proceso Actual:** modelo del flujo de actividades actuales que realiza el equipo administrativo, evidenciando las deficiencias del proceso manual.
- **Diagrama de Estados:** documento que muestra las transiciones posibles en el ciclo de vida de un horario (propuesto, en revisión, validado, rechazado, publicado).

4.2 Fase de elaboración

La fase de elaboración tiene como propósito consolidar los resultados obtenidos en la fase de inicio y definir de manera detallada la solución propuesta para el sistema SIGHA. En esta etapa se profundiza en la especificación de los requerimientos del sistema y se establece el diseño arquitectónico que guiará su implementación en la fase de construcción.

Actividades

- Identificar los riesgos críticos en la seguridad del sistema
- Construir un plan técnico que servirá como guía para el desarrollo posterior.
- Asegurar la viabilidad técnica del proyecto

Entregables:

- **Prototipos Visuales:** diseños preliminares de las interfaces gráficas que faciliten la validación de usabilidad con los usuarios finales.
- **Modelo de Datos:** diseño lógico de la base de datos, que incluye las principales entidades (docente, asignatura, grupo, disponibilidad y espacio físico) y sus relaciones.
- **Documento de Diseño Arquitectónico (DDA):** descripción de la estructura técnica del sistema, detallando los módulos principales (disponibilidad, generación de horarios, validación de solapamientos) y sus interacciones.
- **Plan de Gestión de Riesgos:** identificación y propuesta de mitigación de riesgos técnicos y organizacionales asociados al desarrollo del sistema

4.3 Fase de construcción

La fase de construcción tiene como propósito implementar el sistema SIGHA de acuerdo con los requerimientos y el diseño definidos en la fase de elaboración. En esta etapa se desarrollaron los módulos principales del sistema de manera iterativa e incremental, asegurando la correcta aplicación de las decisiones arquitectónicas y de las reglas de negocio establecidas previamente.

Actividades:

- Recolectar la disponibilidad docente.
- Se construirá la validación automática de solapamientos.
- Se implementará la arquitectura diseñada.
- Se ejecutarán pruebas unitarias para verificar el funcionamiento de los componentes.

Entregables:

- **Código Fuente Modular:** implementación del sistema bajo un enfoque modular que favorezca la mantenibilidad y escalabilidad.

4.4 Fase de transición

La fase de transición tiene como propósito poner el sistema SIGHA en funcionamiento dentro del entorno real de la Escuela de Ingeniería de Sistemas e Informática de la Universidad Industrial de Santander, asegurando que el sistema sea comprendido, aceptado y utilizado correctamente por los usuarios finales. En esta etapa se valida que el sistema implementado

cumpla con los requerimientos institucionales y que se encuentre en condiciones adecuadas para apoyar el proceso de gestión de horarios académicos.

Actividades:

- Se capacitará al equipo administrativo y docente en el uso del software
- Se realizarán pruebas de aceptación para validar que cumpla con los requerimientos institucionales.

Entregables:

- Pruebas de Usuario Final: resultados de pruebas de aceptación con docentes y personal administrativo.
- Manual de Usuario y Guía de Administración: documentación para capacitar a los actores en el uso y administración del sistema.

5. Resultados

El presente capítulo expone los resultados obtenidos durante el desarrollo del Sistema de Información para la Gestión de Horarios Académicos (SIGHA), organizado de acuerdo con las fases establecidas en la metodología Rational Unified Process (RUP).

Cada sección presenta no solo los productos generados, sino el impacto que estos tuvieron en la estructuración formal del proceso académico y en la delimitación funcional del sistema propuesto.

5.1 Fase de Inicio

El desarrollo de esta etapa permitió establecer las bases conceptuales y organizacionales necesarias para el diseño del sistema SIGHA, garantizando que las fases posteriores se construyeran sobre un entendimiento sólido del problema institucional.

5.1.1 Contexto y gestión del proyecto

El Sistema de Información para la Gestión de Horarios Académicos (SIGHA) se desarrolló en el marco de un trabajo de grado bajo la modalidad de práctica empresarial, orientado a dar solución a una necesidad institucional real de la Escuela de Ingeniería de Sistemas e Informática. La gestión del proyecto se enfocó en coordinar las actividades de análisis inicial, definir responsabilidades y asegurar la alineación entre los objetivos académicos del trabajo de grado y las necesidades operativas de la Escuela.

En esta etapa se identificaron los actores que participan directa o indirectamente en el proceso de gestión de horarios académicos, así como su rol dentro del contexto institucional.

Tabla 2.

Actores involucrados en el proceso

<i>Actores</i>	<i>Rol</i>	<i>Responsabilidad</i>
----------------	------------	------------------------

<i>Director de escuela</i>	<i>Máxima autoridad académica y administrativa de la escuela</i>	<i>Dirigir la toma de decisiones sobre los horarios de la escuela</i>
<i>Coordinador Académico</i>	<i>Responsable de coordinación académica y operativa de la escuela.</i>	<i>Coordinar horarios acordes a las necesidades de la escuela.</i>
<i>Profesor</i>	<i>Miembro del cuerpo docente de la escuela</i>	<i>Llenar el formulario de Disponibilidad y Dictar la clase programada</i>

5.1.1.1 Vista general del proyecto. La gestión de horarios académicos en la Escuela de Ingeniería de Sistemas e Informática se realiza actualmente mediante un proceso manual apoyado principalmente en hojas de cálculo y comunicación informal con los docentes. Este enfoque requiere que el personal administrativo recopile las disponibilidades horarias, consolide la información y valide manualmente posibles conflictos entre docentes y grupos académicos.

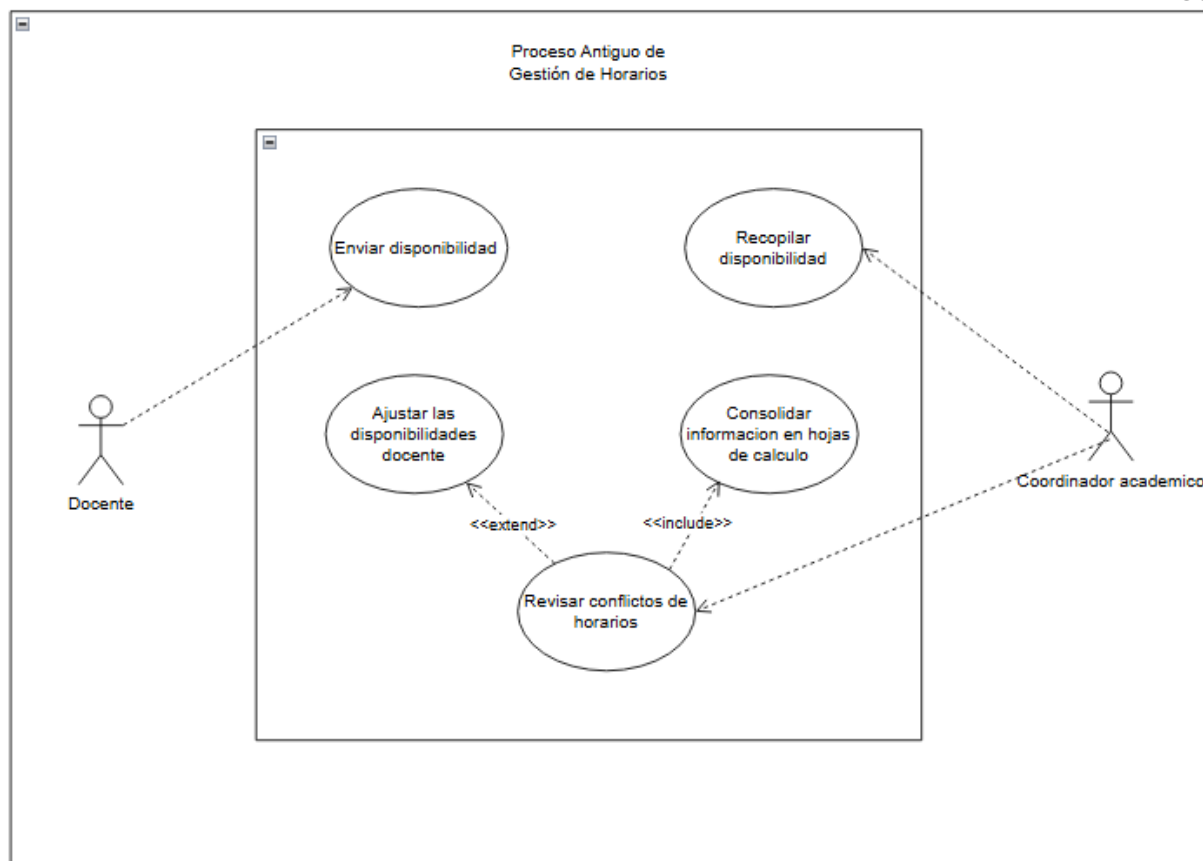
Este proceso presenta dificultades significativas, tales como la alta probabilidad de errores, la falta de trazabilidad de los cambios realizados y el elevado tiempo invertido en la validación de solapamientos. Ante esta situación, surge la necesidad de desarrollar un sistema de información especializado que permita centralizar la gestión de horarios académicos y automatizar las validaciones asociadas al proceso.

SIGHA se concibe como una aplicación web de apoyo a la planeación académica, orientada a facilitar la recolección de disponibilidades docentes, la gestión de grupos académicos y la validación automática de conflictos de horario, sin reemplazar los sistemas académicos centrales de la Universidad Industrial de Santander.

Como se observa en la Figura 1, el proceso actual de gestión de horarios se caracteriza por la intervención directa de los actores en actividades manuales, donde el docente registra y ajusta su disponibilidad, mientras que el coordinador académico se encarga de recopilar la información, consolidarla en hojas de cálculo y validar los posibles conflictos de horario.

Figura 1.

Diagrama de Casos de Uso - proceso antiguo



El diagrama de casos de uso - proceso antiguo representa la interacción entre los actores involucrados en el proceso actual de gestión de horarios académicos. En este modelo, el docente participa mediante el registro y ajuste de su disponibilidad horaria, mientras que el coordinador académico realiza la recopilación de la información, su consolidación y la validación manual de conflictos.

Asimismo, se evidencia que la validación de conflictos depende directamente de la consolidación previa de la información, y que el ajuste de disponibilidades se ejecuta como una actividad posterior cuando se detectan inconsistencias. Este modelo permite identificar la alta

dependencia de actividades manuales y la ausencia de mecanismos automatizados, lo cual justifica la necesidad de implementar un sistema de información que optimice el proceso.

5.1.1.2 Organización del proyecto.

Participantes en el proyecto

El equipo de desarrollo del Sistema de Información para la Gestión de Horarios Académicos (SIGHA) está conformado por distintos actores que desempeñan funciones clave en el diseño, implementación y validación del sistema.

- **Autor:** Este proyecto está concebido como el desarrollo de un trabajo de grado bajo la modalidad de práctica empresarial para optar al título de Ingeniero de Sistemas, por tal motivo el autor ha de cubrir todos los requisitos académicos y de conocimiento para el abordaje del proyecto y cumplir el perfil del puesto de acción.
- **Director del proyecto y tutor académico:** Deberán poseer el título de pregrado en la especialidad de la carrera a optar por el alumno, y tener como mínimo una ubicación administrativa equivalente a profesor Instructor otorgada por la Universidad Industrial de Santander.
- **Director de escuela y coordinador académico:** Son los usuarios asignados que laboran dentro de la Escuela de Ingeniería de Sistemas e Informática. Su función es evaluar el sistema, realizar observaciones y definir los requerimientos necesarios para garantizar que cumpla con su propósito.

- **Docentes:** Son todos los profesores de la Escuela de Ingeniería de Sistemas e Informática a quienes está dirigido el módulo de disponibilidad del sistema. Registran sus franjas horarias disponibles para la asignación de clases.

Roles y responsabilidades

A continuación se describen las principales responsabilidades de cada uno de los roles en el equipo de desarrollo, de acuerdo a la siguiente **Tabla 3**.

Tabla 3.

Roles y funciones dentro del equipo de trabajo

<i>Puesto de acción</i>	<i>Roles</i>	<i>Responsabilidades</i>
	<i>Líder del proyecto</i>	<i>Dirigir y coordinar las interacciones entre los puestos de acción. Establecer las prácticas que aseguren la integridad y calidad de los artefactos del proyecto.</i>
<i>Tutor académico</i>	<i>Analista de calidad</i>	<i>Revisar todos los documentos que reflejan el avance del proyecto. Verificar que los objetivos del marco de desarrollo se cumplan.</i>
	<i>Mentor</i>	<i>Acompaña y apoya a los equipos de trabajo mediante revisiones de los artefactos.</i>

Hace recomendaciones de mejora durante todo el ciclo de vida del sistema.

	<i>Analista de producto</i>	<i>Dirigir el proceso de captura de requerimientos. Definir los actores y casos de uso y estructurar el modelo de casos de uso</i>
<i>Autores</i>	<i>Arquitecto de Software</i>	<i>Establecer la arquitectura del desarrollo y refinarla en cada iteración. Construir prototipos necesarios para probar aspectos técnicos. Definir los lineamientos generales del diseño y la implementación.</i>
	<i>Desarrollador</i>	<i>Codificar los componentes en lenguaje de alto nivel. Elaborar y ejecutar pruebas unitarias.</i>

Documentar y actualizar clases bajo control de configuración.

Sección de Personal Docente

Tester

Realizar las pruebas identificadas y definidas previamente.

Usar las instrucciones, métodos y herramientas ,necesarias

Personal de la escuela de ingeniería de sistemas

Usuario final

Asegurar que sus necesidades sean satisfechas por el proyecto.

Proporcionar retroalimentación sobre la funcionalidad del sistema

Estimaciones de Costos del Proyecto

Según lo establecido por la hora cátedra docente y la información que proporcionó el Director de proyecto Juan Pernaletе, se define el presupuesto del proyecto tomando en cuenta lo allí descrito basado en la modalidad de trabajo actual y los recursos disponibles.(J. Pernaletе, comunicación personal, 4 de octubre de 2025)

Tabla 4.*Estimación costos*

<i>Concepto</i>	<i>Responsable</i>	<i>Unidad</i>	<i>Cantidad</i>	<i>Valor unitario</i>	<i>Valor total</i>
Recursos humanos					
<i>Director del proyecto</i>	<i>Director</i>	<i>Hora</i>	<i>34</i>	<i>\$ 44.716/hora</i>	<i>\$ 1.520.345</i>
<i>Autor 1</i>	<i>Estudiante</i>	<i>Hora</i>	<i>170</i>	<i>\$ 16.000/hora</i>	<i>\$ 2.720.000</i>
<i>Autor 2</i>	<i>Estudiante</i>	<i>Hora</i>	<i>170</i>	<i>\$ 16.000/hora</i>	<i>\$ 2.720.000</i>
<i>Sub total</i>					<i>\$ 18.530.000</i>
Recursos tecnológicos					
<i>Alojamiento en la nube</i>	<i>Estudiante</i>	<i>Mes</i>	<i>4</i>	<i>\$ 66.250/més</i>	<i>\$ 265.000</i>
<i>Internet</i>	<i>Estudiante</i>	<i>Mes</i>	<i>4</i>	<i>\$ 70.000</i>	<i>\$ 280.000</i>
<i>Computador personal</i>	<i>Estudiante</i>	<i>Unidad</i>	<i>2</i>	<i>\$ 3.500.000</i>	<i>\$ 7.000.000</i>
<i>Sub total</i>					<i>\$ 7.525.000</i>
Gastos adicionales					
<i>Insumos</i>	<i>Estudiante</i>	<i>Imprevistos</i>			<i>\$130.000</i>

Presupuesto general	
<i>Concepto</i>	<i>Valor</i>
<i>Recursos humanos</i>	<i>\$ 6.960.000</i>
<i>Recursos tecnológicos</i>	<i>\$ 7.525.000</i>
<i>Gastos adicionales</i>	<i>\$130.000</i>
<i>Total</i>	<i>\$ 14.615.000</i>

Los costos asociados a los programadores fueron obtenidos de la plataforma (Glassdoor 2024), la cual provee estimaciones salariales basadas en datos reportados por empleados y empleadores. Esta información permitió establecer un rango referencial para el análisis de viabilidad económica del proyecto.

5.1.2 Análisis de la organización y del proceso actual

Con el fin de comprender el contexto organizacional en el cual se implementará el sistema SIGHA, se realizó un análisis del proceso actual de gestión de horarios académicos en la Escuela. Este análisis permitió identificar la forma en que se desarrollan las actividades actualmente y los puntos críticos del proceso que impactan su eficiencia.

5.1.2.1 Descripción del proceso anterior. En la Escuela de Ingeniería de Sistemas e Informática de la Universidad Industrial de Santander (UIS), el proceso de gestión de horarios académicos se desarrolla anteriormente mediante prácticas manuales descentralizadas, lo cual impacta negativamente la eficiencia operativa y la calidad de la información institucional. Este proceso inicia con la captura de la disponibilidad horaria de los docentes, la cual se obtiene por medio de comunicación directa —principalmente vía mensajería instantánea o correo electrónico— sin mecanismos formales de estructuración o control.

Posteriormente, el personal administrativo procede a consolidar la disponibilidad docente en hojas de cálculo, integrando datos dispersos y heterogéneos provenientes de distintas fuentes. Esta consolidación se realiza de forma manual, apoyándose en plantillas no estandarizadas y sin esquemas previos de normalización de datos, lo que incrementa la probabilidad de inconsistencias y duplicidades en los registros.

Una vez integrada la información, se realiza la fase de ajuste y construcción de los horarios académicos, utilizando como referencia los archivos de semestres anteriores. Este ajuste tampoco se basa en algoritmos de inferencia o validación computarizada: los administradores deben inspeccionar visualmente celdas en las hojas de cálculo para encontrar posibles solapamientos de actividades docentes, clases simultáneas en un mismo grupo o conflicto de franjas horarias entre profesores y asignaturas.

Finalmente, el proceso concluye con una validación manual exhaustiva para evitar solapamientos entre docentes, grupos académicos y recursos físicos (p. ej. aulas y laboratorios). Dichas verificaciones —carentes de mecanismos automáticos de detección de conflictos— implican revisiones visuales y frecuentes comunicaciones de retroalimentación con los docentes para ajustar disponibilidades o reasignar franjas horarias.

4.1.3 Ejes críticos de la fase de inicio

El desarrollo de la fase de inicio se orientó alrededor de un conjunto de ejes críticos que guiaron el análisis y la toma de decisiones durante esta etapa del proyecto. Estos ejes permiten enfocar los esfuerzos en los aspectos fundamentales del problema y asegurar la viabilidad del sistema propuesto.

Los ejes críticos considerados fueron los siguientes:

- Comprensión del problema institucional a partir del análisis del proceso actual de gestión de horarios académicos.
- Delimitación clara del alcance del sistema, evitando la inclusión de funcionalidades ajenas a los objetivos del proyecto.
- Identificación de los actores involucrados y de sus responsabilidades dentro del proceso.
- Evaluación de la viabilidad técnica y organizacional del desarrollo del sistema SIGHA.

5.1.4 Entregables de la fase de inicio

Como resultado del desarrollo de la fase de inicio se generaron diversos entregables que documentan el análisis realizado y constituyen insumos fundamentales para las fases posteriores del proyecto.

Tabla 5.

Entregables de la fase de inicio

Entregable	Descripción
Requerimientos funcionales y requerimientos no funcionales	<i>Documento que recopila los principales requisitos funcionales y no funcionales, como la recolección de disponibilidades docentes, la gestión de asignación de grupos y las validaciones automáticas de solapamientos.</i>
Diagramas de Caso de Uso	<i>Representaciones visuales de las interacciones entre los actores principales (Docente, Profesional, Coordinador Académico, Director) y el sistema.</i>
Diagrama BPMN del Proceso Actual	<i>modelo del flujo de actividades actuales que realiza el equipo administrativo, evidenciando las deficiencias del proceso manual.</i>
Diagrama de Estados	<i>Documento que muestra las transiciones posibles en el ciclo de vida de un horario (propuesto, en revisión, validado, rechazado, publicado).</i>

La fase de inicio permitió establecer una comprensión estructurada del problema institucional asociado a la gestión manual de horarios académicos en la Escuela de Ingeniería de Sistemas e Informática de la Universidad Industrial de Santander.

Como resultado de esta fase, se logró:

- Identificar formalmente los actores institucionales involucrados.
- Modelar el proceso actual (AS-IS).
- Rediseñar el proceso incorporando software especializado(TO-BE).
- Delimitar el alcance funcional preliminar del sistema.
- Identificar de manera estructurada los requerimientos iniciales.

5.1.5 Identificación y caracterización de actores

Se consolidó la caracterización de los actores que intervienen directa e indirectamente en el proceso de gestión de horarios académicos.

Tabla 6.

Caracterización de actores del sistema SIGHA

<i>Actores</i>	<i>Rol</i>	<i>Responsabilidad</i>	<i>Responsabilidad</i>
<i>Director de escuela</i>	<i>Máxima autoridad académica y administrativa de la escuela</i>	<i>Dirigir la toma de decisiones sobre los horarios de la escuela</i>	<i>Toma de decisiones estratégicas del sistema</i>

<i>Coordinador Académico</i>	<i>Responsable de coordinación académica y operativa de la escuela.</i>	<i>Coordinar horarios acordes a las necesidades de la escuela.</i>	<i>Supervisión del proceso de asignación</i>
<i>Profesor</i>	<i>Miembro del cuerpo docente de la escuela</i>	<i>Llenar el formulario de Disponibilidad y Dictar la clase programada</i>	<i>Registro de disponibilidad en el sistema</i>

Este resultado permitió delimitar claramente las responsabilidades institucionales y establecer las bases para el modelamiento funcional del sistema.

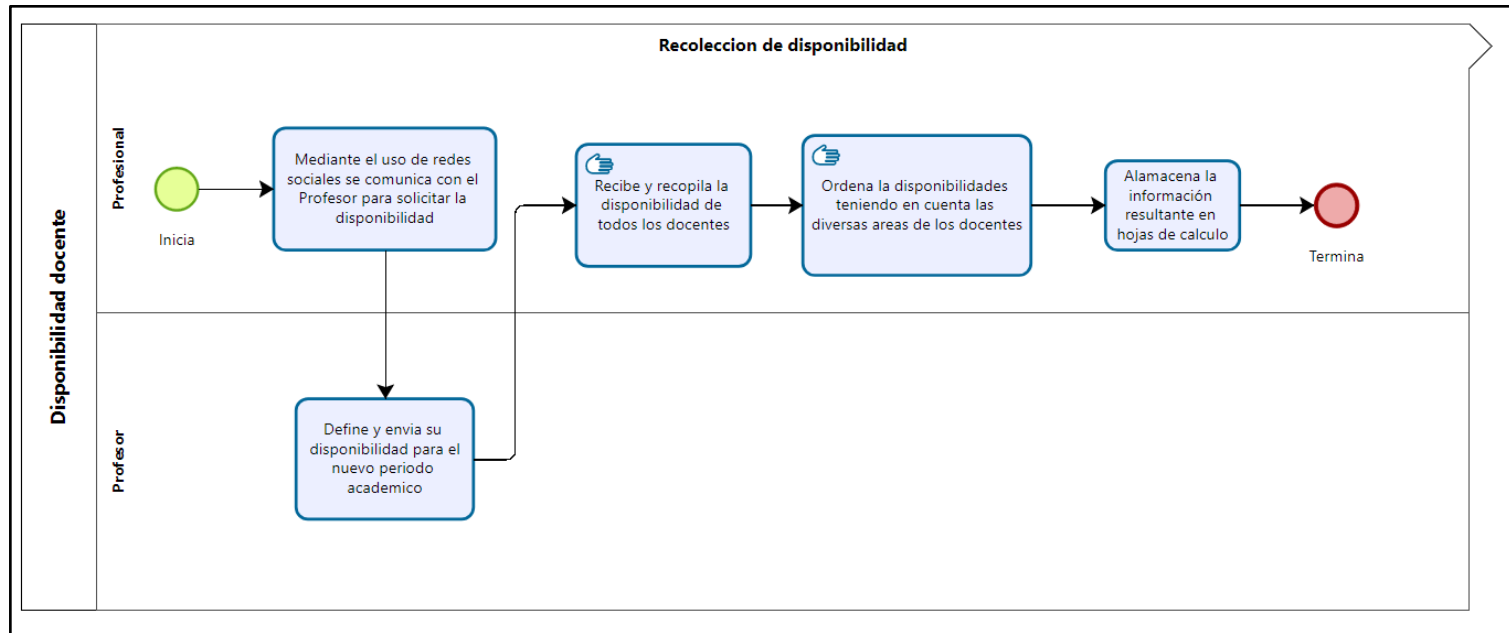
5.1.6 Modelamiento del proceso actual

Con el fin de comprender las deficiencias del proceso manual, se modelaron mediante diagramas BPMN las actividades desarrolladas antes de la implementación del sistema.

5.1.6.1 Recolección de disponibilidad - proceso antiguo. Se elaboró el modelo del proceso actual mediante los siguientes diagramas BPMN, en los cuales se permite visualizar las actividades manuales, los puntos de control y las dependencias operativas existentes.

Figura 2.

Recolección de disponibilidad - proceso antiguo



El modelamiento del proceso actual mediante BPMN permitió representar de forma estructurada la secuencia real de actividades involucradas en la recolección de disponibilidad docente antes de la implementación del sistema.

El análisis evidenció que esta actividad se desarrollaba de manera completamente manual y descentralizada. El proceso iniciaba con el contacto individual a cada docente por parte del personal administrativo, utilizando principalmente correo electrónico o mensajería instantánea como medio de comunicación. Este mecanismo implica que cada profesor enviará su disponibilidad en formatos no estandarizados, generando heterogeneidad en la información recibida.

Posteriormente, el personal encargado debía interpretar, organizar y normalizar los datos enviados, clasificándolos según áreas de conocimiento y asignaturas habilitadas para cada docente. Esta actividad añadía un nivel adicional de complejidad operativa, ya que requería validaciones informales basadas en conocimiento previo y experiencia administrativa.

El BPMN permitió identificar que la consolidación de las disponibilidades se realizaba manualmente en hojas de cálculo independientes, donde se transcriben y organizan las franjas horarias. Este procedimiento implicaba:

- Revisión individual de cada respuesta.
- Organización manual por áreas académicas.

- Actualización constante ante cambios o correcciones.
- Almacenamiento en archivos sin integración automática.

