

# FACULTAD DE INGENIERÍA

Carrera de Ingeniería de Sistemas Computacionales

**“Desarrollo de sistema de administración de docentes  
para una institución académica”**

**Trabajo de suficiencia profesional para optar al título**

**profesional de:**

**Ingeniero de sistemas computacionales**

**Autor:**

Christopher Javier Huaman Guevara

**Asesor:**

Dra. Ing. Patricia Janet Uceda Martos

<https://orcid.org/0000-0003-1771-9970>

Lima - Perú

2025

## Informe de Similitud



Página 2 de 60 - Descripción general de integridad

Identificador de la entrega: trnoid...1342090395




### 6% Similitud general

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para ca...

#### Filtrado desde el informe

- Bibliografía
- Texto citado
- Texto mencionado
- Coincidencias menores (menos de 8 palabras)

#### Fuentes principales

- 6%  Fuentes de Internet
- 1%  Publicaciones
- 2%  Trabajos entregados (trabajos del estudiante)

#### Marcas de integridad

##### N.º de alertas de integridad para revisión

No se han detectado manipulaciones de texto sospechosas.

Los algoritmos de nuestro sistema analizan un documento en profundidad para buscar inconsistencias que permitirían distinguirlo de una entrega normal. Si advertimos algo extraño, lo marcamos como una alerta para que pueda revisarlo.

Una marca de alerta no es necesariamente un indicador de problemas. Sin embargo, recomendamos que preste atención y la revise.

## **Dedicatoria**

A todos aquellos que creyeron en mí y me motivaron para alcanzar esta meta. Esto va dedicado para todos ustedes.

## **Agradecimiento**

A mis padres, por su amor, apoyo incondicional y por enseñarme a nunca rendirme.

A mi hermana, por su cariño y por estar siempre a mi lado.

A mis amigos, por acompañarme con su amistad y motivación en cada etapa de este camino.

Gracias a todos por ser parte de este logro.

## Tabla de contenido

Índice de tablas.....	6
Índice de Figuras .....	7
RESUMEN EJECUTIVO .....	9
CAPÍTULO I. INTRODUCCIÓN .....	10
CAPÍTULO II. MARCO TEÓRICO .....	15
CAPÍTULO III. DESCRIPCIÓN DE LA EXPERIENCIA.....	18
CAPÍTULO IV. RESULTADOS.....	52
CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES .....	57
REFERENCIAS.....	61

## Índice de tablas

<b>Tabla 1</b> Roles de miembros del proyecto .....	21
<b>Tabla 2</b> Tiempo estimados en la administración de docentes .....	52
<b>Tabla 3</b> Comparativa de Tecnologías implementadas.....	53
<b>Tabla 4</b> Métricas de consultas y rendimiento.....	53
<b>Tabla 5</b> Métricas de Procesamiento de Planillas .....	54
<b>Tabla 6</b> Gestión de datos maestros.....	55
<b>Tabla 7</b> Eficiencia en generación de reportes.....	55
<b>Tabla 8</b> Métricas de automatización Zoom .....	56
<b>Tabla 9</b> Métricas de automatización Zoom .....	56

## Índice de Figuras

<b>Figura 1</b> Organigrama del área de sistemas .....	13
<b>Figura 2</b> Organigrama de las sedes .....	14
<b>Figura 3</b> <i>Sistema Educativo</i> .....	18
<b>Figura 4</b> Interfaz de Modulo de clase PeopleSoft .....	19
<b>Figura 5</b> Diagrama de Ishikawa .....	20
<b>Figura 6</b> Diagrama de GANTT .....	24
<b>Figura 7</b> Arquitectura de Software.....	25
<b>Figura 8</b> Conexión de base de datos y prueba de estrés.....	27
<b>Figura 9</b> Grupo de Hilos .....	28
<b>Figura 10</b> Petición HTTP .....	29
<b>Figura 11</b> Reporte Resumen.....	29
<b>Figura 12</b> Trata de data de Reporte Resumen .....	30
<b>Figura 13</b> Estructura MVC del Sistema .....	31
<b>Figura 14</b> Módulos principales del sistema .....	32
<b>Figura 15</b> Módulo de registros .....	33
<b>Figura 16</b> Interfaz de Clases .....	33
<b>Figura 17</b> Vista previa de clases .....	34
<b>Figura 18</b> Opciones de clases.....	35