Como resultado del modelamiento, se evidenció que el proceso presentaba las siguientes debilidades estructurales:

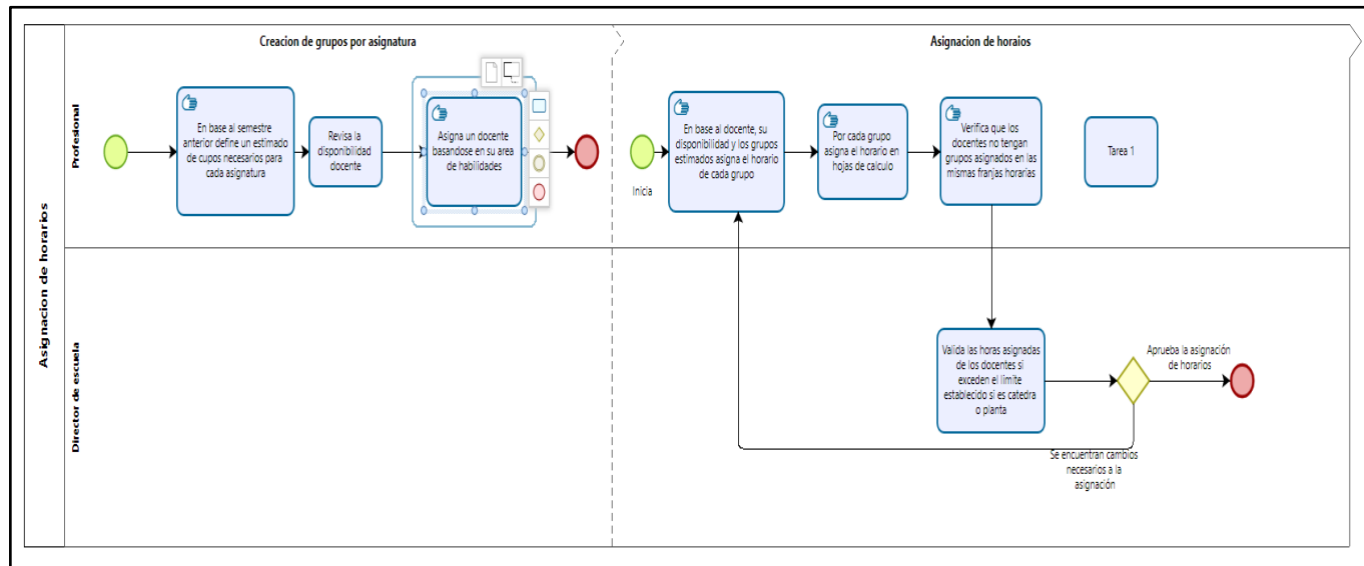
- Falta de estandarización en la captura de información.
- Alta dependencia de la interpretación de cada administrativo.
- Riesgo elevado de errores de digitación y transcripción.
- Ausencia de trazabilidad histórica de modificaciones.
- Duplicidad potencial de registros.

En términos organizacionales, el proceso dependía en gran medida de la precisión del personal administrativo y de la claridad en la comunicación con los docentes, lo que incrementa la probabilidad de inconsistencias y retrabajos. El análisis permitió concluir que la recolección de disponibilidad constituía un punto crítico dentro del flujo general de gestión de horarios, debido a su impacto directo en las fases posteriores de asignación.

5.1.6.2 Asignación de horarios – proceso antiguo

Figura 3.

Asignación de horarios - proceso antiguo



El modelamiento del proceso de asignación de horarios mediante BPMN permite representar de forma secuencial las actividades desarrolladas por el personal administrativo una vez consolidada la disponibilidad docente.

El análisis evidenció que la asignación de grupos y franjas horarias se realizaba exclusivamente mediante hojas de cálculo, utilizando como referencia archivos de periodos académicos anteriores. Este procedimiento implicaba una construcción progresiva del horario, donde cada nueva asignación debía ser verificada manualmente contra las ya existentes.

Se identificó que al momento de asignar una franja horaria a una asignatura específica, el personal debía validar manualmente que no existieran conflictos tales como:

- Un mismo docente asignado simultáneamente a dos grupos distintos.
- Un grupo académico con dos asignaturas en la misma franja.
- Inconsistencias en la distribución de intensidades horarias.
- Posibles sobrecargas en la jornada docente.

Estas validaciones se realizan mediante inspección visual de celdas y comparación manual de horarios, lo que convertía el proceso en una actividad altamente dependiente de la atención y experiencia del responsable.

Asimismo, el modelamiento evidenció que cualquier modificación —como el cambio de disponibilidad de un docente o la creación de un nuevo grupo— obligaba a rehacer parcialmente el archivo y revisar nuevamente múltiples asignaciones previas. Esta

característica genera un efecto acumulativo de complejidad, incrementando el tiempo requerido y el riesgo de inconsistencias.

Como resultado del modelamiento, se evidenció que el proceso presentaba las siguientes debilidades estructurales:

- Ausencia de validaciones automáticas de reglas de negocio.
- Alta probabilidad de solapamientos no detectados oportunamente.
- Dependencia crítica del conocimiento tácito del personal administrativo.
- Incremento exponencial de complejidad conforme aumentaban grupos y asignaturas.
- Baja trazabilidad de cambios y decisiones.

Esto permitió concluir que la ausencia de mecanismos sistemáticos de validación y control no solo afectaba la eficiencia operativa, sino que comprometía la confiabilidad institucional del horario final publicado.

5.1.7 Problemáticas identificadas

El análisis del proceso actual de gestión y asignación de horarios académicos permitió identificar un conjunto de limitaciones estructurales que afectan la eficiencia operativa y la confiabilidad de la información generada.

A partir del modelamiento realizado, se evidenció que las dificultades observadas no corresponden a fallas aisladas, sino a consecuencias inherentes a un esquema de trabajo manual y descentralizado, soportado en herramientas no especializadas.

Con el fin de estructurar el análisis, las problemáticas identificadas se organizaron en función de tres elementos: situación observada, causa que la origina e impacto institucional generado.

A continuación, se presenta la matriz consolidada de problemáticas:

Tabla 7.

Problemáticas Identificadas

<i>Problemática</i>	<i>Causa Principal</i>	<i>Impacto</i>
<i>Dependencia de herramientas no especializadas</i>	<i>Uso de hojas de cálculo y comunicación informal para captura y consolidación de datos</i>	<ul style="list-style-type: none"> - Incremento de errores por formato y duplicación - Baja trazabilidad de cambios - Procesos difíciles de auditar
<i>Ausencia de validaciones automáticas</i>	<i>Falta de mecanismos de software que verifiquen reglas de negocio en tiempo real</i>	<ul style="list-style-type: none"> - Detectar solapamientos manualmente - Requiere revisiones minuciosas - Posibilidad de conflictos en los horarios

<i>Alta carga operativa para el personal administrativo</i>	<i>Actividades repetitivas y manuales como revisar celdas y reconstruir horarios</i>	<ul style="list-style-type: none"> - <i>Consumo excesivo de tiempo del personal</i> - <i>Menor enfoque en actividades estratégicas</i> - <i>Saturación operativa durante periodos de ajuste</i>
<i>Falta de estandarización de datos</i>	<i>No existen formatos normalizados para disponibilidad docente y asignaciones</i>	<ul style="list-style-type: none"> - <i>Dificultad para integrar información de diferentes fuentes</i> - <i>Inconsistencias entre periodos académicos</i>
<i>Limitaciones en la escalabilidad del proceso</i>	<i>El uso de sistemas manuales no soporta el crecimiento de datos</i>	<ul style="list-style-type: none"> - <i>Dificultad para manejar mayor número de grupos y asignaturas</i> - <i>Incremento exponencial de complejidad sin soporte tecnológico</i>
<i>Carencia de soporte analítico para toma de decisiones</i>	<i>Ausencia de paneles, reportes o métricas automatizadas</i>	<ul style="list-style-type: none"> - <i>Información limitada para planificación</i> - <i>Dificultad para priorizar recursos o detectar patrones</i>
<i>Riesgos por errores humanos y pérdidas de información</i>	<i>Procesos no respaldados de forma sistemática</i>	<ul style="list-style-type: none"> - <i>Pérdida de datos</i> - <i>Necesidad de rehacer trabajo por errores</i> - <i>Baja confianza en la información resultante</i>

La matriz permite observar que la mayoría de las problemáticas se originan en la ausencia de mecanismos formales de validación, integración y control de la información.

Asimismo, se evidencia una alta dependencia de actividades manuales, lo que incrementa el riesgo de errores administrativos.

En consecuencia, se identifica la necesidad de diseñar un sistema de información que permita centralizar datos académicos, automatizar validaciones y garantizar coherencia en la asignación de horarios.

5.1.8 Delimitación y estructuración funcional del sistema

Como resultado del análisis del proceso manual y de las problemáticas identificadas, se procedió a delimitar formalmente el alcance y la estructura funcional del Sistema de Información para la Gestión de Horarios Académicos (SIGHA).

Esta etapa tuvo como objetivo definir los límites del sistema, identificar sus interacciones con los actores institucionales y establecer las funcionalidades que permitirán automatizar y controlar el proceso de gestión académica.

A diferencia del modelamiento de procesos presentado anteriormente, el cual describe la secuencia de actividades operativas, esta sección se enfoca en la definición estructural del sistema como solución tecnológica. Para ello, se emplearon diagramas de contexto y diagramas de casos de uso, herramientas que permiten representar de manera clara tanto el entorno del sistema como las funcionalidades que ofrecerá.

La delimitación funcional constituye un paso fundamental dentro de la fase de inicio, ya que establece las bases para la identificación formal de requerimientos y orienta el diseño posterior de la arquitectura del sistema.

5.1.8.1 Gestión de áreas y asignaturas.

Figura 4.

Diagrama de contexto – Gestión de áreas y asignaturas



El diagrama de contexto correspondiente a la gestión de áreas y asignaturas representa la interacción entre el Coordinador Académico o Director de Escuela y el sistema SIGHA en relación con la configuración de la estructura académica básica.

En este modelo, el sistema se presenta como una entidad central que recibe y administra la información asociada a las áreas de conocimiento y a las asignaturas que pertenecen a cada una de ellas. El Coordinador Académico o Director de Escuela actúa como el actor responsable de:

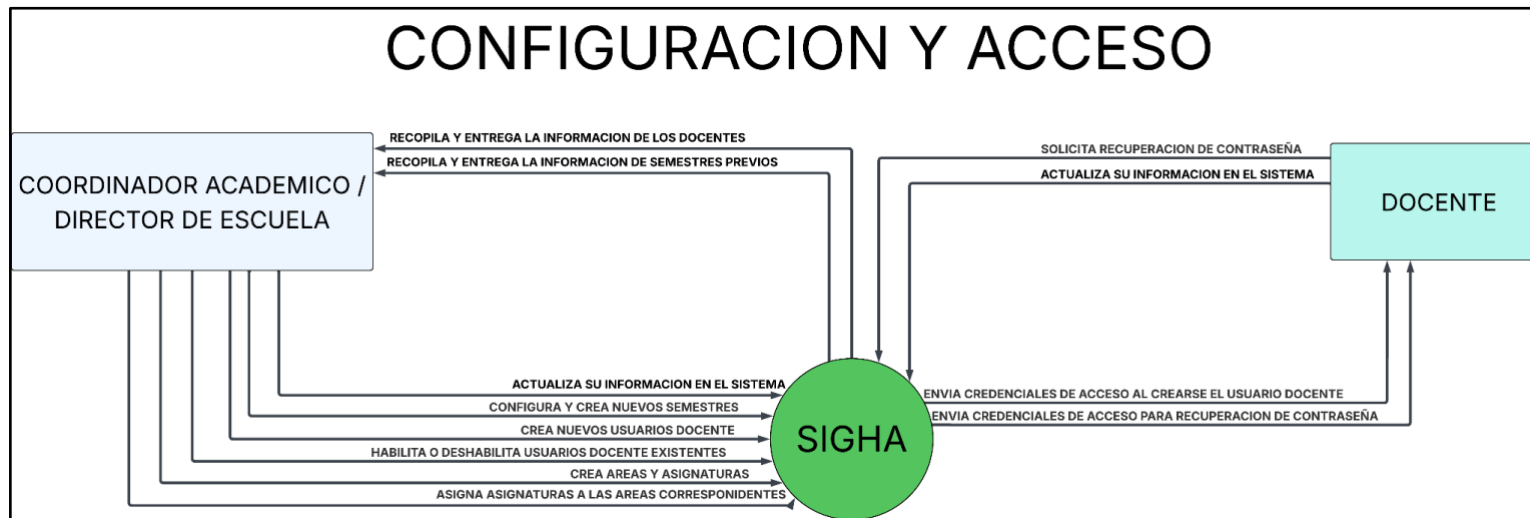
- Crear áreas de conocimiento.
- Editar áreas existentes.
- Crear asignaturas.
- Asignar cada asignatura a un área específica.
- Modificar la información de las asignaturas y su relación con el área correspondiente

Este diagrama permite delimitar claramente que la administración de la estructura académica (áreas y asignaturas) constituye una responsabilidad funcional del sistema, bajo control del Coordinador Académico o Director de Escuela, estableciendo así la base organizativa sobre la cual se soportará los procesos posteriores de disponibilidad y asignación de horarios.

5.1.8.2 Configuración y acceso al sistema

Figura 5.

Diagrama de contexto – Configuración y acceso



Este diagrama describe la interacción relacionada con la administración general del sistema y la gestión de acceso de los usuarios.

El Coordinador Académico o Director de Escuela recopila y suministra al sistema información correspondiente a docentes y datos históricos de semestres anteriores. A partir de esta información, el sistema:

- Configura y crea nuevos semestres académicos.
- Crea usuarios docentes.
- Habilita o deshabilita usuarios existentes.
- Genera y envía credenciales de acceso.
- Gestiona procesos de recuperación de contraseña.

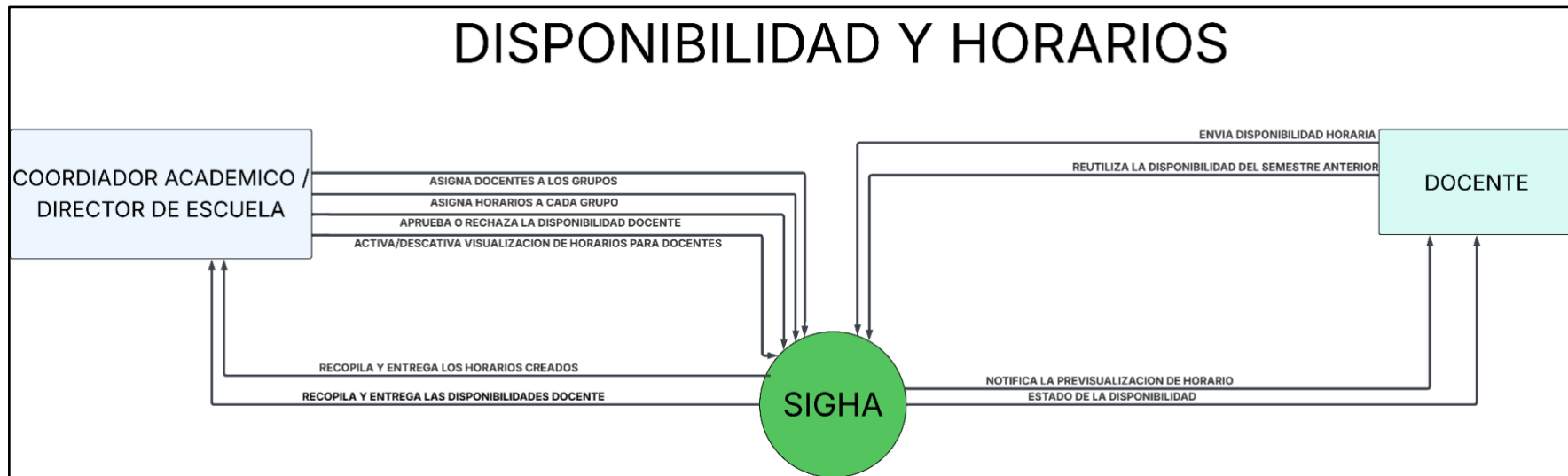
Por su parte, el Docente interactúa con el sistema solicitando recuperación de contraseña y actualizando su información personal.

Este diagrama permite establecer que la gestión de acceso y configuración institucional es responsabilidad del sistema bajo control administrativo, garantizando seguridad y control sobre los usuarios.

5.1.8.3 Disponibilidad y horarios

Figura 6.

Diagrama de contexto – Disponibilidad y horarios



Este diagrama representa la interacción entre el Coordinador Académico o Director de Escuela, el Docente y el sistema SIGHA en el proceso general de gestión de disponibilidad y horarios.

El Docente:

- Envía su disponibilidad horaria.
- Puede reutilizar la disponibilidad del semestre anterior.
- Recibe notificaciones sobre el estado de su disponibilidad y la previsualización de horarios.

El Coordinador Académico o Director de Escuela:

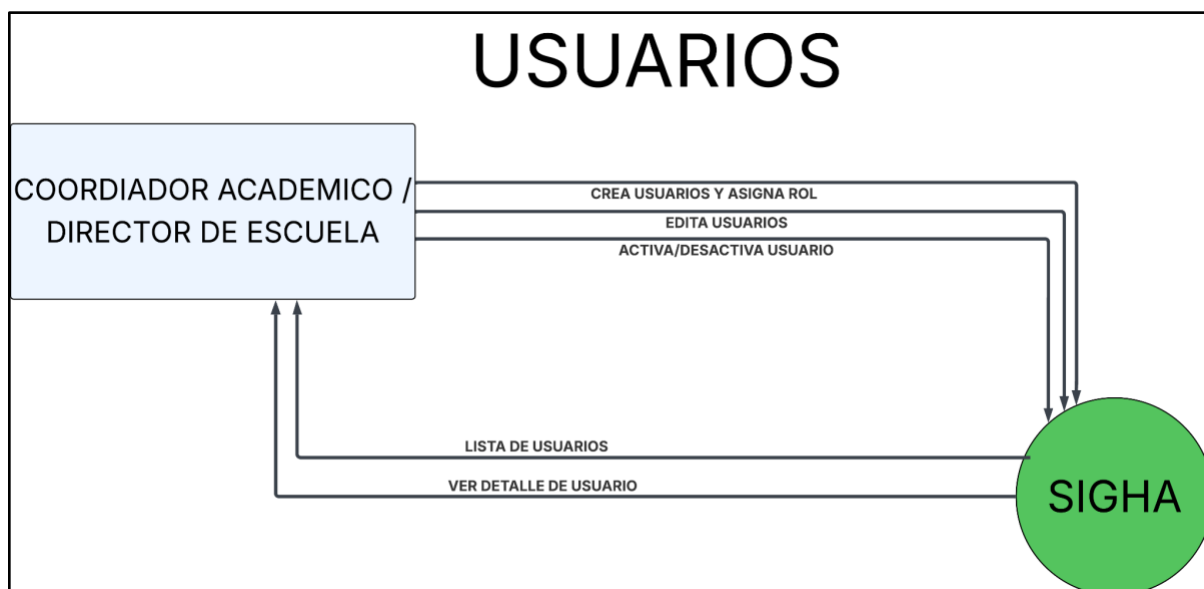
- Asigna docentes a grupos.
- Asigna horarios a cada grupo.
- Aprueba o rechaza las disponibilidades.
- Activa o desactiva la visualización de horarios para docentes.
- Consulta y recopila los horarios creados y las disponibilidades registradas.

Este contexto delimita que la gestión académica operativa se desarrolla dentro del sistema, el cual actúa como eje de validación y control.

5.1.8.4 Gestión de usuarios

Figura 7.

Diagrama de contexto – Gestión de usuarios



Este diagrama representa la interacción específica relacionada con la administración de usuarios dentro del sistema.

El Coordinador Académico o Director de Escuela puede:

- Crear usuarios y asignarles un rol.
- Editar usuarios existentes.
- Activar o desactivar cuentas.

- Consultar la lista de usuarios.
- Visualizar el detalle de cada usuario.

5.1.8.5 Diagramas de Casos de Uso. Los diagramas de casos de uso del sistema propuesto permiten representar las funcionalidades principales del Sistema de Información para la Gestión de Horarios Académicos (SIGHA) y la interacción de los actores con cada proceso del sistema.

En todos los diagramas intervienen, según el proceso, los siguientes actores:

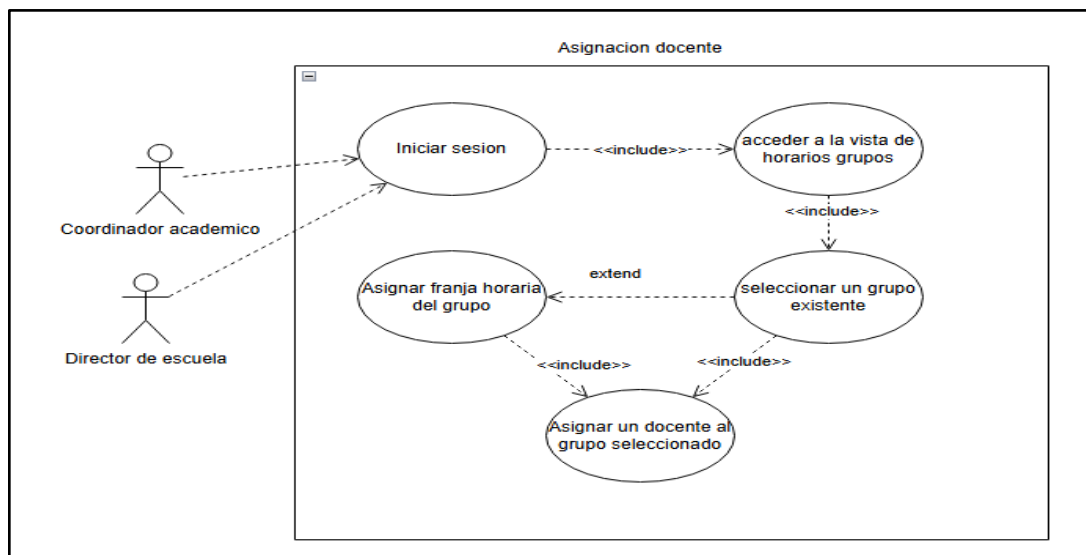
- Coordinador Académico
- Director de Escuela
- Docente (Profesor)

Cada diagrama modela un proceso específico del sistema nuevo.

5.1.8.5.1 Asignación de profesores

Figura 8.

Diagrama de caso de uso – Asignación de profesores



Este caso de uso describe cómo se asignan docentes a los grupos y franjas horarias en el sistema.

- **Actores:** Coordinador Académico, Director de Escuela.
- **Flujos principales:**
 - Iniciar sesión (incluye acceder a la vista de horarios de grupos).
 - Acceder a la vista de horarios de grupos existentes.
 - Seleccionar un grupo al que se asignará la franja horaria y el docente.
 - Asignar la franja horaria al grupo (extiende “seleccionar un grupo existente”).

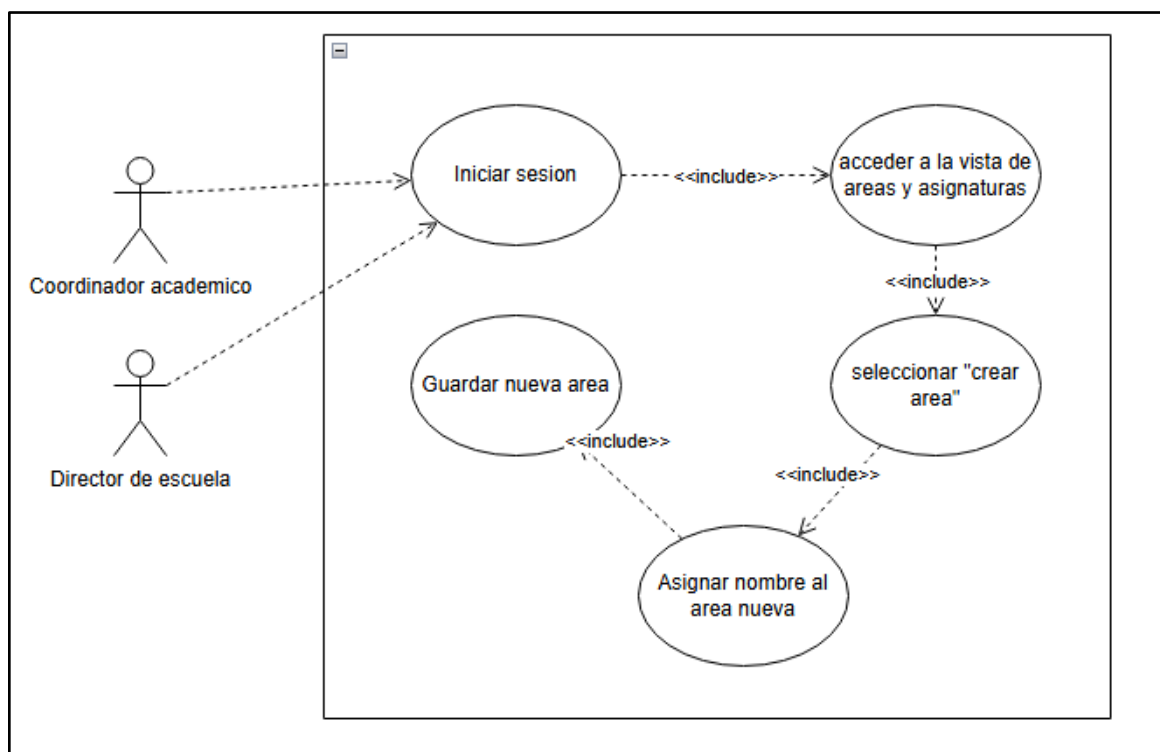
- Asignar un docente al grupo (incluido por “asignar franja horaria” y “seleccionar grupo existente”).

Propósito: permite que los responsables administrativos organicen los horarios y asignen docentes a cada grupo de manera centralizada y controlada.

5.1.8.5.2 Creación de Áreas

Figura 9.

Diagrama de caso de uso – Creación de áreas



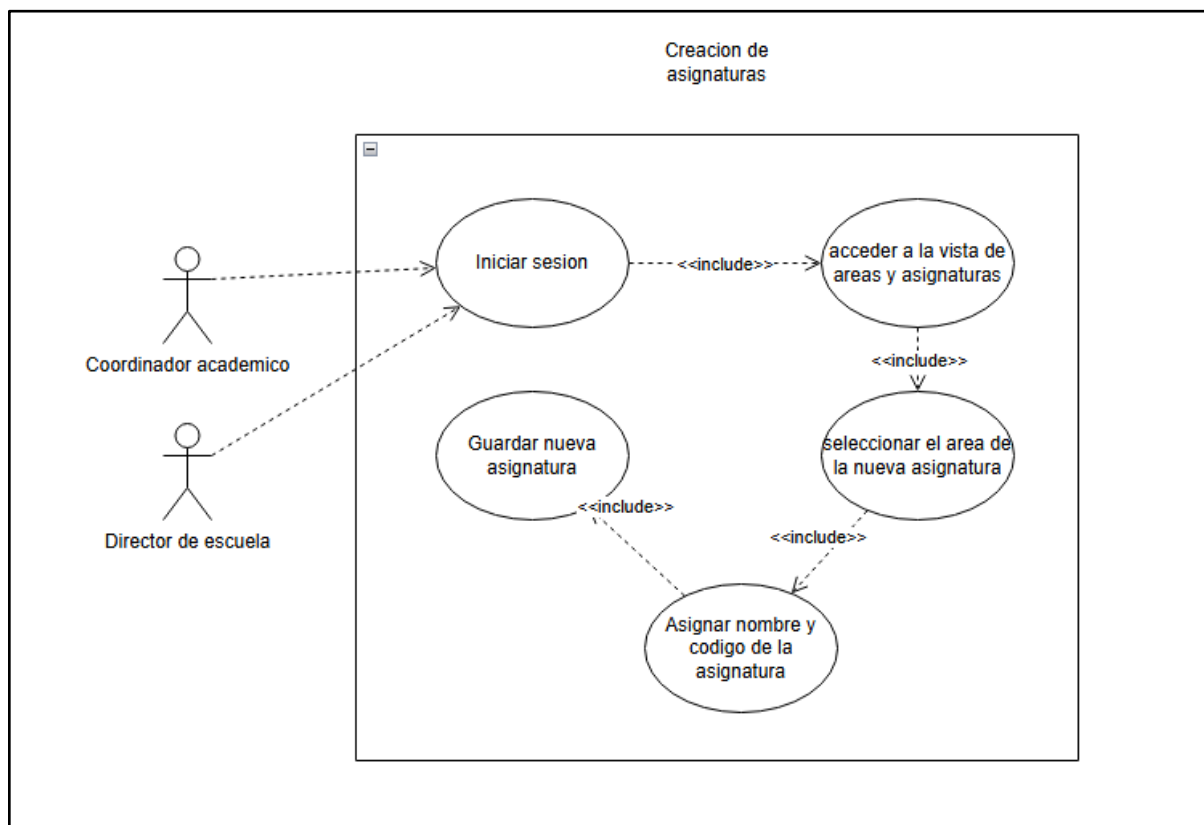
- **Actores:** Coordinador Académico, Director de Escuela.
- **Flujos principales:**
 - Iniciar sesión (incluye acceder a la vista de áreas y asignaturas).
 - Acceder a la vista de áreas y asignaturas (incluye seleccionar “crear área”).
 - Seleccionar “crear área” (incluye asignar nombre al área nueva).
 - Asignar nombre al área.
 - Guardar nueva área (incluye asignar nombre al área).

Propósito: Permite la creación y administración de la estructura académica de la escuela de manera organizada y supervisada.

5.1.8.5.3 Creación de Asignaturas

Figura 10.

Diagrama de caso de uso – Creación de Asignaturas



- **Actores:** Coordinador Académico, Director de Escuela.
- **Flujos principales:**
 - Iniciar sesión (incluye acceder a la vista de áreas y asignaturas).
 - Acceder a la vista de áreas y asignaturas.
 - Seleccionar el área a la que pertenece la nueva asignatura.

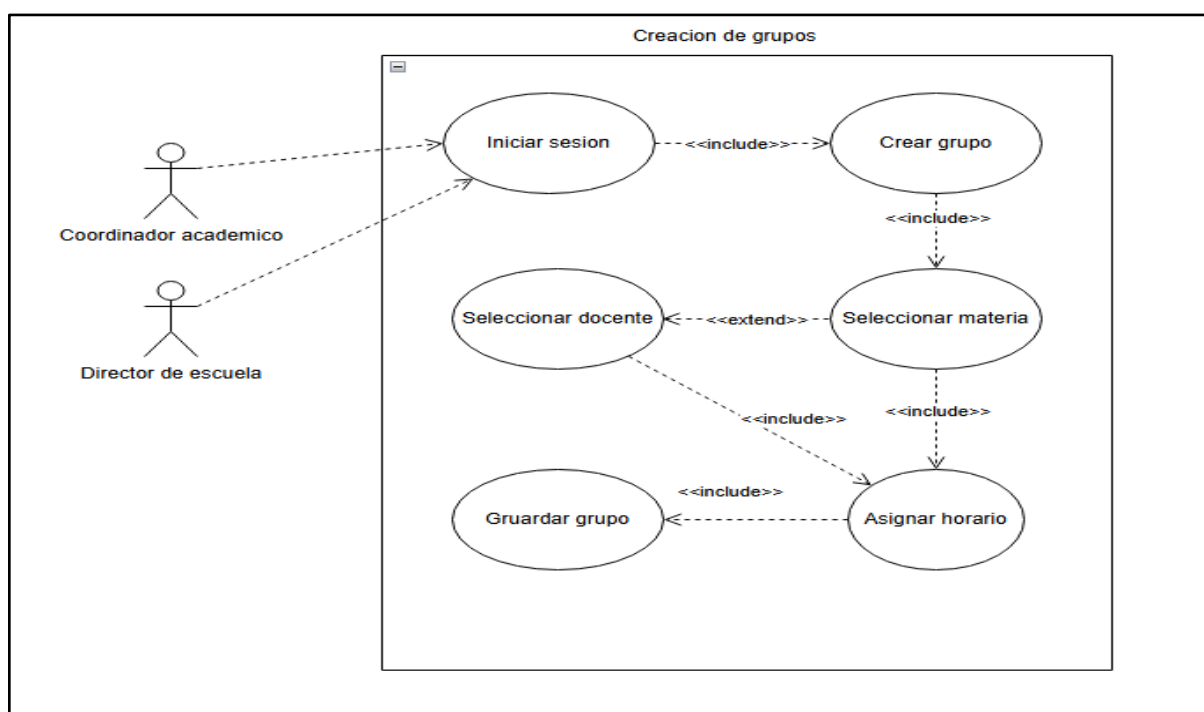
- Asignar nombre y código de la asignatura.
- Guardar nueva asignatura (incluye asignar nombre y código).

Propósito: asegurar que todas las asignaturas se registren correctamente con sus identificadores y relaciones a áreas académicas, facilitando la asignación de horarios.

5.1.8.5.4 Creación de Grupos

Figura 11.

Diagrama de caso de uso – Creación de Grupos



- **Actores:** Coordinador Académico, Director de Escuela.
- **Flujos principales:**

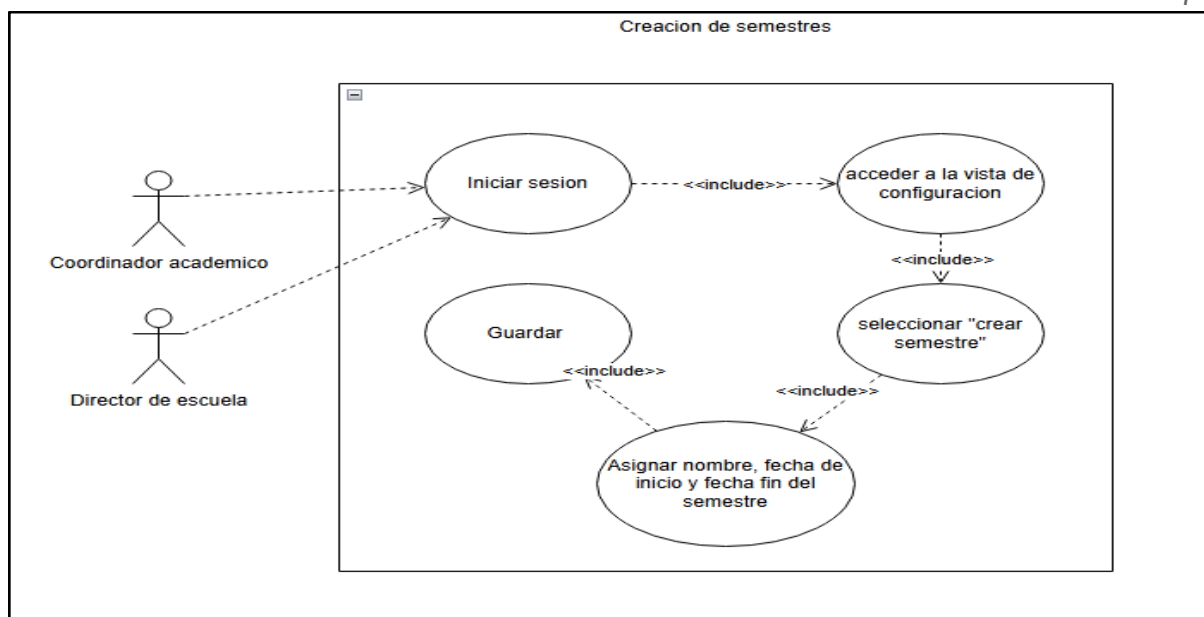
- Iniciar sesión.
- Crear un nuevo grupo.
- Seleccionar la materia a asignar al grupo.
- Seleccionar el docente que impartirá la materia (extiende “seleccionar materia”).
- Asignar horario al grupo.
- Guardar grupo.

Propósito: Gestionar la formación de grupos académicos vinculando asignaturas, docentes y horarios, manteniendo consistencia en la información del sistema.

5.1.8.5.5 Creación de Semestres

Figura 12.

Diagrama de caso de uso – Creación de Semestres



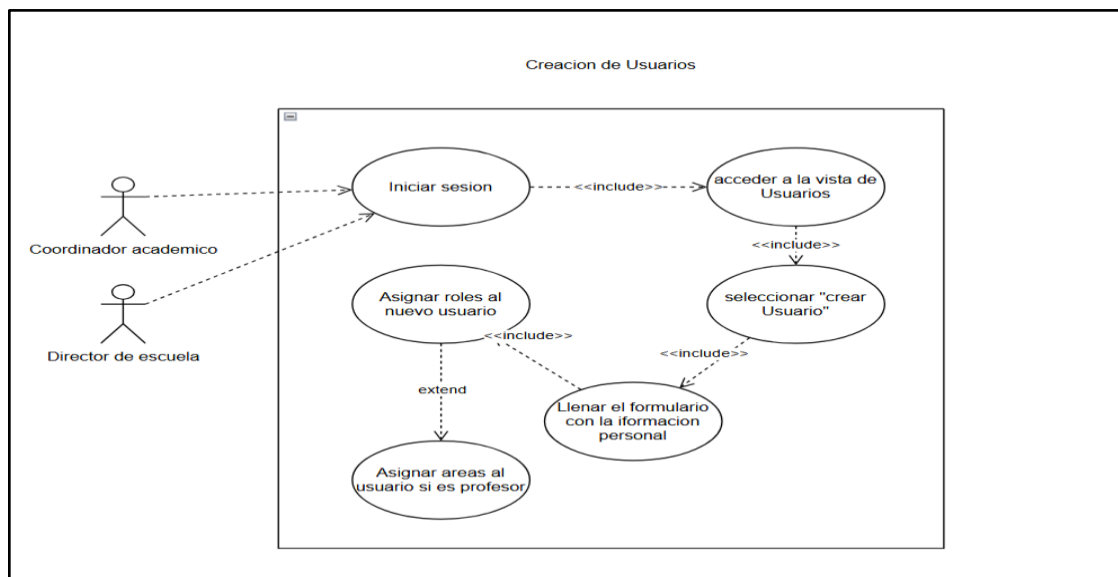
- **Actores:** Coordinador Académico, Director de Escuela.
- **Flujos principales:**
 - Iniciar sesión (incluye acceder a la vista de configuración).
 - Acceder a la vista de configuración.
 - Seleccionar “crear semestre”.
 - Asignar nombre, fecha de inicio y fecha fin del semestre (incluido en “crear semestre”).
 - Guardar el semestre.

Propósito: permitir la planificación semestral y la gestión de datos históricos, facilitando la reutilización de disponibilidad y horarios previos.

5.1.8.5.6 Creación de Usuarios

Figura 13.

Diagrama de caso de uso – Creación de Usuarios



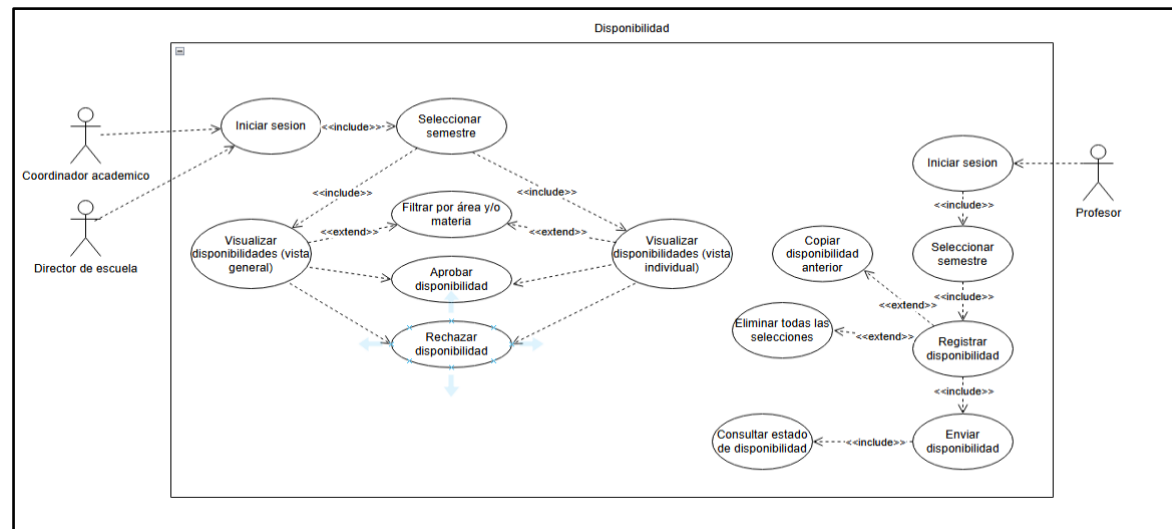
- **Actores:** Coordinador Académico, Director de Escuela.
- **Flujos principales:**
 - Iniciar sesión (incluye acceder a la vista de usuarios).
 - Acceder a la vista de usuarios.
 - Seleccionar “crear usuario”.
 - Llenar el formulario con información personal.
 - Asignar roles al usuario (incluye llenar formulario).
 - Asignar áreas si el usuario es profesor (extiende “asignar roles”).

Propósito: garantizar control sobre los usuarios del sistema, sus roles y áreas de gestión, asegurando seguridad y trazabilidad en la administración de cuentas.

5.1.8.5.7 Disponibilidad

Figura 14.

Diagrama de caso de uso – Disponibilidad



- **Actores:** Coordinador Académico, Director de Escuela.
- **Flujos principales:**
 - Iniciar sesión (incluido en otros casos de uso).
 - Seleccionar semestre.
 - Registrar disponibilidad (Profesor).
 - Copiar disponibilidad del semestre anterior (Profesor, opcional).
 - Eliminar todas las selecciones (extiende registrar disponibilidad).
 - Enviar disponibilidad (incluye consultar estado).
 - Aprobar o rechazar disponibilidad (Coordinador o Director).
 - Consultar estado de disponibilidad (Profesor, incluido en enviar disponibilidad).

Propósito: centralizar, validar y controlar la disponibilidad docente, eliminando duplicidad de información y mejorando la planificación de horarios.

5.1.9 Modelamiento del sistema propuesto

Como resultado del análisis del proceso manual y de las problemáticas identificadas, se elaboró el modelamiento del proceso rediseñado incorporando el Sistema de Información para la Gestión de Horarios Académicos (SIGHA) como eje central de validación y control.

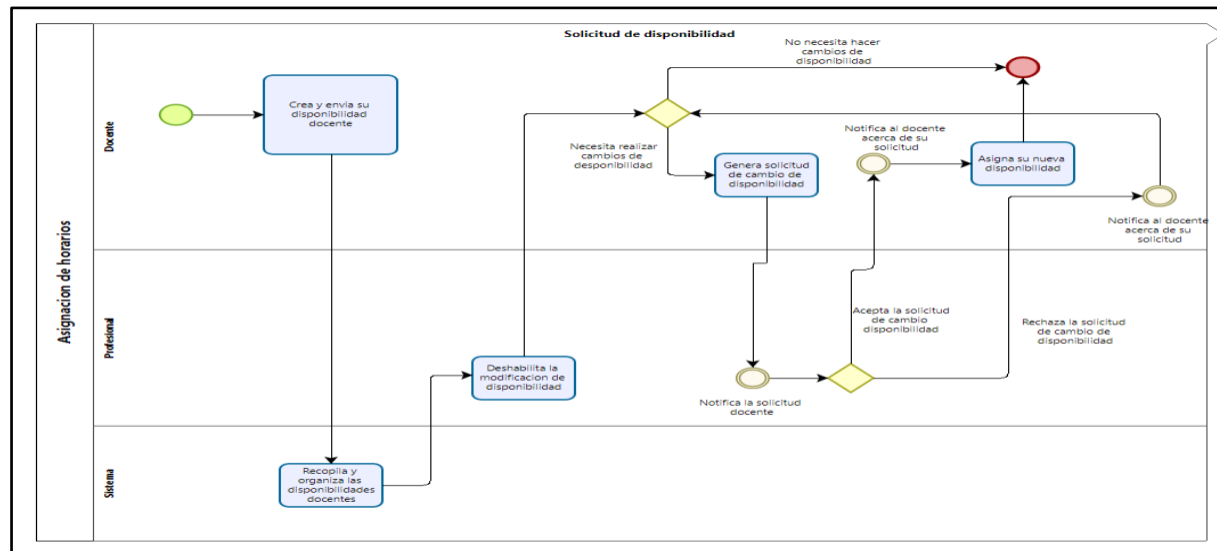
El objetivo de este modelamiento fue visualizar cómo la automatización permitiría estandarizar la captura de información, incorporar validaciones en tiempo real y reducir la dependencia de actividades manuales.

A continuación, se presentan los diagramas BPMN correspondientes al proceso rediseñado.

5.1.9.1 Solicitud y gestión de disponibilidad – proceso rediseñado

Figura 15.

Solicitud de disponibilidad – proceso propuesto



El diagrama BPMN del proceso rediseñado de solicitud de disponibilidad representa la interacción entre el Docente, el Profesional y el Sistema durante el registro, validación y confirmación de la disponibilidad horaria académica.

El proceso inicia cuando el docente ingresa al sistema y registra su disponibilidad en un formato estructurado. A diferencia del procedimiento manual anteriormente descrito, la información se introduce directamente en la plataforma, eliminando la necesidad de intermediación por correo electrónico o mensajería externa.

Una vez enviada la solicitud, el sistema evalúa si la disponibilidad requiere ajustes o validaciones adicionales conforme a las reglas académicas establecidas. En caso de detectarse inconsistencias o necesidad de modificación, el sistema genera automáticamente una solicitud de cambio y notifica al docente para que realice los ajustes correspondientes.

Posteriormente, el profesional revisa la disponibilidad registrada y decide su aceptación o rechazo. Este punto de control permite garantizar que la información cumpla con los lineamientos institucionales antes de ser utilizada en la fase de asignación de horarios.

Si la disponibilidad es aceptada:

- El sistema deshabilita su edición.
- La información queda consolidada en la base de datos.
- Se notifica al docente la aprobación.

Si la disponibilidad es rechazada:

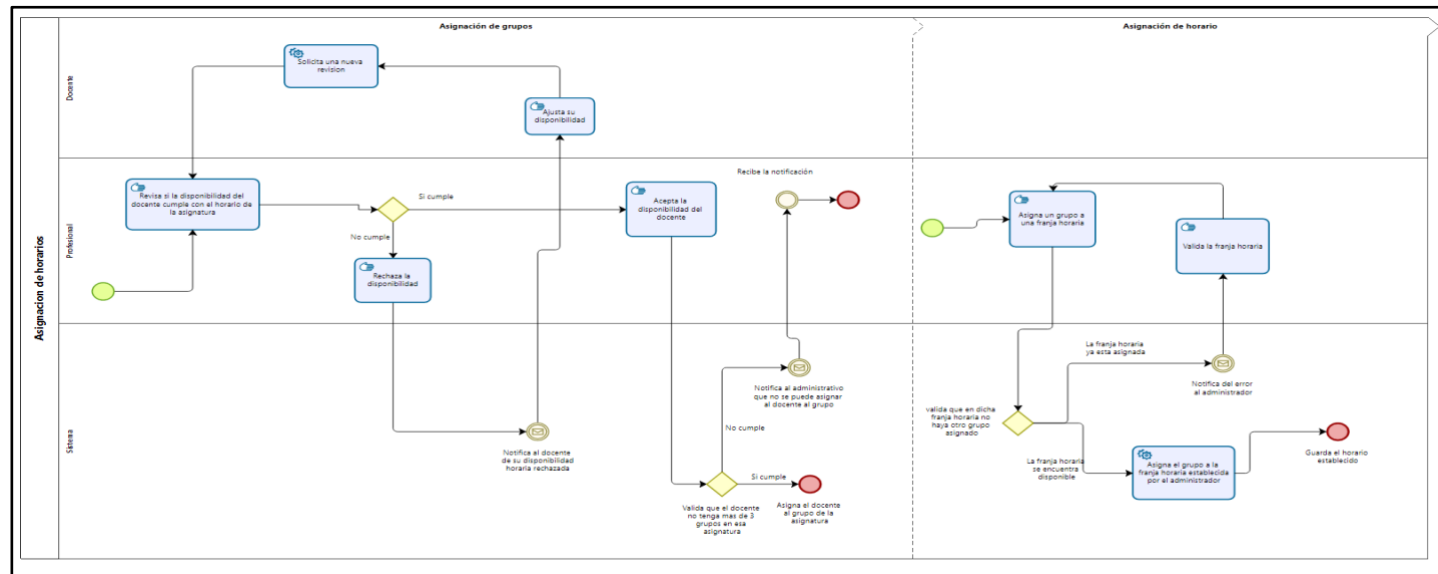
- El proceso retorna a la etapa de ajuste.

Este modelamiento evidencia que el sistema introduce mecanismos de estandarización, control y trazabilidad que no existían en el proceso manual. La automatización del registro y validación reduce errores de transcripción, elimina duplicidad de información y centraliza los datos necesarios para la posterior asignación de horarios.

5.1.9.2 Asignación de horarios – proceso rediseñado

Figura 16.

Asignación de grupos y horarios – proceso propuesto



El diagrama BPMN del proceso rediseñado de asignación de horarios representa la interacción entre el Profesional, el Docente y el Sistema durante la asignación de grupos y franjas horarias académicas.

El proceso inicia cuando el profesional revisa la disponibilidad registrada previamente en el sistema y verifica su correspondencia con el horario definido para la asignatura. A diferencia del proceso manual, esta verificación se realiza sobre información estructurada y almacenada de forma centralizada.

El sistema incorpora validaciones automáticas que permiten determinar:

- Si la disponibilidad del docente coincide con el horario requerido.
- Si la franja horaria ya se encuentra asignada a otro grupo.

En caso de que la disponibilidad no cumpla con los requisitos establecidos, el sistema genera una notificación que informa la inconsistencia y el proceso puede retornar a la etapa de ajuste de disponibilidad, evitando que se realicen asignaciones inválidas.

Cuando las validaciones son satisfactorias, el coordinador académico procede a asignar el grupo a una franja horaria específica. El sistema verifica en tiempo real la disponibilidad de dicha franja y confirma que no existan conflictos con otras asignaciones registradas.

Si todas las reglas de negocio se cumplen, el sistema registra la asignación y guarda el horario establecido, asegurando consistencia, trazabilidad y almacenamiento centralizado de la información.

En caso contrario, se genera una notificación de error que impide la asignación, evitando la publicación de conflictos no detectados y reduciendo significativamente el riesgo de inconsistencias operativas.

El modelamiento del proceso rediseñado evidencia una mejora sustancial respecto al procedimiento manual anteriormente descrito, al incorporar mecanismos automáticos de validación, control de restricciones y registro sistemático de decisiones, fortaleciendo así la confiabilidad institucional del horario académico generado.

5.1.10 Análisis comparativo del proceso

El análisis comparativo entre el proceso anterior y el proceso rediseñado evidencia una transformación significativa en la gestión de la disponibilidad docente y la asignación de horarios académicos.

En el modelo previo, ambas actividades dependían de mecanismos manuales, comunicación descentralizada y consolidación en herramientas externas, lo que generaba riesgos de error, inconsistencias en la información y ausencia de trazabilidad. La verificación de restricciones académicas se realizaba de manera manual, incrementando la carga operativa y la probabilidad de conflictos no detectados.

Con la incorporación del sistema, los procesos se estructuran bajo reglas de negocio claramente definidas, validaciones automáticas y almacenamiento centralizado de la información. La disponibilidad docente se registra en formatos estandarizados y la asignación de horarios incorpora controles en tiempo real que impiden conflictos de franja, sobrecarga docente o inconsistencias en la programación académica.

El rediseño no solo optimiza el flujo operativo, sino que introduce mecanismos de control, trazabilidad y validación que fortalecen la confiabilidad del proceso institucional. De esta manera, el sistema propuesto actúa como un elemento de soporte a la toma de decisiones, reduciendo riesgos administrativos y mejorando la eficiencia en la planificación académica.

5.1.11 Elicitación inicial de requerimientos

Durante la fase de inicio se llevó a cabo el proceso de elicitación de requerimientos del sistema SIGHA. Este proceso permitió traducir las necesidades institucionales detectadas en el análisis del proceso

Durante la fase de inicio se llevó a cabo el proceso de elicitación de requerimientos del sistema SIGHA. Este proceso permitió traducir las necesidades institucionales detectadas en el análisis del proceso actual en un conjunto preliminar de requerimientos clasificados en funcionales, no funcionales y complementarios.