<b>Figura 19</b> Interfaz de Reuniones .....	36
<b>Figura 20</b> Interfaz de Eventualidades.....	36
<b>Figura 21</b> Interfaz de Ingreso .....	37
<b>Figura 22</b> Interfaz de Dividir .....	38
<b>Figura 23</b> Interfaz de Zoom .....	39
<b>Figura 24</b> Módulo de Mantenimiento .....	39
<b>Figura 25</b> Interfaz de Modelo Horario .....	40
<b>Figura 26</b> Interfaz de Tipo de eventualidad .....	40
<b>Figura 27</b> Interfaz de Docentes .....	41
<b>Figura 28</b> Interfaz de Usuarios.....	41
<b>Figura 29</b> Interfaz de Usuario .....	42
<b>Figura 30</b> Interfaz de Ubicaciones .....	42
<b>Figura 31</b> Menú de reportes .....	43
<b>Figura 32</b> JasperSoft Studio .....	44
<b>Figura 33</b> Interfaz de Horas para planilla.....	48
<b>Figura 34</b> Job de Api Zoom .....	48
<b>Figura 35</b> Prueba de rendimiento del servidor.....	50
<b>Figura 36</b> Vista de contenedores.....	50

## RESUMEN EJECUTIVO

El problema central identificado consistía en procesos manuales ineficientes de asignación docente, errores frecuentes en cálculo de horas de planilla, ausencia de reportes automatizados y configuración manual de clases virtuales mediante Zoom.

La solución implementada utilizó arquitectura dockerizada con Laravel como framework principal, SQL Server optimizado, Redis para cache e integración con API de Zoom. Se desarrollaron seis módulos específicos: asignación de docentes, seguimiento de matriculados, gestión de horas de planilla, mantenimiento de datos maestros, generación automatizada de reportes e integración Zoom.

Los resultados cuantificables incluyen: reducción en tiempos de asignación, eliminación total de conflictos de horario, disminución de errores en planillas, automatización reuniones Zoom mensuales, y reducción del tiempo de generación de reportes. El sistema alcanzó una alta disponibilidad, soportando grandes cantidades de usuarios simultáneos.

Se aplicaron competencias de análisis de sistemas, arquitectura de software, integración de API, optimización de bases de datos, gestión de proyectos y trabajo colaborativo, demostrando capacidad para diseñar soluciones escalables en entornos empresariales reales.

## CAPÍTULO I. INTRODUCCIÓN

Este trabajo de investigación se basa en mi experiencia laboral en una institución educativa reconocida en el Perú, especializada en la enseñanza del idioma inglés y con una destacada trayectoria cultural. Dicha institución cuenta con un entorno tecnológico complejo que requiere un constante desarrollo y mantenimiento de sus sistemas internos.

La organización brinda diversos servicios orientados a su comunidad estudiantil, entre los cuales se encuentran programas de enseñanza del idioma inglés, actividades culturales y servicios bibliotecarios.

Dentro de **la enseñanza del idioma inglés** se dan los servicios de diversos programas de enseñanza como cursos regulares para niños, adolescentes y adultos, incluyendo enseñanzas de inglés para empresas, programa de conversaciones y preparaciones para exámenes internacionales como IELTS y más.

En el ámbito **cultural**, la institución promueve la difusión del arte y la cultura mediante la organización de exposiciones, obras teatrales, presentaciones musicales, talleres artísticos y diversas actividades relacionadas.

Por otro lado, el **servicio de biblioteca** se especializa en la difusión de material físico y digital, ofreciendo acceso a libros en inglés y español, periódicos, revistas y otros recursos digitales de apoyo académico.

Actualmente trabajo como Analista de Software. Mi trabajo diario incluye varias actividades relacionadas con el desarrollo y mantenimiento de sistemas para todos los servicios de

la empresa.