La elicitación se realizó mediante las siguientes técnicas:

- **Entrevistas semiestructuradas** con el Director de Escuela y el Coordinador Académico, orientadas a comprender las necesidades operativas, los puntos críticos del proceso actual y las expectativas sobre el sistema.
- **Observación directa** del proceso de recolección de disponibilidades y asignación de horarios, identificando las actividades manuales propensas a errores documentadas en los diagramas BPMN AS-IS (Figuras 1-2).
- **Revisión documental** de formatos, hojas de cálculo y registros históricos utilizados para la gestión de horarios.

A partir de estas técnicas, se identificaron **tres categorías de requerimientos**:

Requerimientos funcionales (RF): Se identificaron 10 requerimientos funcionales que cubren los dominios operativos del sistema. Para cada requerimiento se elaboró un formato detallado que incluye reglas de negocio, criterios de aceptación y precondiciones (véase Apéndice A).

Los dominios funcionales identificados son:

Tabla 8.

Requerimientos funcionales del sistema

<i>No.</i>	<i>Requerimientos Funcionales</i>
------------	-----------------------------------

-
- RF01 SIGHA-Aplicación-base-y-Menú-Principal*
- RF02 SIGHA-Gestion-profesores-y-roles-permisos*
- RF03 SIGHA-Gestión-disponibilidades-profesores*
- RF04 SIGHA-Gestión-Áreas-y-Asignaturas*
- RF05 SIGA-Gestión-disponibilidades-administrativos*
- RF06 SIGHA-Gestion-Grupos*
- RF07 SIGHA-Gestion-Calendario-de-grupos*
- RF08 SIGHA-Consulta-Calendario-escuela*
- RF09 SIGHA-Configuración-y-parametrización-sistema*
- RF10 SIGHA-Configuración-usuario*
-

5.1.11.1 Requerimientos No Funcionales. Los requerimientos no funcionales establecen los criterios de calidad y las restricciones bajo las cuales debe operar el sistema. Estos definen aspectos como rendimiento, seguridad, disponibilidad, usabilidad y mantenibilidad, asegurando que SIGHA no solo cumpla con las funcionalidades previstas, sino que también garantice eficiencia, confiabilidad y una adecuada experiencia para el personal de la Escuela de Ingeniería de Sistemas de la Universidad industrial de Santander.

Tabla 9.

Requerimientos no funcionales del sistema

<i>No.</i>	<i>Requerimientos no funcionales</i>
<i>01</i>	<i>El sistema se desarrollará utilizando Vue.js para el frontend y Java con Spring Boot para el backend. La base de datos será Postgres. Se utilizará Git para el control de versiones del código, con ramas separadas para desarrollo y producción</i>
<i>02</i>	<i>El sistema debe responder en máximo 3 segundos para operaciones críticas como la consulta de horarios y consulta de usuarios.</i>
<i>03</i>	<i>El sistema debe ser multiusuario, soportando al menos 100 peticiones simultáneas sin degradación de rendimiento.</i>
<i>04</i>	<i>El sistema debe utilizar encriptación bcrypt para almacenamiento seguro de contraseñas, HTTPS para transmisión de datos y JWT para el manejo de sesiones con tokens de acceso que expiran cada hora.</i>
<i>05</i>	<i>El sistema debe mantener una disponibilidad del 99% durante los períodos críticos de creación, actualización y organización de horarios, con un tiempo máximo de recuperación ante fallos de 30 minutos.</i>
<i>06</i>	<i>El tiempo de carga inicial no superará los 1.5 segundos. La interfaz seguirá los colores institucionales y propios para mantener la consistencia visual.</i>

07 *El manual de usuario será mediante videotutoriales del uso del sistema.*



5.1.11.2 Requerimientos Complementarios. Disponer de equipos de cómputo con acceso a internet para la implementación y uso del sistema:

- Cumplir con las normas de comunicación TCP/IP
- El personal administrativo debe recibir capacitación en el uso del sistema
- Se debe mantener una copia de seguridad actualizada del código fuente en repositorios Git
- Se debe contar con un plan de contingencia en caso de fallos durante períodos críticos

- Mantener documentación técnica actualizada del sistema para futuros desarrollos

5.2 Fase de Elaboración

Durante esta fase se busca reducir los riesgos técnicos y funcionales del proyecto mediante la definición clara de las actividades institucionales identificadas y la solución planteada.

5.2.1 Consolidación de requerimientos

A partir del análisis del proceso actual y de los resultados obtenidos en la fase de inicio, se realizó la consolidación de los requerimientos del sistema SIGHA. Esta actividad permitió traducir las necesidades identificadas en especificaciones claras y estructuradas que orientan el diseño y la construcción del sistema.

Los requerimientos se clasifican en funcionales, no funcionales y complementarios, con el fin de facilitar su análisis y trazabilidad.

5.2.1.1 Requerimientos funcionales

Los requerimientos funcionales describen las funcionalidades que el sistema SIGHA debe ofrecer para apoyar el proceso de gestión de horarios académicos. Estos requerimientos fueron identificados mediante entrevistas al personal administrativo, lo que permitió detectar las principales falencias y necesidades del flujo actual. Se elaboró un formato de requerimiento

donde se abordan reglas de negocio, criterios de aceptación y precondiciones, estableciendo una visión inicial del alcance funcional del sistema.

Los requerimientos funcionales consolidados se presentan en el capítulo de Resultados (véase Tabla 8) y sus formatos detallados se incluyen en el Apéndice A.

5.2.1.2 Requerimientos no funcionales

Los requerimientos no funcionales establecen las características de calidad que debe cumplir el sistema SIGHA, tales como desempeño, seguridad, disponibilidad y usabilidad. Estos requerimientos garantizan que el sistema no solo funcione correctamente, sino que también cumpla con las expectativas institucionales y técnicas.

Los requerimientos no funcionales consolidados se presentan en el capítulo de Resultados (véase Tabla 9).

5.2.1.3 Requerimientos complementarios

Los requerimientos complementarios agrupan aquellas condiciones y restricciones que influyen en el desarrollo del sistema, pero que no corresponden directamente a funcionalidades o atributos de calidad. Entre estos se encuentran consideraciones institucionales, tecnológicas y normativas.

5.2.2 *Diseño del sistema*

El diseño del sistema SIGHA se realizó con el objetivo de definir una estructura técnica coherente y flexible que permita soportar los requerimientos identificados. En esta etapa se establecen los principales modelos y decisiones de diseño que servirán como guía para la fase de construcción.

5.2.2.1 Modelo de datos

El modelo de datos describe las entidades principales involucradas en el dominio del sistema y las relaciones existentes entre ellas. Este modelo permite representar conceptualmente la información que será gestionada por SIGHA y establecer las bases para su almacenamiento y manipulación.

El sistema utiliza una base de datos relacional PostgreSQL, cuyo diseño contempla las entidades principales: usuarios, áreas, materias, semestres, disponibilidades, grupos y horarios, junto con sus relaciones y restricciones de integridad referencial. El modelo detallado y el diccionario de datos se presentan en el capítulo de Resultados (véase sección 5.2.1).

5.2.2.2 Diseño arquitectónico del sistema

El diseño arquitectónico define la estructura general del sistema SIGHA, identificando los principales componentes, módulos y sus interacciones. Esta arquitectura fue definida considerando criterios de escalabilidad, mantenibilidad y alineación con las necesidades institucionales.

Se adoptó una arquitectura de tres capas (cliente-servidor) que separa la capa de presentación (Vue.js), la capa de lógica de negocio (Spring Boot con Java) y la capa de datos (PostgreSQL). La descripción detallada de la arquitectura se presenta en el capítulo de Resultados (véase sección 5.2.2).

5.2.2.3 Prototipos visuales

Con el fin de validar de manera temprana la usabilidad del sistema, se desarrollaron prototipos visuales de las interfaces principales utilizando la herramienta Figma. Estos prototipos permiten representar el flujo general de navegación y las pantallas clave del sistema, sin llegar a una implementación funcional.

Los prototipos fueron presentados al usuario final para su retroalimentación. Las observaciones recibidas se incorporaron en las iteraciones posteriores de diseño. Los prototipos se presentan en el capítulo de Resultados (véase sección 5.2.3).

5.2.3 Análisis de riesgos

Durante la fase de elaboración se identificaron riesgos técnicos y organizacionales que podrían afectar el desarrollo del sistema SIGHA. Este análisis permitió anticipar posibles dificultades y proponer estrategias de mitigación que reduzcan su impacto en el proyecto.

Se identificaron 8 riesgos principales, clasificados por probabilidad e impacto, con sus correspondientes estrategias de mitigación. La matriz de riesgos completa se presenta en el capítulo de Resultados (véase Tabla 21).

5.2.4 Entregables de la fase de elaboración

Como resultado de la fase de elaboración se generaron diversos entregables que documentan la definición funcional y el diseño del sistema SIGHA, y que sirven como base directa para la fase de construcción.

Tabla 10.

Entregables: fase de elaboración

<i>Entregable</i>	<i>Descripción</i>
<i>Modelo de Datos</i>	<i>Diseño lógico de la base de datos relacional PostgreSQL, incluyendo entidades principales, relaciones y diccionario de datos.</i>
<i>Documento de Diseño Arquitectónico</i>	<i>Descripción de la arquitectura de tres capas (Vue.js, Spring Boot, PostgreSQL), módulos del sistema y sus interacciones</i>
<i>Prototipos Visuales</i>	<i>Diseños preliminares de las interfaces gráficas elaborados en Figma, validados con usuarios finales.</i>

Plan de Gestión de riesgos

Identificación de riesgos técnicos y organizacionales con su probabilidad, impacto y estrategia de mitigación.

La fase de elaboración del proyecto tuvo como propósito transformar la comprensión funcional obtenida en la fase de inicio en una definición técnica estructurada del sistema SIGHA. En esta etapa se consolidaron los requerimientos identificados previamente y se establecieron las bases arquitectónicas, estructurales y técnicas que orientaron el desarrollo posterior del sistema.

Mientras que la fase de inicio permitió delimitar el alcance funcional, modelar el proceso actual y proponer el proceso de rediseño, la fase de elaboración se enfocó en reducir la incertidumbre técnica del proyecto. Para ello, se definieron con mayor precisión las especificaciones funcionales y no funcionales, se diseñó la arquitectura del sistema bajo un enfoque cliente–servidor con separación en capas, se estructuró el modelo de datos que soporta la gestión académica y se identificaron los principales riesgos técnicos asociados a la implementación.

Los artefactos generados en esta fase cumplen un doble propósito: por una parte, garantizan la coherencia entre los requerimientos y la solución técnica propuesta; por otra, sirven como guía directa para la fase de construcción, asegurando trazabilidad entre las necesidades institucionales y los componentes desarrollados.

Como resultado de la fase de elaboración se obtuvieron los siguientes productos:

- Consolidación formal de requerimientos funcionales, no funcionales y su representación mediante historias de usuario.
- Definición del diseño arquitectónico del sistema.
- Modelamiento conceptual y lógico de la base de datos.
- Identificación y análisis de riesgos técnicos del proyecto.

Estos resultados constituyen la base estructural sobre la cual se desarrolló la implementación del sistema en la fase siguiente.

5.2.5 Consolidación y Especificación de Requerimientos

A partir de los resultados obtenidos en la fase de inicio, particularmente del proceso de elicitación descrito en la Sección 5.1.7, se realizó la consolidación formal de los requerimientos del sistema SIGHA. Esta actividad tuvo como objetivo transformar las necesidades identificadas en especificaciones técnicas claras, verificables y trazables, que sirvieran como fundamento para el diseño arquitectónico y la posterior implementación del sistema.

La consolidación permitió depurar, estructurar y normalizar los requerimientos, asegurando coherencia entre los actores identificados, los procesos modelados y las funcionalidades que el sistema debe ofrecer. Asimismo, facilitó la identificación de relaciones entre requerimientos funcionales, restricciones técnicas y criterios de calidad definidos para el sistema.

Los requerimientos fueron organizados en tres categorías principales:

- **Requerimientos funcionales:** describen las capacidades operativas del sistema.
- **Requerimientos no funcionales:** establecen criterios de calidad, desempeño y seguridad.
- **Historias de usuario,** que complementan la especificación funcional desde la perspectiva de los actores del sistema, utilizando un enfoque orientado al comportamiento.

Esta estructuración no sólo permite mantener trazabilidad entre las necesidades institucionales y la solución tecnológica propuesta, sino que también reduce ambigüedades y riesgos durante la fase de construcción.

A continuación, se presenta la consolidación de los requerimientos del sistema SIGHA.

5.2.5.1 Requerimientos funcionales. Los requerimientos funcionales del sistema SIGHA describen las capacidades operativas que el sistema debe ofrecer para soportar el proceso de gestión de horarios académicos en la Escuela de Ingeniería de Sistemas e Informática. Estos requerimientos fueron definidos a partir del análisis del proceso actual, la identificación de problemáticas estructurales y las necesidades manifestadas por los actores institucionales durante la fase de inicio.

Cada requerimiento funcional fue estructurado mediante un formato formal que incluye: descripción general, actor involucrado, precondiciones, criterios de aceptación bajo

enfoque BDD (Behavior Driven Development), reglas de negocio y restricciones. Este nivel de detalle permite garantizar claridad en la interpretación, verificabilidad en su implementación y trazabilidad hacia los componentes arquitectónicos definidos posteriormente.

Con el fin de facilitar su organización y referencia, los requerimientos funcionales fueron codificados bajo la nomenclatura RFXX. La **Tabla 11** presenta el resumen consolidado de los requerimientos funcionales identificados para el sistema SIGHA.

Tabla 11.

Requerimientos Funcionales - Descripción

Código	Nombre del Requerimiento	Descripción Resumida
RF01	<i>SIGHA – Aplicación Base y Menú Principal</i>	<i>Estructura base del sistema con navegación por módulos según el rol del usuario.</i>
RF02	<i>SIGHA – Gestión de Profesores, Roles y Permisos</i>	<i>Administración de usuarios del sistema, asignación de roles y control de acceso.</i>
RF03	<i>SIGHA – Gestión de Disponibilidades de Profesores</i>	<i>Registro, edición y envío de disponibilidad horaria docente, incluyendo consulta de estado.</i>
RF04	<i>SIGHA – Gestión de Áreas y Asignaturas</i>	<i>Creación y administración de áreas académicas y asignaturas asociadas.</i>

RF05	<i>SIGHA – Gestión Administrativa de Disponibilidades</i>	<i>Visualización, filtrado, aprobación y rechazo de disponibilidades docentes.</i>
RF06	<i>SIGHA – Gestión de Grupos Académicos</i>	<i>Creación y administración de grupos, asignación de docentes y configuración general.</i>
RF07	<i>SIGHA – Gestión de Calendario de Grupos</i>	<i>Asignación de franjas horarias con validación automática de solapamientos.</i>
RF08	<i>SIGHA – Consulta de Calendario General de la Escuela</i>	<i>Visualización consolidada del horario institucional del periodo activo.</i>
RF09	<i>SIGHA – Configuración y Parametrización del Sistema</i>	<i>Creación de semestres, configuración de franjas horarias y ajustes generales.</i>
RF10	<i>SIGHA – Configuración de Usuario</i>	<i>Gestión del perfil del usuario autenticado, cambio de contraseña y preferencias.</i>

Los diez requerimientos funcionales definidos cubren los siguientes dominios operativos del sistema:

- Estructura base y navegación principal de la aplicación.
- Gestión de usuarios, roles y permisos.
- Registro y gestión de disponibilidad docente.
- Administración de áreas académicas y asignaturas.
- Visualización y aprobación de disponibilidades.

- Creación y gestión de grupos académicos.
- Asignación y validación de calendarios de grupos.
- Consulta consolidada de horarios institucionales.
- Parametrización de semestres y configuración general del sistema.
- Gestión del perfil del usuario autenticado.

El detalle completo de cada requerimiento funcional, incluyendo sus criterios de aceptación y reglas de negocio asociadas, se presenta en el Apéndice A, con el fin de mantener la claridad estructural del capítulo principal y evitar redundancias extensas dentro del cuerpo del documento.

La consolidación formal de estos requerimientos permitió establecer una base funcional sólida que orientó directamente el diseño arquitectónico del sistema y la estructuración del modelo de datos.

5.2.5.2 Historias de Usuario. Con el fin de complementar la especificación formal de los requerimientos funcionales, se estructuraron Historias de Usuario (HUs) que describen el comportamiento esperado del sistema desde la perspectiva de los actores institucionales. Esta representación permite expresar las funcionalidades en un lenguaje orientado al usuario final, facilitando la comprensión funcional y la validación temprana de los escenarios de uso.

Las historias de usuario fueron formuladas bajo un enfoque de desarrollo orientado al comportamiento (BDD – Behavior Driven Development), incorporando criterios de aceptación

redactados en formato “Dado – Cuando – Entonces”. Este enfoque permitió definir con mayor precisión las condiciones bajo las cuales una funcionalidad se considera implementada correctamente, reduciendo ambigüedades y fortaleciendo la trazabilidad hacia la fase de construcción.

Cada historia de usuario se encuentra asociada a un requerimiento funcional previamente identificado (RF01–RF10), garantizando coherencia entre la especificación formal y la perspectiva operativa del actor. De esta manera, las historias no reemplazan los requerimientos funcionales, sino que los complementan, proporcionando un nivel adicional de detalle orientado a la implementación y validación.

A continuación, se presenta un ejemplo representativo correspondiente al requerimiento RF01 – Aplicación Base y Menú Principal. El conjunto completo de historias de usuario del sistema SIGHA se encuentra consignado en el Apéndice B.

Figura 17.

Formato de requerimientos funcionales - Historias de usuario

	PROCESO	PROYECTO DE GRADO	Código:	
	FORMATO	REQUISITOS FUNCIONALES	Versión:	01

	RF01-SIGHA-Aplicación-base-y-Menú-Principal		
--	---	--	--

1. DESCRIPCIÓN GENERAL	
Sistema	Sistema de Información para la Gestión de Horarios Académicos
Módulo	Transversal
Normatividad	
Tipo requerimiento	Nuevo X Mantenimiento

2. YO	
Como Coordinador Académico del Sistema de Información para la Gestión de Horarios Académicos	
Quiero visualizar dentro del Sistema de Información para la Gestión de Horarios Académicos con estructura de la Aplicación Base asociada al menú principal con sus respectivos módulos	
Para que los roles definidos puedan ingresar a realizar parametrización de semestres, Áreas y Materias, Grupos, Gestion de Usuarios, Solicitudes, información personal y Horarios de Grupos. visualización de: Disponibilidad horaria de los profesores y Dashboard	

3. CAMPOS DE ENTRADA				
Nombre	Tipo campo	Tamaño	Obligatoriedad	Descripción
N/A	N/A	N/A	N/A	N/A

4. CAMPOS DE SALIDA				
Nombre	Tipo campo	Tamaño	Obligatoriedad	Descripción
N/A	N/A	N/A	N/A	N/A

5. PRECONDICIONES	
Las precondiciones definen el estado en que se requiere que el sistema se encuentre, para poder ejecutar una funcionalidad específica	
PRE01	La funcionalidad Sistema de Información para la Gestión de Horarios Académicos debe estar disponible para garantizar el acceso y navegación de los usuarios.

6. CRITERIOS DE ACEPTACIÓN	
Su sintaxis se representa usando la Comprensión de los comportamientos esperados de un sistema a través de la escritura de escenarios de usuario (BDD Behavior Driven Development). Cada criterio de aceptación debe cumplir los objetivos Smart (Específico, medible, Alcanzable, realista, programarse para cumplirse en un tiempo específico).	
CRO1	<p>Dado que se accede al Sistema de Información para la Gestión de Horarios Académicos</p> <p>Cuando se visualiza la pantalla principal del Sistema de Información para la Gestión de Horarios Académicos</p> <p>Entonces se debe identificar la siguiente estructura con menús y módulos asociados:</p> <p>Menús y módulos por visualizar:</p>

La incorporación de historias de usuario en esta fase permitió establecer criterios claros de aceptación antes de la implementación, reducir riesgos de interpretación funcional y

asegurar que el desarrollo del sistema respondiera directamente a las necesidades operativas de la Escuela.

5.2.5.3 Requerimientos No Funcionales. Los requerimientos no funcionales del sistema SIGHA establecen las condiciones de calidad y las restricciones técnicas bajo las cuales debe operar la solución propuesta. A diferencia de los requerimientos funcionales, que describen qué debe hacer el sistema, los requerimientos no funcionales determinan cómo debe comportarse en términos de desempeño, seguridad, disponibilidad, escalabilidad y usabilidad.

Durante la fase de elaboración, estos requerimientos fueron revisados y consolidados con el fin de asegurar su coherencia con la arquitectura definida y con las tecnologías seleccionadas para la implementación. En particular, se validó la viabilidad técnica de cumplir con los criterios establecidos en relación con:

- Tiempo máximo de respuesta para operaciones críticas.
- Soporte multiusuario bajo carga simultánea.
- Seguridad en autenticación y manejo de sesiones.
- Disponibilidad del sistema durante periodos académicos críticos.
- Estándares de desarrollo y control de versiones.
- Consistencia visual y experiencia de usuario.

Estos requerimientos sirvieron como restricciones directas para el diseño arquitectónico del sistema, influyendo en decisiones como la adopción de una arquitectura cliente–servidor, el uso de Spring Boot para el backend, Vue.js para el frontend, PostgreSQL como gestor de base de datos y mecanismos de autenticación basados en JWT con cifrado bcrypt.

El listado detallado de los requerimientos no funcionales se encuentra consignado en la **Sección 5.1.7.2** y constituye un marco de referencia obligatorio para la fase de construcción, garantizando que la solución desarrollada no solo cumpla con las funcionalidades previstas, sino también con los estándares institucionales de calidad y seguridad.

5.2.6 Diseño Arquitectónico del Sistema SIGHA

Con el fin de garantizar la viabilidad técnica del proyecto y reducir riesgos durante la fase de construcción, se definió la arquitectura del sistema SIGHA a partir de los requerimientos consolidados en la sección anterior. Esta arquitectura establece la estructura general del sistema, la distribución de responsabilidades entre sus componentes y las tecnologías empleadas para su implementación.

El diseño arquitectónico se fundamenta en una arquitectura web de tipo cliente–servidor, donde el frontend y el backend se encuentran desacoplados y se comunican mediante servicios REST. Esta decisión permite asegurar escalabilidad, mantenibilidad y separación de responsabilidades, alineándose con los requerimientos no funcionales relacionados con rendimiento, seguridad y disponibilidad.

El sistema se compone de los siguientes elementos principales:

- Un cliente web desarrollado en Vue.js, responsable de la interacción con el usuario.
- Un servidor backend desarrollado en Java con Spring Boot, encargado de la lógica de negocio y validaciones.
- Un sistema de persistencia basado en PostgreSQL para el almacenamiento estructurado de la información académica.
- Un esquema de autenticación y autorización basado en JWT, con almacenamiento seguro de contraseñas mediante bcrypt.
- Un entorno de despliegue soportado mediante contenedores Docker y servidor NGINX para publicación del frontend.

La arquitectura fue diseñada bajo el principio de separación en capas (Controller–Service–Repository) dentro del backend, garantizando organización modular, facilidad de pruebas y claridad en la implementación de reglas de negocio.

A continuación, se presentan los diagramas que describen la arquitectura del sistema desde diferentes perspectivas estructurales.

5.2.6.1 Vista General de Arquitectura. La vista general de arquitectura del sistema SIGHA presenta una perspectiva macro de la solución tecnológica adoptada, mostrando la interacción entre el cliente web, el servidor de aplicaciones y la base de datos. Esta representación permite comprender la estructura global del sistema antes de profundizar en sus componentes internos.

El sistema fue diseñado bajo un modelo cliente–servidor desacoplado, donde el frontend y el backend operan como unidades independientes que se comunican mediante servicios REST sobre el protocolo HTTP/HTTPS. Esta separación facilita la escalabilidad del sistema, permite el mantenimiento independiente de cada componente y reduce el acoplamiento entre la interfaz de usuario y la lógica de negocio.

En esta arquitectura:

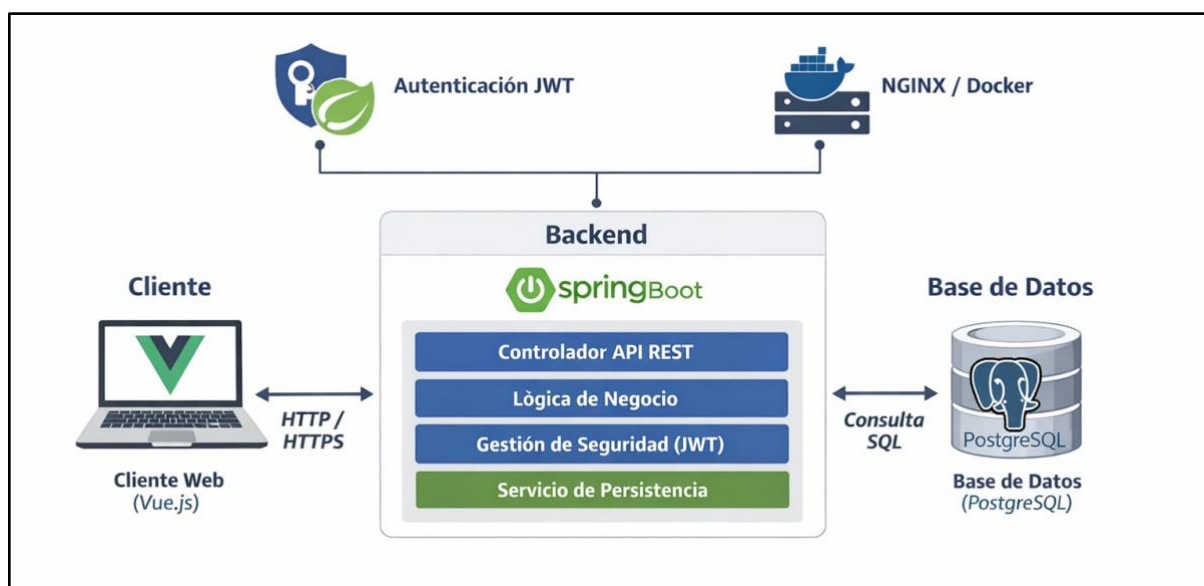
- El **cliente web**, desarrollado en Vue.js, gestiona la interfaz gráfica y la experiencia de usuario. Es responsable de enviar solicitudes al backend y representar la información recibida.
- El **backend**, implementado en Java con Spring Boot, centraliza la lógica de negocio, validaciones de reglas académicas, control de autenticación y autorización, así como la gestión de transacciones con la base de datos.
- La **base de datos PostgreSQL** almacena de manera estructurada la información relacionada con usuarios, docentes, grupos, disponibilidades, asignaturas y semestres.

- La comunicación entre frontend y backend se realiza mediante APIs REST protegidas mediante autenticación basada en JWT.

Esta vista arquitectónica permite evidenciar la separación clara entre presentación, lógica de negocio y persistencia de datos, cumpliendo con los principios de diseño orientados a modularidad y mantenibilidad establecidos en los requerimientos no funcionales del proyecto.

Figura 18.

Diagrama general cliente–servidor.



5.2.6.2 Diagrama de Componentes. El diagrama de componentes del sistema SIGHA presenta una vista estructural que descompone la solución en módulos funcionales principales, evidenciando cómo se organizan internamente las responsabilidades del sistema. A diferencia de la vista general, que muestra la interacción entre grandes bloques tecnológicos, esta representación permite identificar los subsistemas que implementan los requerimientos definidos previamente.

El backend del sistema fue organizado en componentes alineados con los dominios funcionales identificados en los requerimientos (RF01–RF10). Cada componente encapsula una responsabilidad específica y se comunica con los demás a través de interfaces bien definidas, evitando dependencias innecesarias.

Los principales componentes identificados son:

- **Módulo de Autenticación y Seguridad:** gestiona el inicio de sesión, validación de credenciales, generación y validación de tokens JWT, y control de acceso basado en roles.
- **Módulo de Gestión de Usuarios:** administra la creación, edición y activación de usuarios, así como la asignación de roles institucionales.
- **Módulo de Gestión Académica:** comprende la administración de áreas, asignaturas y grupos académicos.

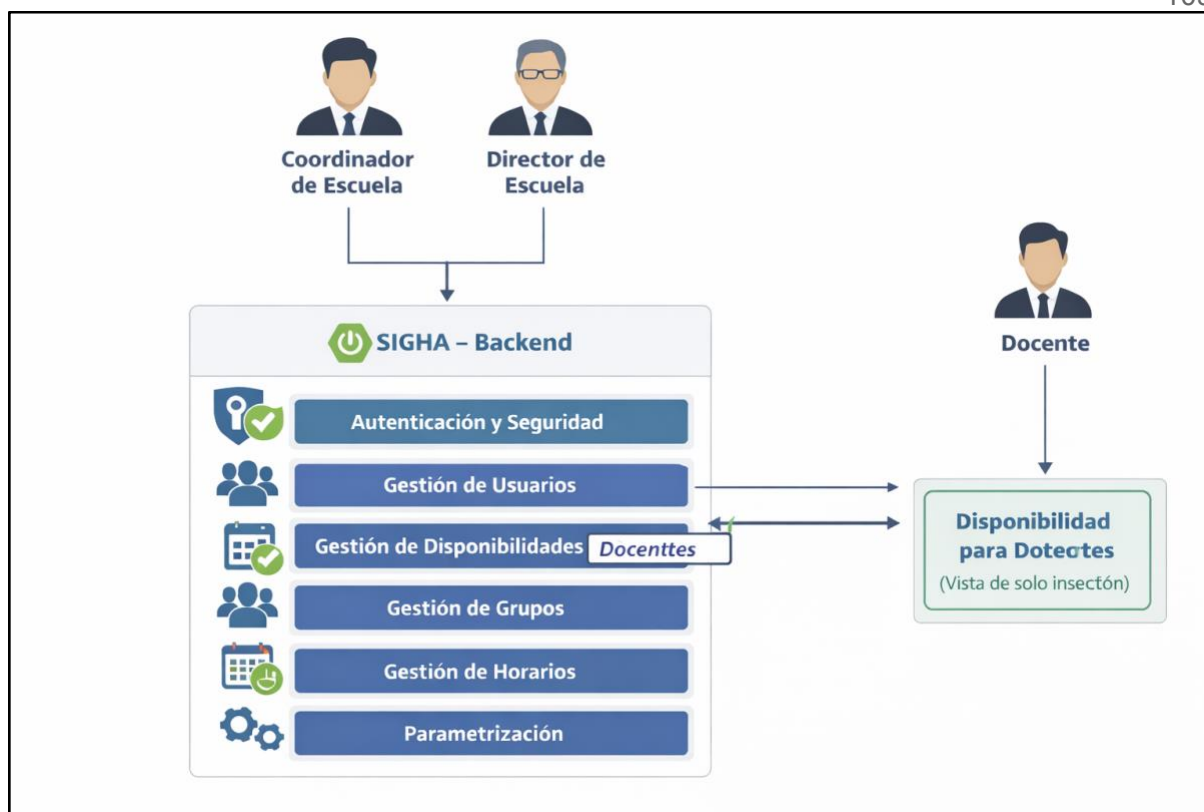
- **Módulo de Disponibilidades:** permite el registro, envío, consulta y aprobación de la disponibilidad docente.
- **Módulo de Gestión de Horarios:** implementa la asignación de franjas horarias y la validación automática de solapamientos.
- **Módulo de Parametrización:** gestiona la configuración de semestres académicos y parámetros generales del sistema.

Cada uno de estos componentes interactúa con la capa de persistencia para almacenar y recuperar información, manteniendo la lógica de negocio centralizada en el backend. Esta organización modular facilita la mantenibilidad, permite futuras ampliaciones del sistema y reduce el impacto de cambios en funcionalidades específicas.

El diagrama de componentes evidencia la correspondencia directa entre los requerimientos funcionales consolidados y los módulos implementados, garantizando trazabilidad entre especificación y diseño.

Figura 19.

Diagrama de Componentes



5.2.6.3 Diagrama de Capas del Backend. El backend del sistema SIGHA fue estructurado bajo un modelo de arquitectura por capas, siguiendo el patrón Controller – Service – Repository, el cual permite separar responsabilidades y facilitar la mantenibilidad del sistema.

La arquitectura interna se organiza de la siguiente manera:

1. Capa de Controladores (REST Controller)

Encargada de:

- Exponer los endpoints HTTP.
- Recibir solicitudes del frontend.
- Validar datos de entrada.
- Retornar respuestas en formato JSON.

2. Capa de Servicios (Service)

Encargada de:

- Implementar la lógica de negocio.
- Aplicar reglas del sistema.
- Coordinar operaciones entre repositorios.
- Ejecutar validaciones como solapamientos de horarios.

3. Capa de Persistencia (Repository - JPA)

Encargada de:

- Gestionar acceso a base de datos.
- Ejecutar consultas mediante Spring Data JPA.
- Manipular entidades.

4. Entidades (Entities)

Representan:

- Tablas de la base de datos.
- Relaciones entre objetos del dominio (User, Group, Subject, Availability, Schedule, etc.).

5. DTO y Mappers

Se emplean:

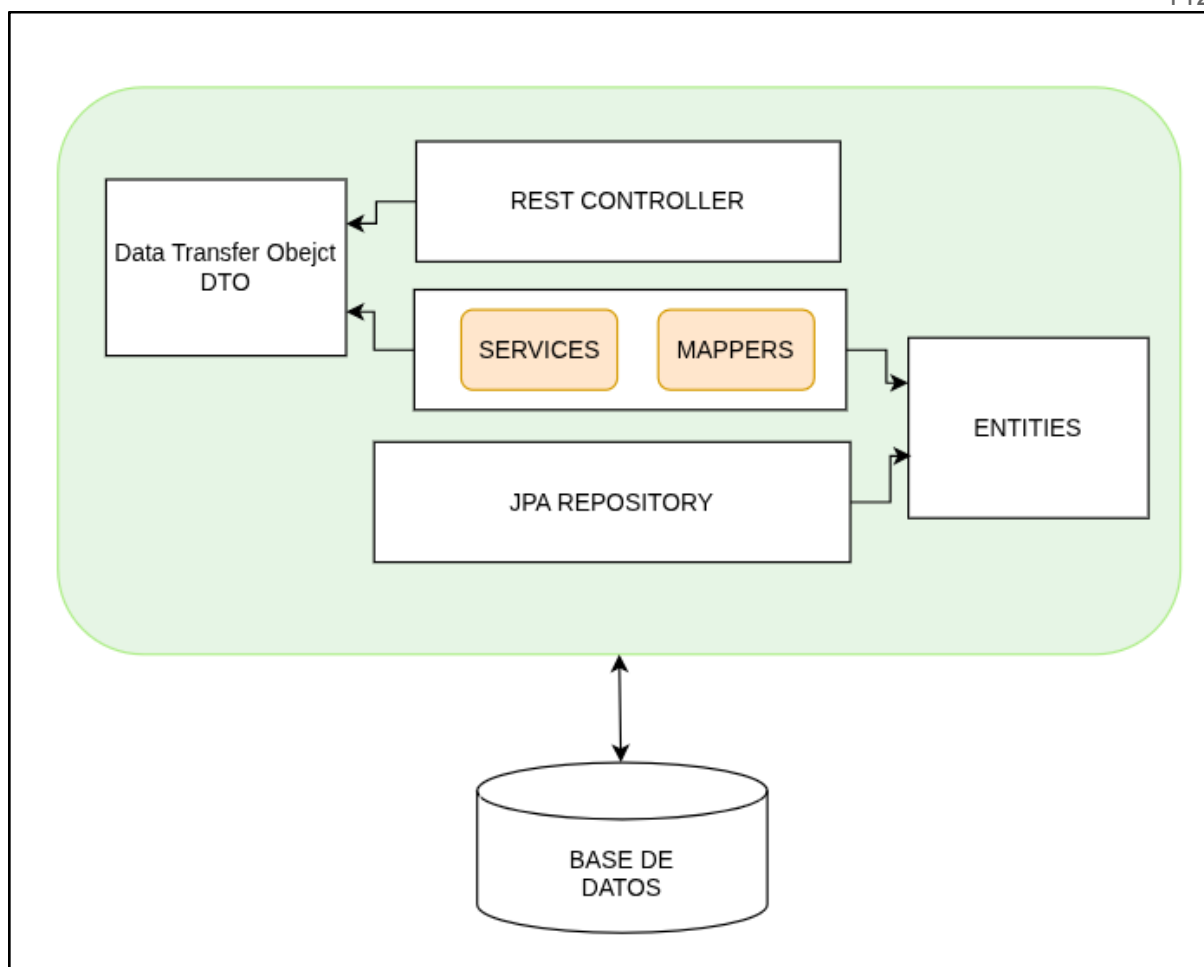
- DTO para transferencia segura de datos entre capas.
- Mappers para transformar Entities en DTO y viceversa.

Esta separación garantiza:

- Bajo acoplamiento.
- Alta cohesión.
- Escalabilidad futura.
- Fácil mantenimiento y pruebas unitarias.

Figura 20.

Diagrama de Capas del Backend – SIGHA



5.2.7 Modelo de Datos del Sistema

Con el fin de soportar estructuralmente los requerimientos funcionales definidos y garantizar la coherencia entre la lógica de negocio y la persistencia de información, se diseñó el modelo de datos del sistema SIGHA.

Este modelo constituye la base estructural sobre la cual se implementan los procesos de gestión de usuarios, disponibilidad docente, creación de grupos y asignación de horarios académicos.

El diseño del modelo de datos se desarrolló en dos niveles:

- Modelo Entidad–Relación conceptual, que representa las entidades del dominio y sus relaciones desde una perspectiva abstracta.
- Modelo lógico de base de datos, que traduce dichas entidades en estructuras implementables dentro del gestor PostgreSQL, incluyendo claves primarias, claves foráneas y tipos de datos.

La definición formal del modelo de datos permitió:

- Garantizar integridad referencial entre las entidades.
- Reducir redundancias.
- Facilitar la implementación de validaciones de negocio.
- Asegurar trazabilidad entre requerimientos funcionales y estructuras de almacenamiento.

A continuación, se presenta el modelo diseñado.

5.2.7.1 Modelo de Datos Conceptual y Lógico. El modelo de datos del sistema SIGHA fue diseñado para representar de manera estructurada las entidades que intervienen en la gestión académica y las relaciones existentes entre ellas. Dado que el modelo conceptual y el modelo lógico comparten la misma estructura base, se presenta una representación unificada que integra tanto la perspectiva abstracta del dominio como su traducción a una estructura implementable en PostgreSQL.

El modelo contempla las principales entidades del sistema, tales como:

- Usuario
- Rol
- Área
- Asignatura
- Grupo
- Semestre
- Disponibilidad
- Horario

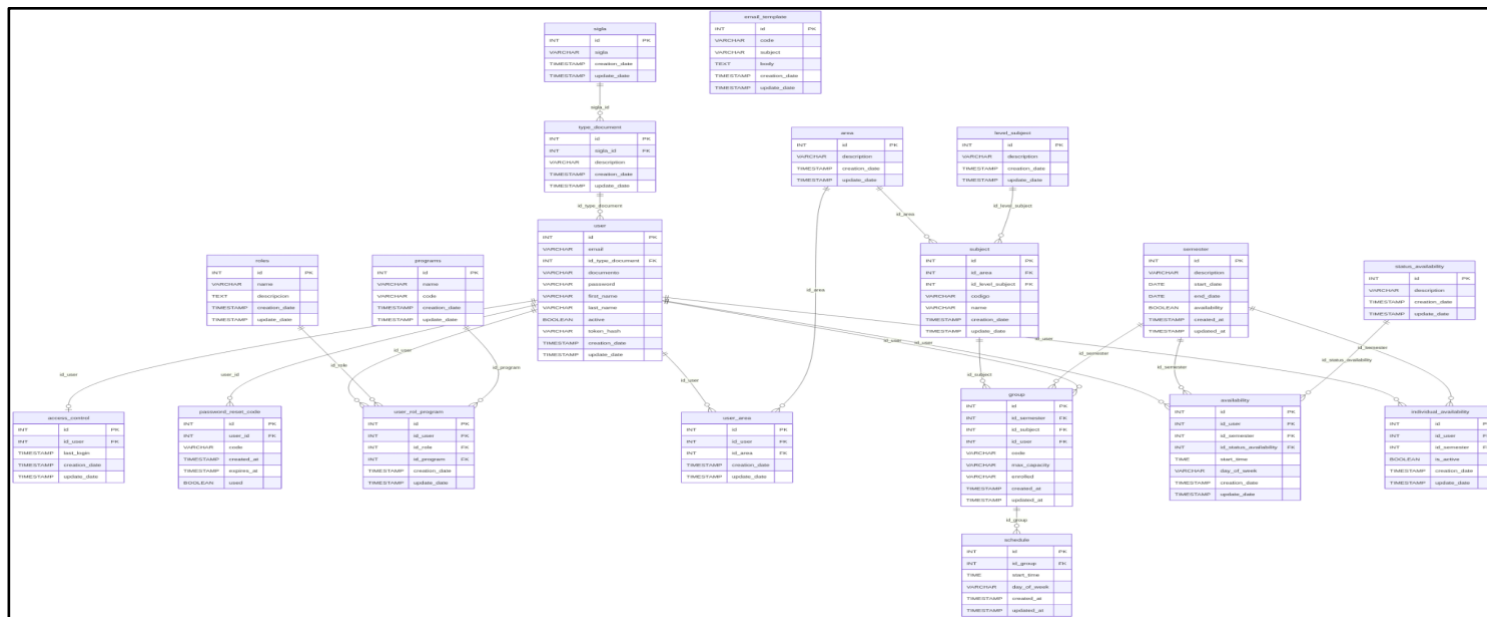
Asimismo, se definen las relaciones que permiten garantizar la coherencia académica, tales como:

- Asociación entre asignaturas y áreas.
- Vinculación de grupos a un semestre específico.

- Registro de disponibilidades por docente y periodo académico.
- Asignación de horarios a grupos bajo restricciones de solapamiento.

Figura 21.

Modelo de Datos Conceptual y Lógico del Sistema SIGHA

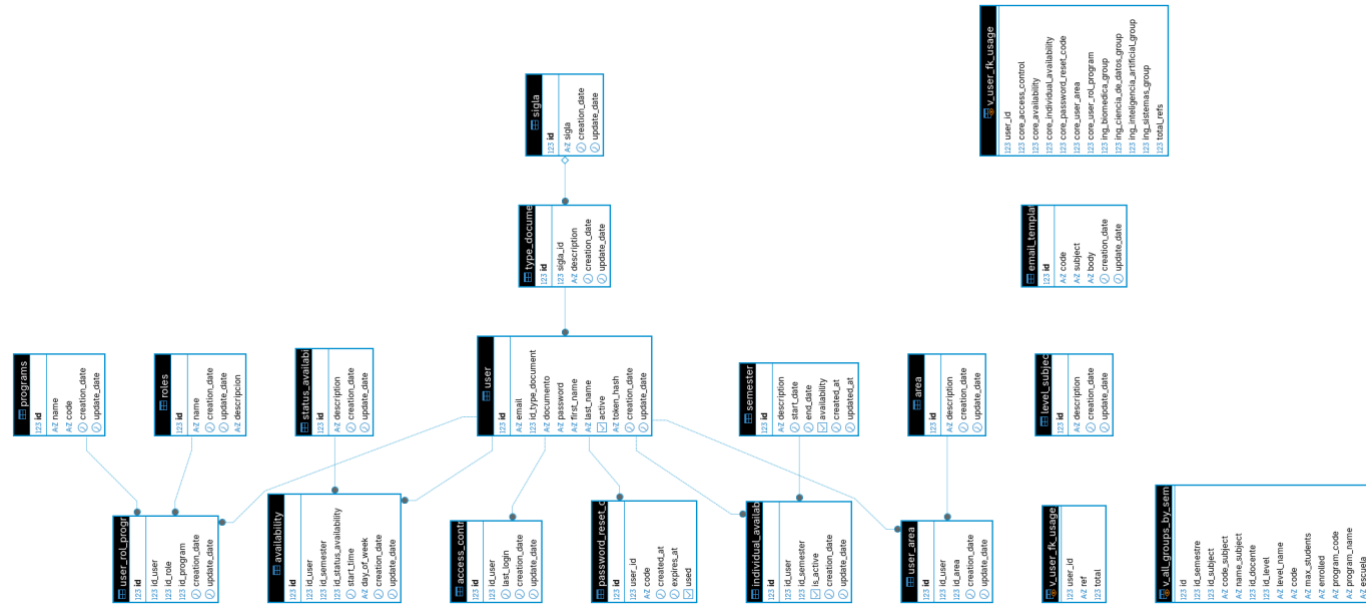


5.2.7.2 Modelo de Datos del Sistema SIGHA. El modelo de datos del sistema SIGHA fue diseñado con el objetivo de representar de manera estructurada las entidades que intervienen en la gestión académica y las relaciones existentes entre ellas. La estructura propuesta permite soportar los requerimientos funcionales definidos previamente, garantizando integridad referencial, consistencia de la información y normalización adecuada de las tablas.

El modelo implementado corresponde a un esquema relacional desarrollado para el gestor PostgreSQL. En él se definen explícitamente claves primarias (PK), claves foráneas (FK), tipos de datos y relaciones de cardinalidad entre las entidades, permitiendo su implementación directa en la base de datos del sistema.

Figura 22.

Modelo de Datos del Sistema SIGHA



El modelo se organiza en cuatro grandes dominios funcionales que responden directamente a la estructura del sistema.

Dominio de Seguridad y Usuarios

Incluye las entidades *user*, *roles*, *user_rol_program*, *user_area*, *access_control*, *password_reset_code*, *type_document* y *sigla*. Estas tablas permiten gestionar la autenticación, el control de acceso, la asignación de roles y la vinculación del usuario con programas académicos y áreas de conocimiento.

La entidad *user* actúa como eje central del sistema, estableciendo relaciones con los módulos académicos y administrativos, lo que permite mantener coherencia en la asignación de permisos y responsabilidades.

Dominio Académico

Comprende las entidades *area*, *level_subject*, *subject*, *group* y *semester*.

La estructura permite asociar asignaturas a un área académica, clasificar las asignaturas por nivel, vincular grupos a una asignatura específica y a un semestre determinado, y garantizar que cada grupo pertenezca a un único periodo académico. Esta organización asegura claridad en la oferta académica por semestre y facilita la planificación institucional.

Dominio de Disponibilidades

Incluye las entidades *availability*, *individual_availability* y *status_availability*. Estas tablas permiten registrar la disponibilidad declarada por cada docente para un semestre específico y controlar su estado dentro del flujo de aprobación. La relación entre disponibilidad, usuario y semestre garantiza trazabilidad y evita inconsistencias entre periodos académicos.

Dominio de Programación de Horarios

Está representado principalmente por la entidad *schedule*, la cual almacena la asignación efectiva de franjas horarias a cada grupo académico. Esta entidad se relaciona directamente con *group*, asegurando que no puedan existir horarios sin grupo asociado y permitiendo implementar validaciones de solapamiento desde la capa de servicios del backend.

El modelo implementa integridad referencial mediante el uso de claves foráneas que aseguran coherencia estructural entre las entidades. De esta manera, no es posible registrar disponibilidades sin usuario y semestre asociado, crear grupos sin asignatura, ni generar horarios sin grupo previamente definido. Esta estructura relacional permite mantener consistencia en la información, reducir redundancias y facilitar la aplicación de reglas de negocio dentro de la arquitectura por capas definida anteriormente.

5.2.7.2.1 Relaciones entre tablas. En esta sección se presenta el diccionario de datos del sistema SIGHA, el cual describe la estructura de las tablas pertenecientes a los esquemas core e ing_sistemas de la base de datos. Para cada tabla se detallan los campos, tipos de datos y descripciones funcionales, permitiendo comprender la organización de la información dentro del sistema.

Este diccionario de datos complementa el modelo de datos previamente presentado, proporcionando una visión detallada de la implementación física de la base de datos.

Tabla 12.

Diccionario de datos. esquema: core, tabla: user

<i>Campo</i>	<i>Tipo</i>	<i>Descripción</i>	<i>Registros</i>
<i>id</i>	<i>integer</i>	<i>Identificador único del usuario</i>	
<i>email</i>	<i>varchar(255)</i>	<i>Correo electrónico del usuario</i>	
<i>id_type_document</i>	<i>integer</i>	<i>Referencia al tipo de documento</i>	
<i>documento</i>	<i>varchar(255)</i>	<i>Número de documento del usuario</i>	
<i>password</i>	<i>varchar(255)</i>	<i>Contraseña encriptada del usuario</i>	
<i>first_name</i>	<i>varchar(255)</i>	<i>Nombre del usuario</i>	
<i>last_name</i>	<i>varchar(255)</i>	<i>Apellido del usuario</i>	
<i>active</i>	<i>boolean</i>	<i>Estado del usuario dentro del sistema</i>	
<i>token_hash</i>	<i>varchar(255)</i>	<i>Token de autenticación asociado</i>	
<i>creation_date</i>	<i>timestamp</i>	<i>Fecha de creación del registro</i>	
<i>update_date</i>	<i>timestamp</i>	<i>Fecha de última actualización</i>	

Seguidamente, se presenta la tabla correspondiente a los roles del sistema, la cual define los diferentes perfiles de usuario y sus características dentro de la aplicación.

Tabla 13.

Diccionario de datos. esquema: core, tabla: roles

Campo	Tipo	Descripción	Registros
<i>id</i>	<i>integer</i>	<i>Identificador único del rol</i>	
<i>name</i>	<i>varchar(255)</i>	<i>Nombre del rol en el sistema</i>	
<i>descripcion</i>	<i>varchar(250)</i>	<i>Descripción del rol</i>	<i>4</i>
<i>creation_date</i>	<i>timestamp</i>	<i>Fecha de creación del registro</i>	
<i>update_date</i>	<i>timestamp</i>	<i>Fecha de actualización del registro</i>	

A continuación, se describe la tabla correspondiente a los programas académicos, la cual almacena la información general de los programas disponibles en el sistema.

Tabla 14.

Diccionario de datos. esquema: core, tabla: programs

Campo	Tipo	Descripción	Registros
<i>id</i>	<i>integer</i>	<i>Identificador único del programa</i>	<i>4</i>
<i>name</i>	<i>varchar(255)</i>	<i>Nombre del programa académico</i>	
<i>code</i>	<i>varchar(100)</i>	<i>Código del programa</i>	
<i>creation_date</i>	<i>timestamp</i>	<i>Fecha de creación del registro</i>	
<i>update_date</i>	<i>timestamp</i>	<i>Fecha de actualización del registro</i>	

Posteriormente, se presenta la tabla correspondiente a las áreas académicas, la cual permite clasificar las asignaturas dentro del sistema.

Tabla 15.

Diccionario de datos. esquema: core, tabla: area

<i>Campo</i>	<i>Tipo</i>	<i>Descripción</i>	<i>Registros</i>
<i>id</i>	<i>integer</i>	<i>Identificador único del área</i>	<i>12</i>
<i>description</i>	<i>varchar(255)</i>	<i>Descripción del área académica</i>	
<i>creation_date</i>	<i>timestamp</i>	<i>Fecha de creación del registro</i>	
<i>update_date</i>	<i>timestamp</i>	<i>Fecha de actualización del registro</i>	

A continuación, se presenta la tabla de semestres, la cual define los periodos académicos utilizados en el sistema.

Tabla 16.

Diccionario de datos. esquema: core, tabla: semester

<i>Campo</i>	<i>Tipo</i>	<i>Descripción</i>	<i>Registros</i>
<i>id</i>	<i>integer</i>	<i>Identificador único del semestre</i>	<i>4</i>
<i>description</i>	<i>varchar(255)</i>	<i>Descripción del semestre</i>	
<i>start_date</i>	<i>date</i>	<i>Fecha de inicio del semestre</i>	
<i>end_date</i>	<i>date</i>	<i>Fecha de finalización del semestre</i>	
<i>availability</i>	<i>boolean</i>	<i>Estado de disponibilidad del semestre</i>	
<i>created_at</i>	<i>timestamp</i>	<i>Fecha de creación del registro</i>	
<i>updated_at</i>	<i>timestamp</i>	<i>Fecha de actualización del registro</i>	

De igual manera, se presenta la tabla de disponibilidades, la cual registra los horarios en los que los docentes pueden impartir clases.

Tabla 17.

Diccionario de datos. esquema: core, tabla: availability

<i>Campo</i>	<i>Tipo</i>	<i>Descripción</i>	<i>Registros</i>
--------------	-------------	--------------------	------------------

<i>id</i>	<i>integer</i>	<i>Identificador único del registro</i>	<i>153</i>
<i>id_user</i>	<i>integer</i>	<i>Referencia al usuario</i>	
<i>id_semester</i>	<i>integer</i>	<i>Referencia al semestre</i>	
<i>id_status_availability</i>	<i>integer</i>	<i>Estado de la disponibilidad</i>	
<i>start_time</i>	<i>time</i>	<i>Hora de inicio</i>	
<i>day_of_week</i>	<i>varchar(20)</i>	<i>Día de la semana</i>	
<i>creation_date</i>	<i>timestamp</i>	<i>Fecha de creación del registro</i>	
<i>update_date</i>	<i>timestamp</i>	<i>Fecha de actualización del registro</i>	

Asimismo, se presenta la tabla que relaciona los usuarios con las áreas académicas, permitiendo identificar su pertenencia dentro del sistema.

Tabla 18.

Diccionario de datos. esquema: core, tabla: user_area

Campo	Tipo	Descripción	Registros
<i>id</i>	<i>integer</i>	<i>Identificador único del registro</i>	<i>138</i>
<i>id_user</i>	<i>integer</i>	<i>Referencia al usuario</i>	
<i>id_area</i>	<i>integer</i>	<i>Referencia al área</i>	
<i>creation_date</i>	<i>timestamp</i>	<i>Fecha de creación del registro</i>	
<i>update_date</i>	<i>timestamp</i>	<i>Fecha de actualización del registro</i>	

De forma complementaria, se presenta la tabla que gestiona la relación entre usuarios, roles y programas académicos dentro del sistema.

Tabla 19.

Diccionario de datos. esquema: core, tabla: user_rol_program

<i>Campo</i>	<i>Tipo</i>	<i>Descripción</i>	<i>Registros</i>
<i>id</i>	<i>integer</i>	<i>Identificador único del registro</i>	<i>284</i>
<i>id_user</i>	<i>integer</i>	<i>Referencia al usuario</i>	
<i>id_role</i>	<i>integer</i>	<i>Referencia al rol</i>	
<i>id_program</i>	<i>integer</i>	<i>Referencia al programa</i>	
<i>creation_date</i>	<i>timestamp</i>	<i>Fecha de creación del registro</i>	
<i>update_date</i>	<i>timestamp</i>	<i>Fecha de actualización del registro</i>	

A continuación, se presenta la tabla de asignaturas del programa de Ingeniería de Sistemas, la cual contiene la información académica de cada materia.

Tabla 20.

Diccionario de datos. esquema: ing_sistemas, tabla: subject

<i>Campo</i>	<i>Tipo</i>	<i>Descripción</i>	<i>Registros</i>
<i>id</i>	<i>integer</i>	<i>Identificador único de la asignatura</i>	<i>72</i>
<i>id_area</i>	<i>integer</i>	<i>Referencia al área</i>	
<i>id_level_subject</i>	<i>integer</i>	<i>Referencia al nivel de la asignatura</i>	
<i>codigo</i>	<i>varchar(255)</i>	<i>Código de la asignatura</i>	
<i>name</i>	<i>varchar(255)</i>	<i>Nombre de la asignatura</i>	
<i>creation_date</i>	<i>timestamp</i>	<i>Fecha de creación del registro</i>	
<i>update_date</i>	<i>timestamp</i>	<i>Fecha de actualización del registro</i>	

Seguidamente, se presenta la tabla de grupos, la cual permite organizar las asignaturas en diferentes secciones dentro de cada semestre académico.

Tabla 21.

Diccionario de datos. esquema: *ing_sistemas*, tabla: *group*

<i>Campo</i>	<i>Tipo</i>	<i>Descripción</i>	<i>Registros</i>
<i>id</i>	<i>integer</i>	<i>Identificador único del grupo</i>	<i>355</i>
<i>id_semester</i>	<i>integer</i>	<i>Referencia al semestre</i>	
<i>id_subject</i>	<i>integer</i>	<i>Referencia a la asignatura</i>	
<i>id_user</i>	<i>integer</i>	<i>Referencia al docente</i>	
<i>code</i>	<i>varchar(50)</i>	<i>Código del grupo</i>	
<i>max_capacity</i>	<i>varchar(10)</i>	<i>Capacidad máxima</i>	
<i>enrolled</i>	<i>varchar(10)</i>	<i>Cantidad de inscritos</i>	
<i>created_at</i>	<i>timestamp</i>	<i>Fecha de creación del registro</i>	
<i>updated_at</i>	<i>timestamp</i>	<i>Fecha de actualización del registro</i>	

Finalmente, se presenta la tabla de horarios, la cual define la programación de las clases asociadas a cada grupo.

Tabla 22.

Diccionario de datos. esquema: *ing_sistemas*, tabla: *schedule*

<i>Campo</i>	<i>Tipo</i>	<i>Descripción</i>	<i>Registros</i>
<i>id</i>	<i>integer</i>	<i>Identificador único del horario</i>	<i>1457</i>
<i>id_group</i>	<i>integer</i>	<i>Referencia al grupo</i>	
<i>start_time</i>	<i>time</i>	<i>Hora de inicio</i>	
<i>day_of_week</i>	<i>varchar(20)</i>	<i>Día de la semana</i>	
<i>created_at</i>	<i>timestamp</i>	<i>Fecha de creación del registro</i>	
<i>updated_at</i>	<i>timestamp</i>	<i>Fecha de actualización del registro</i>	

5.2.8 Análisis y Gestión de Riesgos Técnicos

La fase de elaboración no sólo permitió definir la arquitectura y el modelo de datos del sistema SIGHA, sino también identificar los principales riesgos técnicos asociados a su implementación. Esta actividad fue fundamental para reducir la incertidumbre antes de iniciar la fase de construcción, en coherencia con el enfoque iterativo del proceso RUP.

El análisis de riesgos se realizó considerando los requerimientos funcionales y no funcionales definidos previamente, así como las decisiones tecnológicas adoptadas para el desarrollo del sistema. Los riesgos identificados se clasificaron en función de su impacto potencial sobre el rendimiento, la seguridad, la integridad de los datos y la viabilidad técnica del proyecto.

Riesgos asociados a la arquitectura cliente–servidor

Uno de los principales riesgos identificados fue la correcta comunicación entre el frontend desarrollado en Vue.js y el backend implementado en Spring Boot. Una configuración inadecuada de los servicios REST, manejo incorrecto de tokens JWT o errores en la configuración de CORS podrían impedir la autenticación adecuada de los usuarios o generar fallos en la transmisión de información. Para mitigar este riesgo, se definió una estructura clara de endpoints, manejo centralizado de seguridad mediante filtros JWT y pruebas tempranas de integración entre cliente y servidor.

Riesgos relacionados con la validación de solapamientos

El sistema requiere implementar validaciones automáticas que impidan la asignación de franjas horarias conflictivas para un mismo docente o grupo. La incorrecta implementación de estas reglas podría generar inconsistencias académicas. Como medida de mitigación, se estableció que las validaciones se implementarán en la capa de servicios, separadas de la lógica de presentación, y que se diseñarán consultas específicas en la capa de persistencia para detectar conflictos antes de confirmar una asignación.

Riesgos de integridad y consistencia de datos

Dado que el sistema maneja múltiples relaciones entre usuarios, grupos, asignaturas, semestres y disponibilidades, existía el riesgo de inconsistencias estructurales en la base de datos.

La mitigación se realizó mediante el diseño de un modelo relacional normalizado, con claves primarias y foráneas explícitas, restricciones de integridad y control de estados en entidades críticas como disponibilidad y semestre.

Riesgos de seguridad

El manejo de credenciales y sesiones representa un riesgo crítico. Un almacenamiento inadecuado de contraseñas o una mala gestión de tokens de sesión podría comprometer la seguridad institucional.

Para reducir este riesgo, se adoptó el uso de cifrado bcrypt para almacenamiento de contraseñas y autenticación basada en JWT con expiración controlada de tokens.

Riesgos de desempeño y carga simultánea

El sistema debe soportar múltiples accesos simultáneos durante periodos críticos de creación y ajuste de horarios. La degradación del rendimiento podría afectar la operación administrativa.

Como estrategia preventiva, se optó por una arquitectura desacoplada, consultas optimizadas en base de datos y separación clara de responsabilidades en el backend para facilitar la escalabilidad.

El análisis realizado durante esta fase permitió anticipar escenarios críticos antes de la implementación, establecer estrategias de mitigación tempranas y asegurar que la fase de construcción se desarrollará con una base técnica sólida. De esta manera, la fase de elaboración cumplió su propósito dentro del proceso RUP: reducir riesgos técnicos y consolidar la viabilidad estructural del sistema antes de su desarrollo definitivo.

Tabla 23.*Plan de Gestión de Riesgos Técnicos del Sistema SIGHA*

<i>Riesgo Identificado</i>	<i>Probabilidad</i>	<i>Impacto</i>	<i>Estrategia de Mitigación</i>
Sobrecarga del servidor por múltiples solicitudes concurrentes	<i>Media</i>	<i>Alta</i>	<i>Estructura base del sistema con navegación por módulos según el rol del usuario.</i>
Conflictos no detectados en la asignación de horarios	<i>Media</i>	<i>Alta</i>	<i>Implementación de validaciones automáticas en la capa Service antes de persistencia</i>
Pérdida de información por fallos en la base de datos	<i>Baja</i>	<i>Alta</i>	<i>Uso de respaldos periódicos y configuración de persistencia en PostgreSQL</i>
Vulnerabilidades de seguridad en autenticación	<i>Media</i>	<i>Alta</i>	<i>Implementación de JWT, cifrado bcrypt y transmisión segura HTTPS</i>
Inconsistencias por errores en el mapeo DTO–Entity	<i>Media</i>	<i>Media</i>	<i>Validaciones unitarias y pruebas de integración en servicios críticos</i>
Problemas de rendimiento en consultas complejas	<i>Media</i>	<i>Media</i>	<i>Indexación en base de datos y optimización de consultas JPA</i>
Fallos en despliegue por configuración de contenedores	<i>Baja</i>	<i>Media</i>	<i>Pruebas en entorno de desarrollo antes de producción y versionamiento en Git</i>

			<i>Capacitación previa y elaboración de videotutoriales</i>
	<i>Media</i>	<i>Media</i>	
Resistencia al cambio por parte del personal administrativo			

5.2.9 Prototipos visuales

Se desarrollaron prototipos visuales de las interfaces principales utilizando la herramienta Figma. Estos prototipos permitieron representar el flujo general de navegación y las pantallas clave del sistema antes de la implementación funcional.

Figura 23.

Prototipo – Inicio de sesión

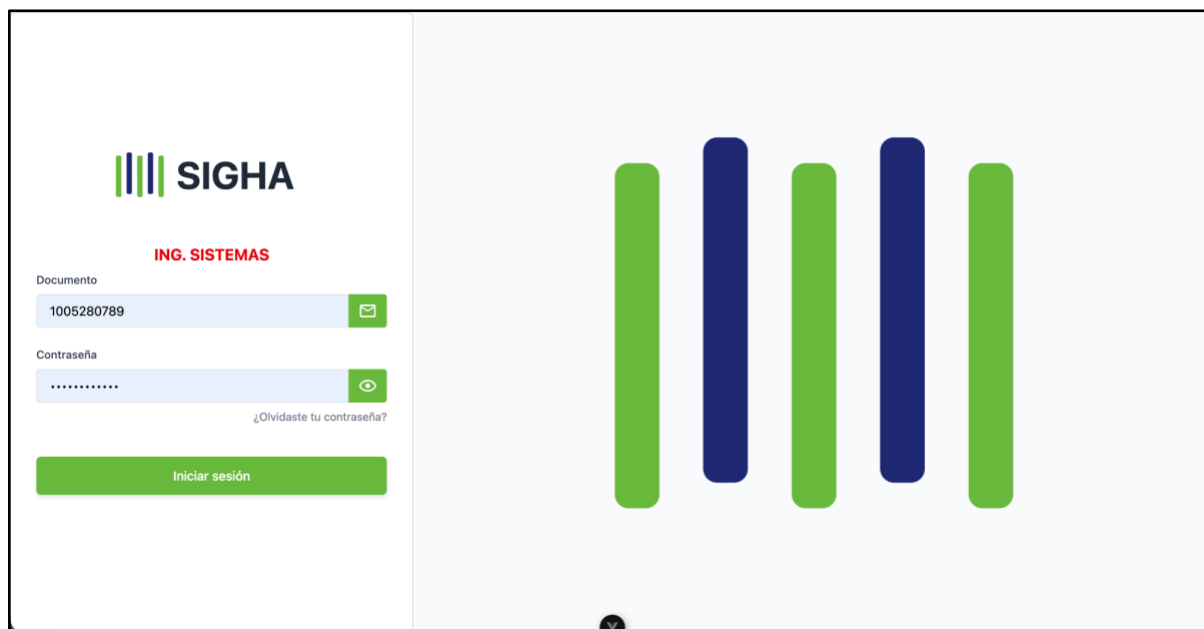


Figura 26.

Prototipo – Gestión de horarios

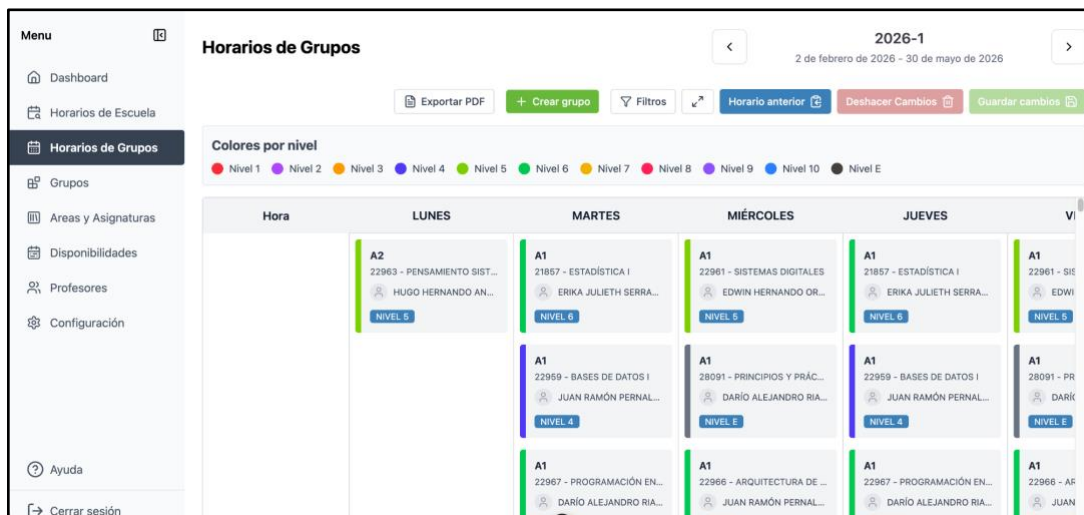


Figura 27.

Prototipo – Gestión de grupos

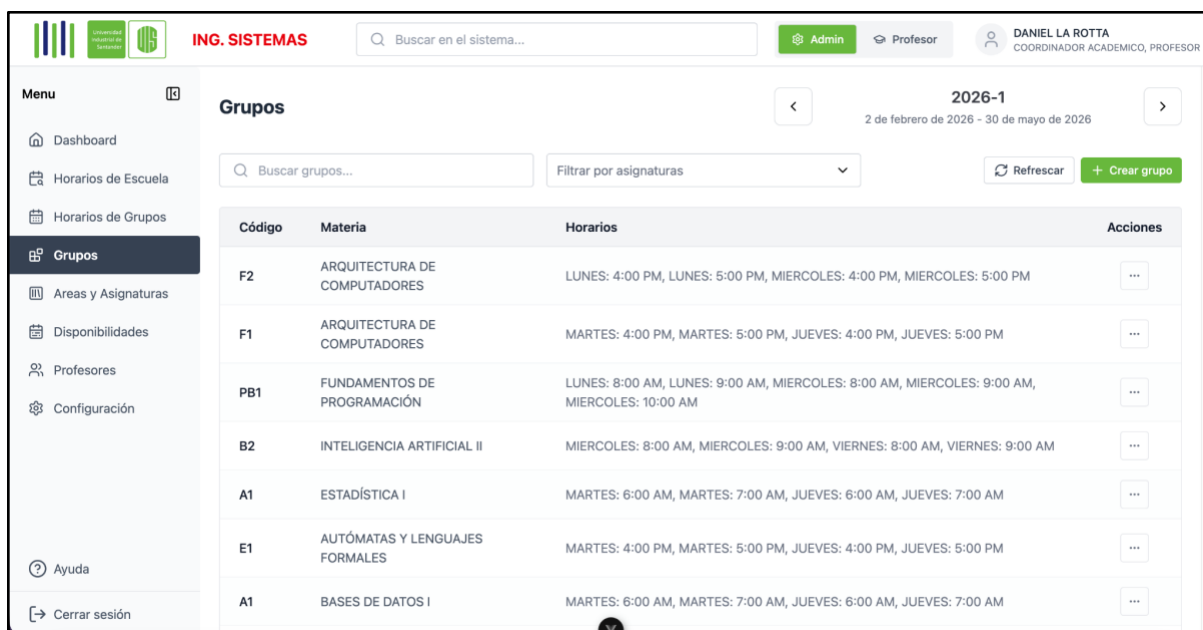
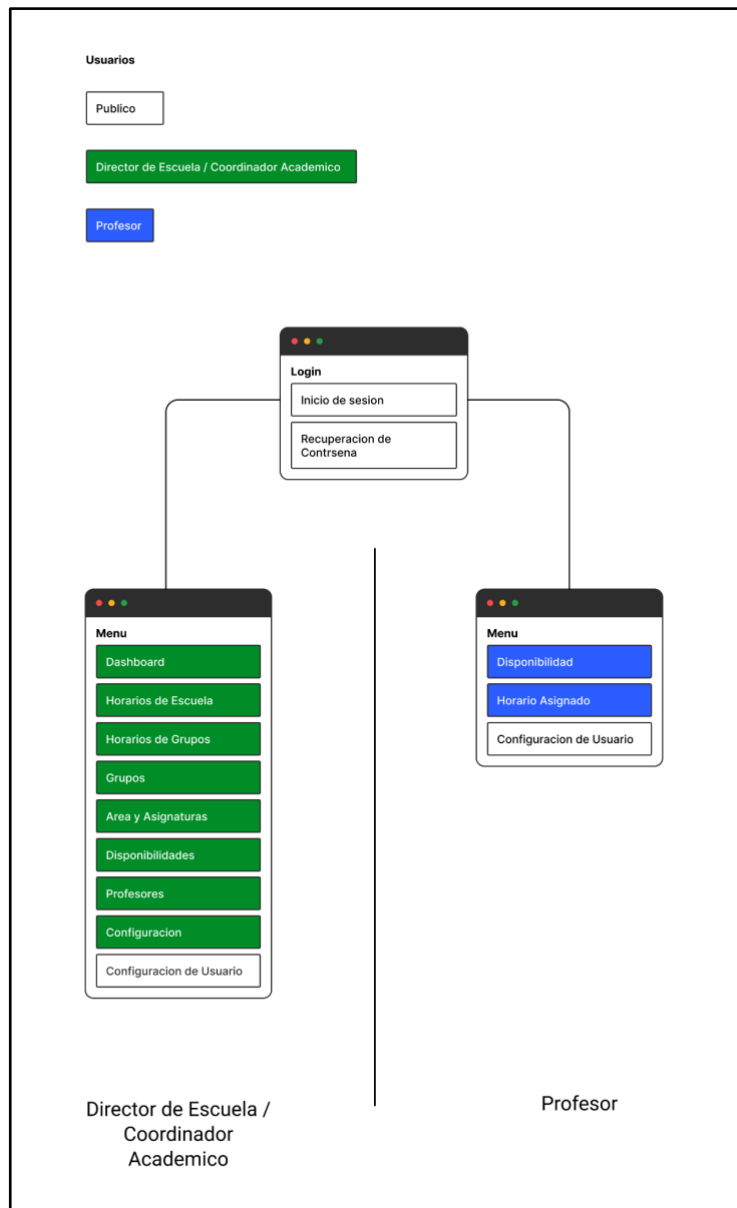


Figura 28.

Mapa de navegación del sistema



5.3 Fase de construcción

Durante esta fase, el énfasis se centró en materializar la solución propuesta, garantizando la coherencia entre los requerimientos definidos y las funcionalidades implementadas, así como la calidad técnica del software desarrollado.

5.3.1 Organización del desarrollo

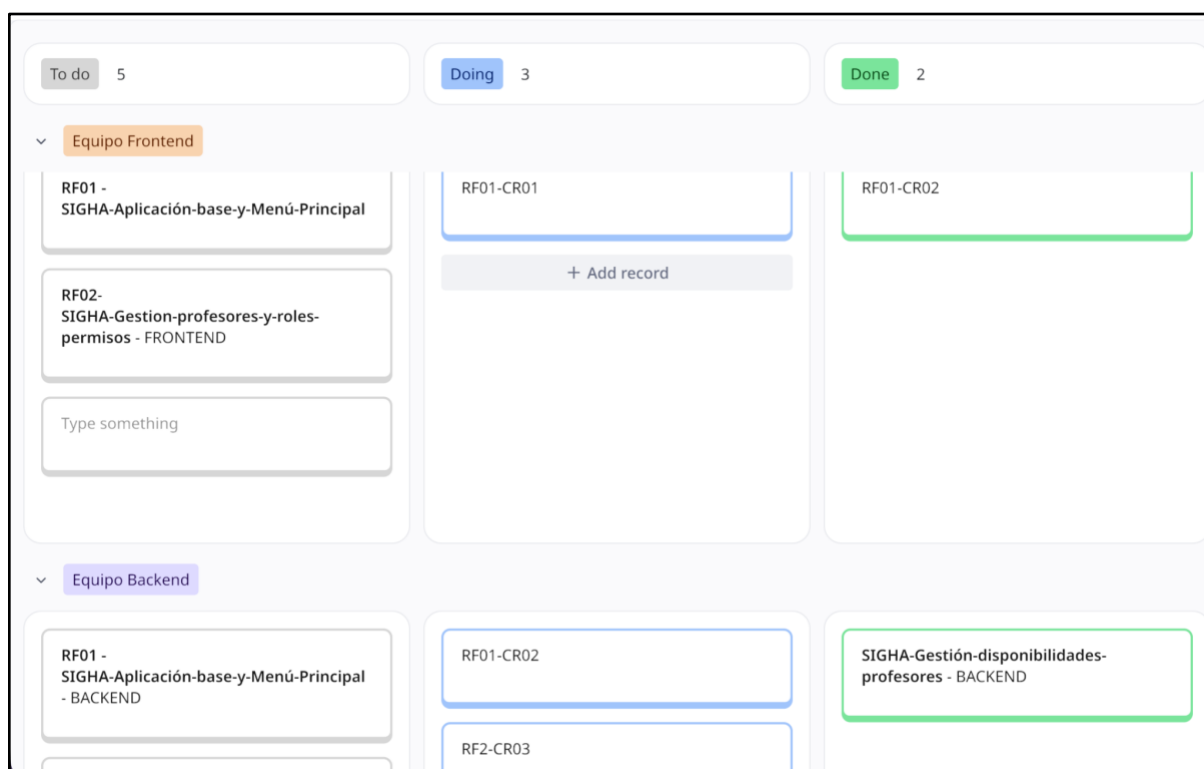
El desarrollo del sistema SIGHA se organizó siguiendo un enfoque iterativo e incremental, apoyado en el uso del marco de trabajo Kanban para la gestión de las actividades. Este enfoque permitió dividir el trabajo en tareas controlables, priorizar funcionalidades y realizar un seguimiento continuo del avance del proyecto.

Se utilizó un tablero Kanban con las columnas "Por hacer", "En progreso" y "Finalizado" para gestionar las actividades de desarrollo. Este tablero permite visualizar el estado de cada tarea, identificar bloqueos y priorizar funcionalidades en cada iteración.

La organización del desarrollo facilitó la incorporación progresiva de funcionalidades, la detección temprana de errores y la aplicación de ajustes a partir de los resultados obtenidos en las pruebas técnicas realizadas durante esta fase.

Figura 29.

Tablero Kanban para la asignación de tareas



5.3.2 Implementación de módulos del sistema

La implementación del sistema SIGHA se realizó de forma modular, permitiendo una separación clara de responsabilidades, favoreciendo la mantenibilidad y escalabilidad del sistema. Cada módulo fue desarrollado conforme a los requerimientos establecidos y validado de manera progresiva durante el proceso de construcción.

5.3.2.1 Módulo de recolección de disponibilidad docente. El módulo de recolección de disponibilidad docente fue implementado con el objetivo de facilitar el registro y la actualización de las franjas horarias en las cuales los docentes se encuentran disponibles para dictar clases. Este módulo permite centralizar la información de disponibilidad, reduciendo la dependencia de medios informales y mejorando la consistencia de los datos.

La información registrada a través de este módulo constituye un insumo para la gestión y validación de los horarios académicos en el sistema. El módulo incluye funcionalidades como la selección visual de franjas horarias mediante una interfaz tipo calendario, la posibilidad de reutilizar la disponibilidad del semestre anterior y la consulta del estado de aprobación de la disponibilidad enviada.

5.3.2.2 Módulo de gestión de horarios académicos. El módulo de gestión de horarios académicos permite la creación, edición y visualización de los horarios asociados a las asignaturas y grupos académicos de la Escuela. Este módulo fue desarrollado para soportar la organización de los grupos, la asignación de docentes y la administración de las franjas horarias definidas para cada periodo académico.

Asimismo, el sistema permite la reutilización de información de semestres anteriores como base para la generación de nuevos horarios, facilitando el proceso de planificación académica y reduciendo el tiempo requerido para su elaboración. El módulo incorpora una

vista de calendario interactivo que permite visualizar de forma consolidada las asignaciones por grupo, por docente o por la totalidad de la escuela.

5.3.2.3 Validaciones automáticas y reglas de negocio. Como parte del proceso de construcción, se implementaron validaciones automáticas orientadas a garantizar la coherencia de la información gestionada por el sistema. Estas validaciones permiten identificar conflictos como solapamientos de horarios, asignaciones inconsistentes y violaciones a las restricciones académicas definidas.

Las principales validaciones implementadas incluyen:

- Verificación de solapamiento de docente (un mismo docente no puede estar asignado a dos grupos en la misma franja).
- Verificación de solapamiento de grupo (un grupo no puede tener dos asignaturas en la misma franja).
- Validación de intensidad horaria mínima (las horas asignadas deben cumplir con la intensidad definida para la materia).
- Validación de unicidad de datos (correo electrónico de usuario, código de grupo).
- Validación de semestre activo requerido para operaciones dependientes.

Las reglas de negocio fueron implementadas de acuerdo con los lineamientos establecidos en la fase de elaboración, asegurando que el comportamiento del sistema refleje

las políticas académicas y administrativas de la Escuela. La tabla completa de validaciones se presenta en el capítulo de Resultados (véase *Tabla 23*).

5.3.3 Pruebas técnicas del sistema

Durante la fase de construcción se realizaron pruebas técnicas con el fin de verificar el correcto funcionamiento de los módulos desarrollados y asegurar la estabilidad del sistema. Estas pruebas permitieron identificar y corregir errores de manera temprana, contribuyendo a la calidad del software implementado.

5.3.3.1 Pruebas unitarias

Las pruebas unitarias se enfocan en verificar el funcionamiento de los componentes individuales del sistema, asegurando que cada módulo cumpliera con las funcionalidades esperadas de manera aislada. Estas pruebas permitieron validar las reglas de negocio y el manejo adecuado de la información.

Se ejecutaron pruebas unitarias para los servicios de autenticación, gestión de usuarios, registro de disponibilidad y creación de grupos, entre otros. Los resultados completos se presentan en el capítulo de Resultados (véase *Tabla 24*).

5.3.3.2 Pruebas de integración

Las pruebas de integración permitieron verificar la correcta interacción entre los diferentes módulos del sistema SIGHA, evaluando el flujo completo del proceso de gestión de

horarios académicos. Estas pruebas aseguraron que los datos se transfirieran de manera consistente entre los componentes del sistema.

Se evaluaron flujos integrados completos como el registro y aprobación de disponibilidad, la creación de grupos con docente, la asignación de horarios con validación, y el flujo completo de un semestre académico. Los resultados completos se presentan en el capítulo de Resultados (véase *Tabla 25*).

5.3.4 Entregables de la fase de construcción

Como resultado de la fase de construcción se generaron diversos entregables que evidencian la implementación del sistema SIGHA y el cumplimiento de los requerimientos definidos en la fase de elaboración.

5.3.5 Entregables de la fase de construcción

El desarrollo de la fase de construcción permitió implementar los módulos principales del sistema SIGHA y validar su funcionamiento técnico conforme a los requerimientos establecidos. Los resultados obtenidos evidencian la materialización de la solución propuesta y la correcta aplicación del diseño definido en la fase de elaboración.

Los productos generados en esta etapa constituyen la base para la puesta en marcha del sistema en el entorno real de la Escuela, actividad que se desarrolla en la fase de transición.

Asimismo, los resultados permanecen abiertos a ajustes y mejoras derivados de la validación con usuarios finales y de las pruebas de aceptación.

Tabla 24.

Entregable: Fase de Construcción

Entregable	Descripción
Código fuente modular	<i>Implementación del sistema bajo un enfoque modular con frontend Vue.js y backend Spring Boot, organizado en repositorio Git con ramas separadas para desarrollo y producción.</i>
Módulos Funcionales Implementados	<i>10 módulos funcionales correspondientes a los requerimientos RF01-RF10, completamente operativos.</i>
Validaciones Automáticas	<i>8 tipos de validaciones implementadas para garantizar coherencia de datos y prevención de solapamientos.</i>
Resultados de pruebas unitarias	<i>Casos de prueba unitaria ejecutados con resultado exitoso.</i>
Resultados de pruebas de integración	<i>Flujos integrados verificados con resultado exitoso.</i>

La fase de construcción permitió implementar el sistema SIGHA de acuerdo con los requerimientos y el diseño definidos en la fase de elaboración. El desarrollo se realizó de forma iterativa e incremental, apoyado en el marco de trabajo Kanban para la gestión de las actividades.

5.3.6 Organización del desarrollo

El desarrollo del sistema SIGHA se organizó siguiendo un enfoque iterativo e incremental. Se utilizó un tablero Kanban con las columnas "Por hacer", "En progreso" y "Finalizado" para gestionar las actividades de desarrollo, lo que permitió visualizar el estado de cada tarea, identificar bloqueos y priorizar funcionalidades (véase en la sección 4.2).

Las iteraciones se distribuyeron de la siguiente manera:

Tabla 25.

Iteraciones de desarrollo

<i>Iteración</i>	<i>Enfoque</i>	<i>Módulos/Funcionalidades</i>
<i>Iteración 1</i>	<i>Infraestructura base</i>	<i>Configuración del proyecto (Vue.js + Spring Boot + PostgreSQL), autenticación JWT, estructura de navegación.</i>
<i>Iteración 2</i>	<i>Gestión de usuarios y parametrización</i>	<i>CRUD de usuarios, asignación de roles, gestión de semestres, configuración del sistema.</i>

<i>Iteración 3</i>	<i>Áreas, asignaturas y disponibilidad</i>	<i>CRUD de áreas y materias, módulo de registro de disponibilidad docente, módulo de gestión de disponibilidades (administrativo).</i>
<i>Iteración 4</i>	<i>Grupos y horarios</i>	<i>Creación de grupos, asignación de docentes, asignación de franjas horarias con validaciones de solapamiento.</i>
<i>Iteración 5</i>	<i>Consultas, dashboard y ajustes</i>	<i>Calendario de escuela, dashboard con estadísticas, refinamiento de interfaz, pruebas de integración.</i>

Este es el cronograma que se establece para el desarrollo de SIGHA junto con todas las fases de la metodología RUP.

Figura 30.

Cronograma

FASES METODOLOGIA RUP	ACTIVIDADES	4 SEMANAS	8 SEMANAS	12 SEMANAS	16 SEMANAS
Fase de inicio	Se realizarán entrevistas con el personal administrativo y docente.	■			
	Se documentaron los procesos actuales de gestión de horarios	■	■		
	Se delimitaron los alcances del sistema.		■	■	
Fase de elaboracion	Identificar los riesgos criticos en las seguridad del sistema		■	■	
	Construir un plan técnico que servirá como guía para el desarrollo posterior.			■	■
	Asegurar la viabilidad técnica del proyecto			■	■
Fase de construccion	Recolectar la disponibilidad docente			■	■
	Se construirá la validación automática de solapamientos.			■	■
	Se implementará la arquitectura diseñada.				■
	Se ejecutarán pruebas unitarias para verificar el funcionamiento de los componentes.				■
Fase de transicion	Se capacitará al equipo administrativo y docente en el uso del software				■
	Se realizarán pruebas de aceptación para validar que cumpla con los requerimientos institucionales.				■

Como resultado se cumplió con el cronograma y con el tiempo de entrega.

5.3.7 Implementación de módulos del sistema

La implementación se realizó de forma modular, permitiendo una separación clara de responsabilidades. Cada módulo fue desarrollado conforme a los requerimientos establecidos y validado de manera progresiva.

Tabla 26.

Descripción de Historias de usuarios

<i>Módulo</i>	<i>RF</i>	<i>Módulos/Funcionalidades</i>
<i>Aplicación base y menú principal</i>	<i>RF01</i>	<i>Estructura de navegación con sidebar, header con logo y buscador, control de acceso por rol.</i>
<i>Gestión de usuarios y roles</i>	<i>RF02</i>	<i>CRUD de usuarios con formulario de registro, asignación de roles (Director, Coordinador, Profesor), activación/desactivación de cuentas.</i>
<i>Disponibilidad (profesor)</i>	<i>RF03</i>	<i>Interfaz tipo calendario para que el docente seleccione sus franjas horarias disponibles, con opción de reutilizar las del semestre anterior.</i>
<i>Áreas y asignaturas</i>	<i>RF04</i>	<i>CRUD de áreas de conocimiento y asignaturas asociadas, con</i>

<i>Disponibilidad(administrativo)</i>	<i>RF05</i>	<i>validación de nombres únicos y asociación área-materia.</i> <i>Vista consolidada de todas las disponibilidades docentes, con filtros por área y materia. Acciones de aprobación y rechazo con observaciones.</i>
<i>Gestión de grupos</i>	<i>RF06</i>	<i>Calendario interactivo de horarios por grupo, con asignación de franjas horarias mediante drag-and-drop o selección manual. Validación automática de solapamientos en tiempo real.</i>
<i>Horarios de grupos</i>	<i>RF07</i>	<i>Calendario interactivo de horarios por grupo, con asignación de franjas horarias mediante drag-and-drop o selección manual. Validación automática de solapamientos en tiempo real.</i>
<i>Horarios de escuela</i>	<i>RF08</i>	<i>Vista consolidada del calendario general con todos los horarios del semestre activo, filtrable por grupo, docente o materia.</i>
<i>Configuración del sistema</i>	<i>RF09</i>	<i>Parametrización de semestres académicos (crear, activar, desactivar), configuración de franjas horarias base.</i>
<i>Configuración de usuario</i>	<i>RF10</i>	<i>Perfil del usuario autenticado con opción de cambio de contraseña y consulta de información personal.</i>

5.3.8 Validaciones automáticas implementadas

Como parte del proceso de construcción, se implementaron validaciones automáticas orientadas a garantizar la coherencia de la información gestionada por el sistema. Estas validaciones permiten identificar conflictos en tiempo real y evitar asignaciones inconsistentes.

Tabla 27.

Validaciones automáticas implementadas

<i>Validación</i>	<i>Descripción</i>
<i>Solapamiento de docente</i>	<i>Verifica que un mismo docente no esté asignado simultáneamente a dos grupos distintos en la misma franja horaria.</i>
<i>Solapamiento de grupo</i>	<i>Verifica que un mismo grupo académico no tenga dos asignaturas programadas en la misma franja horaria.</i>

Disponibilidad docente *Verifica que la franja horaria asignada al grupo se encuentre dentro de las franjas de disponibilidad aprobadas del docente correspondiente.*

5.3.9 Pruebas técnicas del sistema

Durante la fase de construcción se realizaron pruebas técnicas para verificar el correcto funcionamiento de los módulos desarrollados.

5.3.9.1 Pruebas unitarias. Las pruebas unitarias se enfocan en verificar el funcionamiento de los componentes individuales del sistema, en particular la lógica de validación de solapamientos y las reglas de negocio.

Tabla 28.

Pruebas unitarias

<i>Módulo</i>	<i>Componente</i>	<i>Casos de Prueba</i>	<i>Resultado</i>
<i>Autenticación</i>	<i>AuthService – login</i>	<i>Credenciales válidas, credenciales inválidas, usuario inactivo</i>	<i>3/3 Aprobados</i>
<i>Autenticación</i>	<i>JwtService – generación y validación</i>		<i>3/3 Aprobados</i>

		<i>Token válido, token expirado, token manipulado</i>	<i>3/3 Aprobados</i>
Usuarios	<i>UsuarioService – crear</i>		
		<i>Datos válidos, correo duplicado, campos faltantes</i>	<i>3/3 Aprobados</i>
Disponibilidad	<i>DisponibilidadService – registrar</i>		
		<i>Franjas válidas, franjas superpuestas del mismo docente, semestre inactivo.</i>	<i>3/3 Aprobados</i>
Grupos	<i>GrupoService – crear</i>		
		<i>Datos válidos, código duplicado, materia inexistente</i>	<i>3/3 Aprobados</i>
Horarios	<i>ValidacionService – solapamiento docente</i>	<i>Sin conflicto, conflicto en misma franja, conflicto parcial de franja</i>	
			<i>2/2 Aprobados</i>
Horarios	<i>ValidacionService – solapamiento grupo</i>	<i>Sin conflicto, conflicto en misma franja</i>	
			<i>2/2 Aprobados</i>
Horarios	<i>ValidacionService – disponibilidad docente</i>	<i>Franja dentro de disponibilidad, franja fuera de disponibilidad</i>	
			<i>2/2 Aprobados</i>
Horarios	<i>HorarioService – asignar</i>	<i>Asignación válida, solapamiento detectado.</i>	

5.3.9.2 Pruebas de integración. Las pruebas de integración verificaron la correcta interacción entre los diferentes módulos del sistema SIGHA, evaluando el flujo completo del proceso de gestión de horarios académicos.

Tabla 29.

Pruebas de integración

<i>Flujo integrado</i>	<i>Descripción</i>	<i>Módulos involucrados</i>	<i>Resultado</i>
Registro y aprobación de disponibilidad	Un profesor registra su disponibilidad → El coordinador la visualiza → La aprueba	Autenticación, Disponibilidad (profesor), Disponibilidad (admin)	Aprobado
Creación de grupo con docente	Se crea una materia → Se crea un grupo → Se asigna un docente al grupo	Áreas y asignaturas, Grupos, Usuarios	Aprobado
Asignación de horario con validación	Se crea un grupo → Se asigna una franja → El sistema valida	Grupos, Horarios, Validación	Aprobado

Detección de solapamiento de docente	<i>solapamiento → Se confirma la asignación</i>	<i>Aprobado</i>
Flujo completo de semestre	<i>Se asigna una franja al docente X en grupo A → Se intenta asignar la misma franja al docente X en grupo B → El sistema rechaza la asignación</i>	<i>Aprobado</i>
	<i>Se crea el semestre → Se registran disponibilidades → Se crean grupos → Se asignan horarios → Se consulta el calendario</i>	<i>Parametrización, Disponibilidad, Grupos, Horarios, Calendario</i>

5.3.10 Resumen del producto de software

El sistema SIGHA se compone de dos aplicaciones independientes: una aplicación de interfaz de usuario (frontend) y una aplicación de servicios (backend), cada una alojada en su propio repositorio de código fuente.

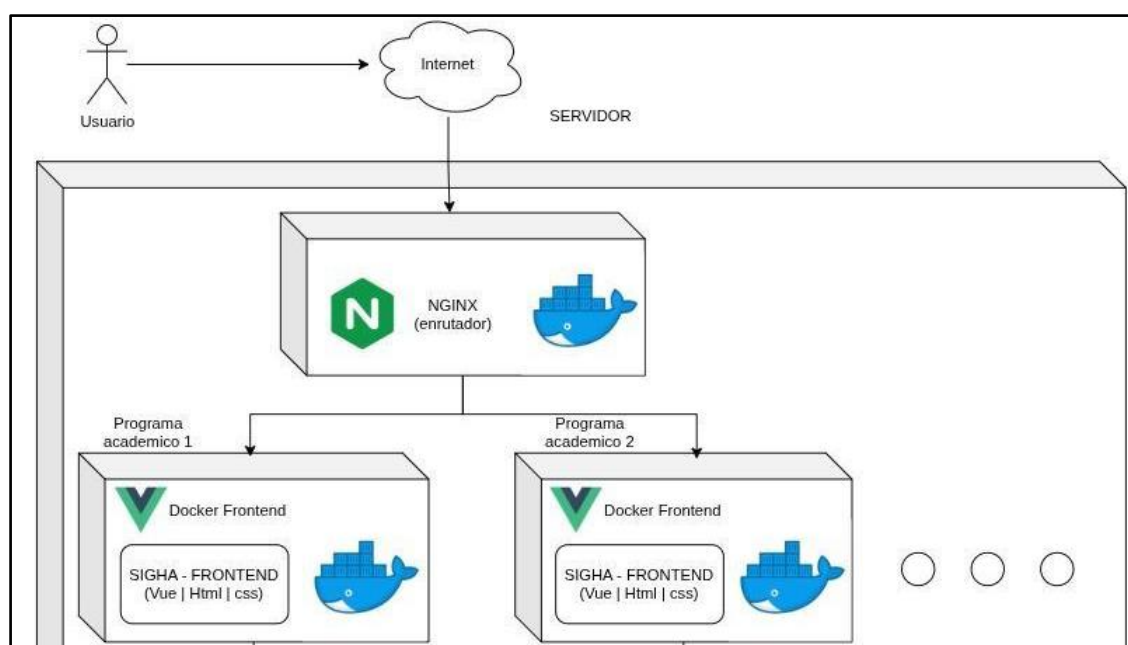
Frontend

La interfaz de usuario fue desarrollada con *Vue.js 3* y *TypeScript*, empaquetada con *Vite 6* y estilizada mediante *Tailwind CSS 4*. El estado global se gestiona con *Pinia 3* y la navegación con *Vue Router 4*. Se emplearon *Chart.js* para los gráficos del dashboard, *jwt-decode* para el

manejo de sesiones. En producción, la aplicación se sirve a través de *Nginx* dentro de un contenedor *Docker*.

Figura 31.

Diagrama de arquitectura Frontend



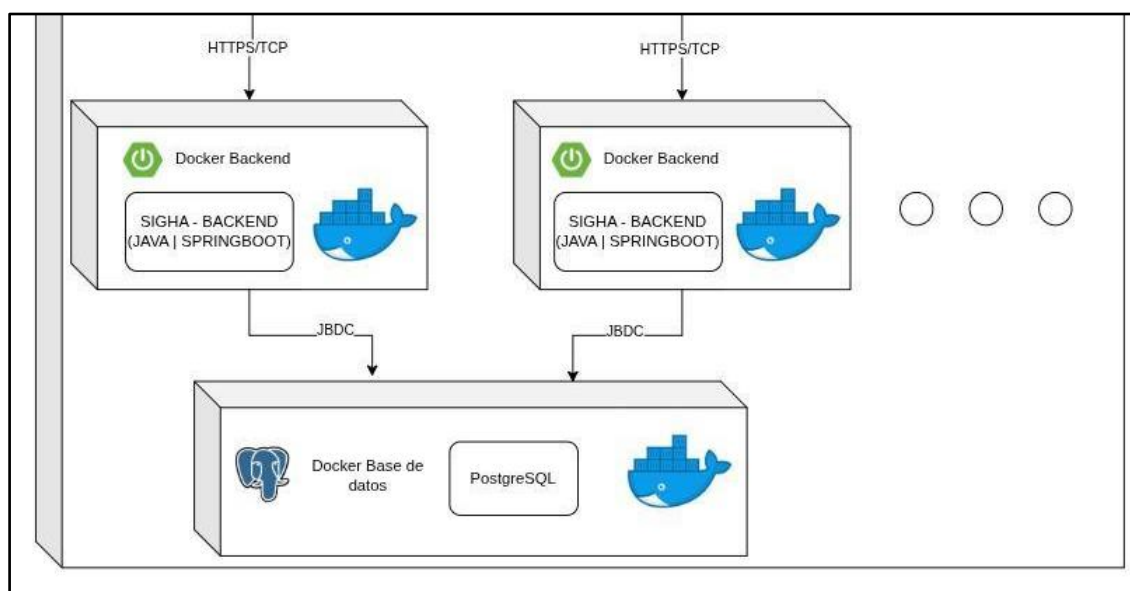
El frontend expone los siguientes módulos funcionales: autenticación con inicio de sesión por documento y contraseña, cambio obligatorio de contraseña en el primer acceso y recuperación por correo; un dashboard con métricas de grupos, gráficos de crecimiento semestral y estadísticas por materia; registro de disponibilidad docente mediante un calendario semanal interactivo con cuatro estados (disponible, preferiblemente no, no disponible, sin definir); gestión de usuarios con asignación de roles y activación de cuentas; administración de

áreas, materias y niveles académicos; gestión de grupos con asignación de profesores y validación de disponibilidad; un calendario de horarios con arrastrar y soltar, detección de conflictos; procesamiento de solicitudes con flujo de aprobación y notificaciones; y configuración de semestres, tipos de documento y parámetros generales.

El repositorio acumula 106 commits y se gestionó con tres ramas: dev para desarrollo, pre-prod para validación previa y master como versión estable de producción.

Backend

Los servicios fueron desarrollados con *Java 17* y *Spring Boot 3.3.2*, utilizando *Spring Web* para la *API REST*, *Spring Data JPA* para el acceso a datos con *PostgreSQL*, y *Spring Security* junto con *JWT* para la autenticación y autorización. La *API* se documenta automáticamente con *Swagger (Springdoc OpenAPI 2.5)* y las notificaciones se envían mediante *Spring Boot Mail*. El proyecto se compila con *Maven* y se despliega en contenedores *Docker* optimizados mediante un *Dockerfile multi-stage*.

Figura 32.*Diagrama de arquitectura Backend*

El backend gestiona la seguridad por roles (Director de Escuela, Coordinador Académico y Profesor), la administración de usuarios y estructura académica (áreas, materias, semestres), el registro y aprobación de disponibilidad docente y el envío de notificaciones por correo electrónico.

A su vez, el backend implementa una arquitectura multi-tenant por programa académico: una misma base de datos *PostgreSQL* contiene un schema compartido (core) con la información común (usuarios, roles, áreas) y un schema independiente por cada programa

(por ejemplo, *ing_sistemas*, *ing_biomedica*). Cada programa se despliega como una instancia separada del backend en un puerto distinto, configurado mediante variables de entorno sin necesidad de recompilar el código, lo que facilita incorporar nuevos programas.

El repositorio acumula 156 commits y se gestionó con dos ramas: *develop* para desarrollo activo y *production* como versión estable desplegada.

Estrategia de ramas

En el frontend, las funcionalidades avanzan de dev a pre-prod y finalmente a master. En el backend, el flujo va directamente de develop a production. En ambos repositorios se emplearon ramas de feature para el desarrollo aislado de cada funcionalidad, integradas mediante pull requests a la rama de desarrollo correspondiente.

5.4 Resultados de la Fase de Transición

Durante esta fase se realizaron actividades orientadas a la validación del sistema con usuarios reales, la capacitación de los actores involucrados y la preparación de la documentación necesaria para su uso y administración.

5.4.1 Preparación para la puesta en marcha

Como parte de la fase de transición, se llevaron a cabo actividades preliminares orientadas a preparar el sistema SIGHA para su despliegue en el entorno institucional. Estas

actividades incluyen la verificación del estado del sistema, la revisión de configuraciones básicas y la validación de que los módulos implementados se encontrarán operativos de acuerdo con los escenarios definidos.

La preparación para la puesta en marcha permitió reducir riesgos asociados al uso del sistema en un entorno real y asegurar condiciones mínimas de estabilidad antes de su utilización por parte de los usuarios finales.

El sistema fue desplegado en un entorno de producción que comprende una aplicación frontend Vue.js servida como SPA, una aplicación backend Spring Boot como servicio, una instancia dedicada de PostgreSQL y comunicación segura mediante HTTPS.

5.4.2 Capacitación de usuarios

La capacitación de los usuarios finales constituyó un aspecto fundamental de la fase de transición, ya que permitió facilitar la adopción del sistema SIGHA por parte del personal administrativo y docente. Las actividades de capacitación estuvieron orientadas a explicar el propósito del sistema, su funcionamiento general y las principales acciones que pueden realizar los diferentes actores.

Se elaboraron videotutoriales del uso del sistema, cubriendo temas como inicio de sesión, registro de disponibilidad, gestión de usuarios y roles, gestión de áreas y grupos, asignación de horarios, consulta de calendario y configuración del sistema.

Durante este proceso se brindaron orientaciones prácticas para el uso del sistema, resolviendo dudas y aclarando el alcance de las funcionalidades implementadas.

5.4.3 Pruebas de aceptación del sistema

Las pruebas de aceptación tuvieron como objetivo validar que el sistema SIGHA cumpliera con los requerimientos funcionales y operativos definidos, desde la perspectiva de los usuarios finales. Estas pruebas se realizaron mediante la ejecución de escenarios representativos del proceso de gestión de horarios académicos.

Los resultados obtenidos durante estas pruebas permitieron identificar observaciones y ajustes menores, los cuales fueron atendidos para mejorar la experiencia de uso y asegurar el correcto funcionamiento del sistema en el contexto institucional. Los resultados tabulados de las pruebas de aceptación se presentan en el capítulo de Resultados (véase ***Tabla 28***).

5.4.4 Documentación del sistema

Como parte de la fase de transición se elaboró la documentación necesaria para apoyar el uso y la administración del sistema SIGHA. Esta documentación tiene como objetivo facilitar la comprensión del sistema y servir como referencia para los usuarios finales y responsables de su operación.

Entre los documentos generados se encuentran:

- **Videotutoriales:** 7 videotutoriales organizados por rol de usuario que cubren todos los módulos del sistema.
- **Guía de administración:** Documentación técnica sobre la configuración, despliegue y mantenimiento del sistema.
- **Documentación del código fuente:** Estructura del proyecto, convenciones de código y guía de contribución.

5.4.5 Entregables de la fase de transición

Como resultado de la fase de transición se generaron diversos entregables que evidencian la puesta en marcha del sistema SIGHA y su validación en el entorno real de la Escuela.

Tabla 30.

Entregables: fase de transición

<i>Entregable</i>	<i>Descripción</i>
<i>Pruebas de usuario final</i>	<i>Resultados de pruebas de aceptación ejecutadas con usuarios reales..</i>
<i>Manual de usuario (videotutoriales)</i>	<i>Video Tutoriales que cubren todos los módulos del sistema, organizados por rol.</i>

<i>Documentación de despliegue</i>	<i>Descripción del entorno de producción y configuración del sistema.</i>
<i>Análisis de costos y beneficios</i>	<i>Comparativa de costos reales, análisis de reducción de tiempos y beneficios de la implementación.</i>

La fase de transición permitió poner el sistema SIGHA en funcionamiento dentro del entorno real de la Escuela de Ingeniería de Sistemas e Informática, validando que el sistema implementado cumpliera con los requerimientos institucionales.

5.4.6 Despliegue del sistema

El sistema SIGHA fue desplegado en un entorno de producción utilizando Docker y Docker Compose como plataforma de contenerización, lo que permite aislar cada componente en su propio contenedor, garantizar la reproducibilidad del entorno y simplificar tanto la puesta en marcha como las actualizaciones posteriores.

El frontend, desarrollado como una aplicación de página única (SPA) con Vue.js, se compila en archivos estáticos que son servidos a través de Nginx dentro de un contenedor Docker. La configuración de Nginx se incluye en el propio repositorio del frontend, lo que

asegura que el comportamiento del servidor web sea consistente entre los entornos de desarrollo y producción.

El backend, construido con Spring Boot, se empaqueta mediante un Dockerfile multi-stage. En la primera etapa se compila el proyecto con Maven y Java 17, y en la segunda etapa se copia únicamente el archivo JAR resultante sobre una imagen ligera que contiene solo el entorno de ejecución de Java (JRE). Este enfoque reduce el tamaño de la imagen final a aproximadamente 180 MB, frente a los 450 MB que genera un empaquetado convencional, y mejora la seguridad al ejecutar la aplicación con un usuario sin privilegios de administrador e incluir verificaciones de salud (health checks) integradas.

La base de datos PostgreSQL se ejecuta en su propio contenedor, con volúmenes persistentes que garantizan la conservación de los datos ante reinicios o actualizaciones del servicio. La base de datos contiene un schema compartido denominado core, donde residen las entidades comunes a todos los programas académicos (usuarios, roles, áreas y semestres), y schemas independientes por programa para los datos específicos de gestión horaria.

La orquestación de todos los contenedores se gestiona mediante Docker Compose. Para el entorno de producción se diseñó una estructura modular de despliegue, en la que cada componente (base de datos y cada instancia del backend por programa académico) cuenta con su propio archivo Docker Compose y sus variables de entorno independientes. Esto permite desplegar, detener o actualizar cada servicio de forma individual sin afectar al resto del sistema.

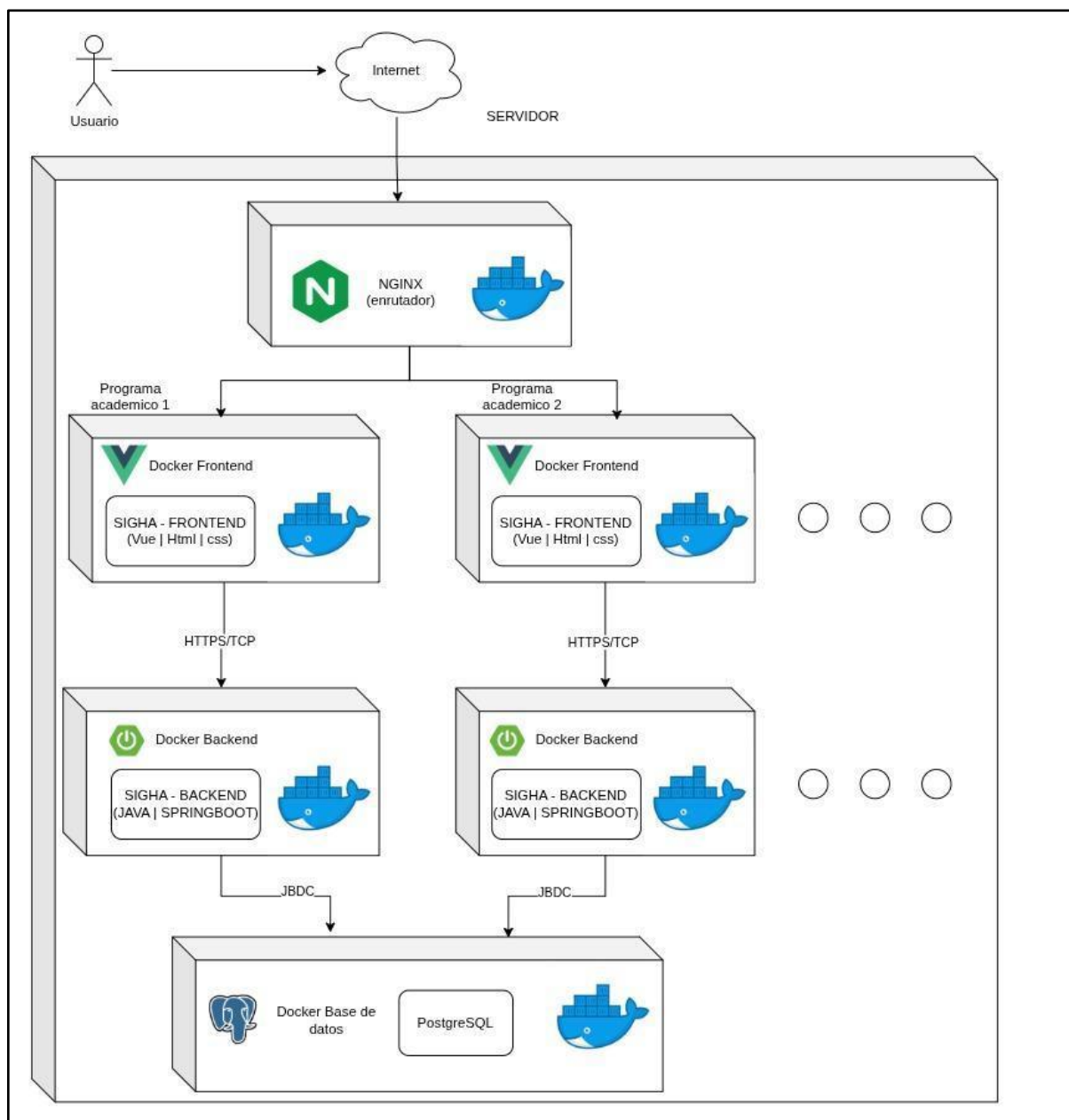
La comunicación entre los contenedores se realiza a través de una red Docker interna, y únicamente los puertos necesarios se exponen al exterior.

El proceso de despliegue se automatizó mediante scripts de construcción y puesta en marcha. El script de construcción compila el código fuente, genera la imagen Docker del backend y le asigna las etiquetas de versión correspondientes. El script de despliegue valida la existencia de los archivos de configuración, crea la red Docker si no existe, levanta los contenedores en el orden adecuado y muestra el estado de los servicios. Todo el despliegue puede ejecutarse con un único comando, lo que reduce el tiempo de puesta en marcha a pocos minutos y minimiza la posibilidad de errores manuales.

La comunicación entre el cliente y el servidor se realiza mediante el protocolo HTTPS, y el acceso al sistema se habilita a través de una URL institucional proporcionada por la Escuela.

Figura 33.

Arquitectura de implantación



5.4.7 Capacitación de usuarios

La capacitación de los usuarios finales constituyó un aspecto fundamental de la fase de transición. Las actividades de capacitación estuvieron orientadas a explicar el propósito del sistema, su funcionamiento general y las principales acciones que pueden realizar los diferentes actores.

Se elaboraron videotutoriales del uso del sistema, estos videos se encuentran en el apéndice B de los Apéndices, cubriendo los siguientes temas:

Tabla 31.

Contenido de manual de usuario (videotutoriales)

<i>Videotutorial</i>	<i>Tema</i>	<i>Duración</i>	<i>Dirigido a</i>
<i>Tutorial 1</i>	<i>Inicio de sesión y navegación general</i>	<i>0.6 min</i>	<i>Todos los usuarios</i>
<i>Tutorial 2</i>	<i>Registro de disponibilidad Horaria</i>	<i>0.6 min</i>	<i>Profesores</i>
<i>Tutorial 3</i>	<i>Gestión de usuario y Roles</i>	<i>0.6 min</i>	<i>Director de escuela / Coordinador Académico</i>
<i>Tutorial 4</i>	<i>Gestión de áreas, asignaturas y grupos</i>	<i>2 min</i>	<i>Director de escuela / Coordinador Académico</i>

Tutorial 5	<i>Asignación de horarios y validación de solapamientos</i>	<i>2 min</i>	<i>Director de escuela / Coordinador Académico</i>
Tutorial 6	<i>Consulta de calendario de escuela y dashboard</i> <i>Configuración del sistema y semestres</i>	<i>2 min</i>	<i>Director de escuela / Coordinador Académico</i>
Tutorial 7		<i>1 min</i>	<i>Director de escuela / Coordinador Académico</i>

Adicionalmente, se realizaron sesiones prácticas presenciales con el personal de la Escuela, en las cuales se resolvieron dudas y se aclaró el alcance de las funcionalidades implementadas

5.4.8 Pruebas de aceptación del sistema

Las pruebas de aceptación tuvieron como objetivo validar que el sistema SIGHA cumpliera con los requerimientos funcionales y operativos definidos, desde la perspectiva de los usuarios finales. Estas pruebas se realizaron mediante la ejecución de escenarios representativos del proceso de gestión de horarios académicos.

Tabla 32.*Pruebas de aceptación con usuarios finales*

ID	Escenario	Actor	Resultado Esperado	Resultado Obtenido	Estado
PA01	<i>El profesor accede al sistema y registra su disponibilidad horaria para el semestre activo</i>	<i>Profesor</i>	<i>El sistema permite seleccionar franjas y enviar la disponibilidad</i>	<i>El profesor registró correctamente su disponibilidad</i>	<i>Aprobado</i>
PA02	<i>El coordinador visualiza las disponibilidades registradas y aprueba una disponibilidad</i>	<i>Coordinador Académico</i>	<i>Se muestra la lista de disponibilidades con opción de aprobar/rechazar</i>	<i>Las disponibilidades se visualizaron y se aprobaron correctamente</i>	<i>Aprobado</i>
PA03	<i>El coordinador crea un área de conocimiento y asocia una asignatura</i>	<i>Coordinador Académico</i>	<i>El sistema permite crear el área y la asignatura asociada</i>	<i>El área y la asignatura se crearon correctamente</i>	<i>Aprobado</i>
PA04	<i>El coordinador crea un grupo para una materia y asigna un docente</i>	<i>Coordinador Académico</i>	<i>El sistema permite crear el grupo y asignar al docente</i>	<i>El grupo se creó y el docente fue asignado</i>	<i>Aprobado</i>
PA05					

PA06	<i>El coordinador asigna una franja horaria a un grupo sin conflictos</i>	<i>Coordinador Académico</i>	<i>El sistema acepta la asignación y la muestra en el calendario</i>	<i>La franja se asignó correctamente</i>	<i>Aprobado</i>
PA07	<i>El coordinador intenta asignar una franja que genera solapamiento de docente</i>	<i>Coordinador Académico</i>	<i>El sistema detecta el conflicto y rechaza la asignación con mensaje explicativo</i>	<i>El solapamiento fue detectado y se mostró mensaje de error</i>	<i>Aprobado</i>
PA08	<i>El director consulta el calendario</i>	<i>Director de escuela</i>	<i>Se muestra el calendario consolidado con todos los horarios del semestre</i>	<i>El calendario se visualizó correctamente</i>	<i>Aprobado</i>
PA09	<i>El director accede al dashboard con reportes del semestre</i>	<i>Director de escuela</i>	<i>Se muestran métricas consolidadas (grupos creados, disponibilidades, solapamientos evitados)</i>	<i>Los reportes se mostraron correctamente</i>	<i>Aprobado</i>
PA10	<i>El profesor consulta su horario asignado para el semestre</i>	<i>Profesor</i>	<i>Se muestra el calendario con las franjas asignadas al profesor</i>	<i>El horario del profesor se visualizó correctamente</i>	<i>Aprobado</i>
	<i>El coordinador crea un nuevo semestre académico y lo activa</i>	<i>Coordinador Académico</i>	<i>El sistema permite crear el semestre y activarlo como período actual</i>	<i>El semestre se creó y activó correctamente</i>	<i>Aprobado</i>

Los resultados obtenidos durante estas pruebas confirmaron que SIGHA cumple con los requerimientos definidos y que los usuarios finales pudieron operar el sistema de manera autónoma tras la capacitación proporcionada.

5.4.9 Análisis de costos y beneficios del sistema

A continuación se presenta el análisis de los costos reales del proyecto, la comparativa de tiempos entre el proceso manual y SIGHA, y los beneficios identificados a partir de la implantación del sistema.

5.4.9.1 Costos reales del proyecto. Los costos del proyecto se distribuyeron en recursos humanos, recursos tecnológicos y gastos adicionales. Cabe destacar que el alojamiento del sistema no representó un costo para el proyecto, ya que la Escuela de Ingeniería de Sistemas e Informática proporcionó la infraestructura de servidor necesaria para el despliegue en producción. La Tabla 32 detalla los costos reales del proyecto.

Tabla 33.*Costos del proyecto*

<i>Concepto</i>	<i>Responsable</i>	<i>Unidad</i>	<i>Cantidad</i>	<i>Valor unitario</i>	<i>Valor total</i>
Recursos humanos					
<i>Director del proyecto</i>	<i>Director</i>	<i>Hora</i>	<i>34</i>	<i>\$ 44.716/hora</i>	<i>\$ 1.520.345</i>
<i>Autor 1</i>	<i>Estudiante</i>	<i>Hora</i>	<i>170</i>	<i>\$ 16.000/hora</i>	<i>\$ 2.720.000</i>
<i>Autor 2</i>	<i>Estudiante</i>	<i>Hora</i>	<i>170</i>	<i>\$ 16.000/hora</i>	<i>\$ 2.720.000</i>
<i>Sub total</i>					<i>\$ 6.960.345</i>
Recursos tecnológicos					
<i>Alojamiento en la nube</i>	<i>Estudiante</i>	<i>Mes</i>	<i>4</i>	<i>Provisto por EISI</i>	<i>\$ 0</i>
<i>Internet</i>	<i>Estudiante</i>	<i>Mes</i>	<i>4</i>	<i>\$ 70.000/mes</i>	<i>\$ 280.000</i>
<i>Computador personal</i>	<i>Estudiante</i>	<i>Unidad</i>	<i>2</i>	<i>\$ 3.500.000</i>	<i>\$ 7.000.000</i>
<i>Sub total</i>					<i>\$ 7.280.000</i>
Gastos adicionales					
<i>Insumos</i>	<i>Estudiante</i>	<i>Imprevistos</i>			<i>\$130.000</i>
Presupuesto general					
<i>Concepto</i>	<i>Valor</i>				
<i>Recursos humanos</i>	<i>\$ 6.960.345</i>				
<i>Recursos tecnológicos</i>	<i>\$ 7.280.000</i>				
<i>Gastos adicionales</i>	<i>\$130.000</i>				
<i>Total</i>	<i>\$ 14.370.343</i>				

5.4.9.2 Comparativa de tiempos. A partir de la observación del proceso manual documentado en la fase de inicio y de las capacidades implementadas en SIGHA, se elaboró

la siguiente comparativa de tiempos estimados. La **Tabla 33** muestra la reducción esperada en cada actividad del proceso de gestión de horarios.

Tabla 34.

Comparativa de tiempos – Proceso manual vs. SIGHA

<i>Actividad</i>	<i>Tiempo Estimado (Manual)</i>	<i>Tiempo Estimado (SIGHA)</i>	<i>Reducción</i>
<i>Recolección de disponibilidad docente (contacto individual, espera de respuestas, consolidación)</i>	<i>5-7 días</i>	<i>1-2 días</i>	<i>~70%</i>
<i>Consolidación de disponibilidades en hoja de cálculo</i>	<i>2-3 días</i>	<i>Automático (tiempo real)</i>	<i>~95%</i>
<i>Asignación de horarios con validación manual de solapamientos</i>	<i>3-5 días</i>	<i>1-2 días</i>	<i>~65%</i>
<i>Verificación de solapamientos (revisión visual celda por celda)</i>	<i>1-2 días</i>	<i>Automático (milisegundos)</i>	<i>~99%</i>
<i>Comunicación de ajustes a docentes</i>	<i>1-2 días</i>	<i>Inmediato (sistema)</i>	<i>~90%</i>
<i>Tiempo total estimado</i>	<i>12-19 días</i>	<i>2-4 días</i>	<i>~75%</i>

5.4.9.3 Beneficios de la implantación del sistema. Dado que el sistema fue puesto en marcha al final del proyecto, aún no se ha completado un ciclo completo de gestión de horarios con la plataforma, por lo que no se dispone de estadísticas de uso real que permitan

cuantificar el impacto con datos concretos. Sin embargo, a partir de las pruebas de aceptación y de la comparativa de tiempos presentada, se identifican los siguientes beneficios esperados:

- **Reducción de tiempos:** se estima una disminución de aproximadamente el 75 % en el tiempo total del proceso, al pasar de 12–19 días a 2–4 días.
- **Eliminación de tareas manuales:** la consolidación de disponibilidades y la verificación de solapamientos se automatizan por completo, suprimiendo actividades repetitivas y propensas a errores.
- **Centralización de la información:** toda la disponibilidad docente se registra en una única plataforma accesible desde cualquier navegador, reemplazando la comunicación dispersa por mensajería instantánea.
- **Autonomía docente:** los profesores pueden registrar y modificar su disponibilidad de forma directa, sin intermediación administrativa.
- **Trazabilidad y apoyo a la toma de decisiones:** el sistema ofrece un historial completo de cambios y un dashboard con información consolidada.
- **Escalabilidad:** la arquitectura permite absorber el crecimiento en número de asignaturas, grupos y docentes sin cambios estructurales.

Estos beneficios podrán ser verificados y medidos con mayor precisión una vez que el sistema sea utilizado durante al menos un ciclo semestral completo.

6. Conclusiones

A partir del desarrollo del Sistema de Información para la Gestión de Horarios Académicos (SIGHA) y de los resultados obtenidos en cada una de las fases del proyecto, se presentan las siguientes conclusiones, alineadas con los objetivos específicos planteados:

Los autores analizaron de manera integral el proceso actual de gestión de horarios académicos en la Escuela de Ingeniería de Sistemas e Informática, identificando formalmente las problemáticas asociadas al uso de herramientas no especializadas, la ausencia de validaciones automáticas y la alta carga operativa del personal administrativo. Como resultado de este análisis, se levantaron 10 requerimientos funcionales (RF01-RF10), Siete (07) requerimientos no funcionales (RNF01-RNF07) y 6 requerimientos complementarios (RC01-RC06) que definen de manera precisa el alcance del sistema. Adicionalmente, se elaboraron los diagramas de casos de uso, los diagramas BPMN del proceso actual y propuesto, y el diagrama de estados del ciclo de vida del horario, los cuales proporcionan una visión completa de la interacción entre los actores y el sistema.

Los autores definieron una arquitectura de tres capas (cliente-servidor) basada en Vue.js para el frontend, Spring Boot para el backend y PostgreSQL como base de datos relacional, la cual permitió implementar de manera eficiente las funcionalidades del sistema. A partir de su uso por parte de los docentes y el personal administrativo, se evidenció una mejora significativa en la gestión de horarios, especialmente en la reducción de errores y en el tiempo requerido

para la asignación y validación de información. Los usuarios destacaron la facilidad para registrar disponibilidades, la claridad en la visualización de horarios y la utilidad de las validaciones automáticas para evitar solapamientos. En este sentido, la experiencia obtenida durante la implementación permite concluir que el uso de una arquitectura adecuada, junto con herramientas que automatizan procesos críticos, impacta positivamente la eficiencia operativa y la satisfacción de los usuarios en entornos académicos.

Se implementaron los 10 módulos funcionales del sistema SIGHA, incluyendo el módulo de recolección de disponibilidad docente y el módulo de generación de horarios con validaciones automáticas de solapamiento. El sistema incorpora 8 tipos de validaciones automáticas que garantizan la coherencia de las asignaciones horarias en tiempo real. Las pruebas unitarias (25 casos, 100% aprobados) y de integración (5 flujos completos, 100% aprobados) verifican el correcto funcionamiento de los componentes implementados.

Se realizó el despliegue del sistema en un entorno de producción accesible para los usuarios de la Escuela, utilizando la infraestructura de servidor proporcionada por la propia Escuela de Ingeniería de Sistemas e Informática. Se capacitó al personal administrativo y docente mediante videotutoriales y sesiones prácticas, y se ejecutaron 10 pruebas de aceptación con usuarios finales, todas con resultado aprobatorio. El análisis de costos y beneficios permitió estimar una reducción de aproximadamente el 75% en el tiempo total de gestión de horarios académicos. Si bien el sistema aún no ha completado un ciclo semestral de uso real que permita confirmar estas cifras con estadísticas de operación, las pruebas de aceptación y la comparativa

con el proceso manual sustentan la viabilidad y el impacto positivo esperado del sistema desarrollado.

En conclusión, el proyecto SIGHA logró transformar un proceso administrativo manual, basado en hojas de cálculo y comunicación informal, en un flujo digital estructurado que centraliza la información, automatiza las validaciones críticas y mejora la experiencia de los actores involucrados. El sistema se encuentra operativo y constituye una herramienta de apoyo efectiva para la planificación académica de la Escuela de Ingeniería de Sistemas e Informática de la Universidad Industrial de Santander.

7. Recomendaciones

7.1 Ampliación del sistema a otras escuelas y sistemas institucionales

SIGHA fue diseñado con una arquitectura modular que permite su extensión a otros programas académicos de la Universidad Industrial de Santander. Se recomienda evaluar la viabilidad de integrar el sistema con las plataformas académicas centrales de la UIS para sincronizar información de docentes, asignaturas y espacios físicos.

7.2 Obtención de permisos para acceso a información en tiempo real

Gestionar los permisos institucionales necesarios para que el sistema pueda acceder a fuentes de datos actualizadas en tiempo real (como el listado oficial de docentes, la

disponibilidad de aulas y la oferta académica vigente), lo que mejoraría la precisión y automatización del proceso.

7.3 Carga automática y gestión inteligente de información de los estudiantes

Explorar la posibilidad de integrar información de matrícula estudiantil para que el sistema pueda considerar la demanda proyectada de cupos al momento de planificar los grupos y horarios, optimizando la distribución de recursos.

7.4 Sistema de recomendación de horarios y asignaturas con IA

La incorporación de algoritmos de inteligencia artificial o aprendizaje automático que, a partir de los datos históricos de disponibilidad, preferencias docentes y demanda estudiantil, generen sugerencias automáticas de asignación de horarios que minimicen conflictos y optimicen la distribución de recursos.

Referencias bibliográficas

Acuña Garzón, M. (2016). *Evaluación del Sistema de Información Académica (SIA) de la Universidad Nacional de Colombia.*

<https://repositorio.unal.edu.co/bitstream/handle/unal/59044/MargothAcuñaGarzón.2016.pdf?sequence=1&isAllowed=y>

Anaya Jaimes, Y. M. (2022). *Presupuesto: Unidad de Apoyo a Gestión de Proyectos, Vicerrectoría de Investigación y Extensión.* Universidad Industrial de Santander. <https://uis.edu.co/wp-content/uploads/2022/08/TARIFAS-INSTITUCIONALES-2022-2.pdf>

Arley, R. M. B., & Ernesto, H. R. F. (2025, 21 mayo). *Automatización Robótica de Procesos (RPA) para el seguimiento analítico de activos y generación de reportes de excedentes en clientes AGPE de la Electrificadora de Santander (ESSA).* <https://noesis.uis.edu.co/items/4c7feb59-9676-44e9-ae9e-627a6c671e73>

Briceño Santiago, W. J. (2024). *Los sistemas de información para la gestión académica y la toma de decisiones de la Universidad Nacional del Comahue. Período post pandemia.* <https://rdi.uncoma.edu.ar/bitstream/handle/uncomaid/18409/Tesina%20%20-%20WHEELER%20-%20BRICEÑO.pdf?sequence=1&isAllowed=y>

Cabrera, R. (2023, 21 de noviembre). *La importancia de la gestión de recursos educativos.* Rededuca.

Chiavenato, I. (2009). *Introducción a la teoría general de la administración.* McGraw-Hill.

Glassdoor. (2024). *Sueldo promedio de desarrolladores web en Bogotá, Colombia.* https://www.glassdoor.com.mx/Sueldos/bogot%C3%A1-desarrollador-web-sueldo-SRCH_IL.0,6_IM1064_KO7,24.htm

IBM. (2025). *Rational Unified Process (RUP): Guía de aprendizaje y directrices del proceso.* IBM Documentation. <https://www.ibm.com/docs/es/>

Laudon Sistemas De Información Gerencial 14 Edición : Free Download, Borrow, and Streaming : Internet Archive. (2020, 2 mayo). Internet Archive. <https://archive.org/details/laudon-sistemas-de-informacion-gerencial-14-edicion>

Laudon, K. C., & Laudon, J. P. (2021). *Management Information Systems: Managing the Digital Firm*. Pearson.

O'Brien, J. A., & Marakas, G. M. (2021). *Introduction to Information Systems*. McGraw-Hill.

Ortega, L. (2022, 22 octubre). *Metodología RUP: ¿Qué es, cuál es su objetivo y cómo se utiliza?* Lean Management Blog. <https://lean-management.site/rup/>

Renne, P. S. F. M. (2023). *Sistema web basado en Scrum y XP para el proceso de gestión de docentes en el Instituto Gilda Ballivián Rosado*. https://alicia.concytec.gob.pe/vufind/Record/AUTO_ad3e36f41f4f6fdefee68edaf8d10f75

Wilver, Q. H. (2021). *Sistema de información web para mejorar la gestión de procesos de adquisiciones y contrataciones del Programa de Apoyo al Desarrollo Rural Andino – PRADERA* 2018. https://alicia.concytec.gob.pe/vufind/Record/RNAP_d4dc961c3859778a04fdb09e56927411

Apéndices**Apéndice A.** Requerimientos Funcionales de SIGHA.*Apéndices de Requerimientos funcionales del sistema*

<i>No.</i>	<i>Apéndices de Requerimientos Funcionales</i>
1	<i>SIGHA-Aplicación-base-y-Menú-Principal.pdf</i>
2	<i>SIGHA-Gestion-profesores-y-roles-permisos.pdf</i>
3	<i>SIGHA-Gestión-disponibilidades-profesores.pdf</i>
4	<i>SIGHA-Gestión-Áreas-y-Asignaturas.pdf</i>
5	<i>SIGA-Gestión-disponibilidades-administrativos.pdf</i>
6	<i>SIGHA-Gestión-Grupos.pdf</i>
7	<i>SIGHA-Gestion-Calendario-de-grupos.pdf</i>
8	<i>SIGHA-Consulta-Calendario-escuela.pdf</i>
9	<i>SIGHA-Configuración-y-parametrización-sistema.pdf</i>
10	<i>SIGHA-Configuración-usuario.pdf</i>

Los apéndices están disponibles en el Repositorio Institucional.

Este Apéndice complementa la información presentada en la Sección 5.2.5.2, donde se muestra un ejemplo representativo de la estructura de las historias de usuario.

Apéndice B. Video tutoriales por tema particular de la tabla 30.

- Juan Morantes, Daniel La Rotta. (2026). Inicio de sesión y navegación general. https://www.youtube.com/watch?v=6_sUfYgBEdo
- Juan Morantes, Daniel La Rotta. (2026). Registro de disponibilidad horaria. <https://www.youtube.com/watch?v=x2JUfapsEbU>
- Juan Morantes, Daniel La Rotta. (2026). Gestión de áreas, asignaturas y grupos <https://www.youtube.com/watch?v=Yk3U5CMpnx0>
- Juan Morantes, Daniel La Rotta. (2026). Asignación de horarios y validación de solapamientos <https://www.youtube.com/watch?v=YJHMOqfw67U>
- Juan Morantes, Daniel La Rotta. (2026). Configuración del sistema y semestres <https://www.youtube.com/watch?v=DXFGtq3s11I>
- Juan Morantes, Daniel La Rotta. (2026). Gestión de usuario y Roles https://www.youtube.com/watch?v=UkFrta_XGFw