En la empresa, el área de sistemas está organizada de manera jerárquica y funcional. En la parte más alta se encuentra la Gerencia General, que administra toda la institución.

Debajo de la gerencia está la Jefatura de Sistemas, que es quien supervisa todas las áreas tecnológicas de la institución. Esta jefatura coordina cuatro áreas principales: Desarrollo, Tecnología, Producción y Helpdesk.

Dentro del Área de Sistemas se identifican tres subdivisiones especializadas que desempeñan funciones críticas para la infraestructura tecnológica de la organización:

**Área de Producción**, esta unidad es responsable de la administración integral de los servidores de la organización. Entre sus funciones principales se encuentra la gestión de certificados digitales (TLS) y dominios para las aplicaciones web corporativas. Asimismo, se encarga del aprovisionamiento y configuración de servidores virtuales que facilitan las comunicaciones tanto entre servidores internos como con sistemas externos, garantizando la disponibilidad y seguridad de la infraestructura productiva.

**Área de Tecnología**, esta área tiene a su cargo la administración completa de la infraestructura de red empresarial. Su responsabilidad incluye la gestión de las conexiones de red de cada servidor mediante diversas herramientas especializadas de conectividad. Adicionalmente, realizan la verificación y monitoreo continuo de las conexiones de red, tanto en el ámbito interno como en las comunicaciones con redes externas, asegurando la integridad y rendimiento de la infraestructura de telecomunicaciones.

**Área de Desarrollo**, esta unidad gestiona todos los sistemas a nivel de desarrollo, siendo

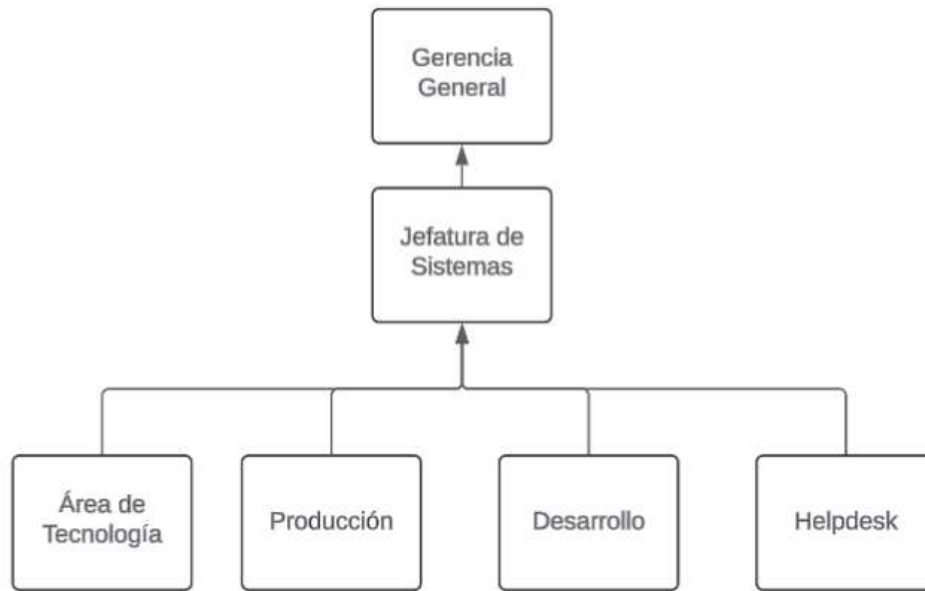
responsable tanto de la implementación de mejoras en sistemas existentes como de la creación de nuevas soluciones tecnológicas. El área emplea diversas tecnologías y frameworks para el desarrollo de sistemas de información, seleccionando las herramientas más apropiadas según los requerimientos específicos solicitados por las diferentes áreas de la organización.

**Área de Helpdesk (Mesa de Ayuda)**, esta área se encarga de gestionar todas las peticiones y requerimientos generados por los usuarios finales de la organización. La administración de estas solicitudes se clasifica en tres categorías principales:

**Requerimientos:** Solicitudes en las que se requiere información, datos o recursos específicos relacionados con las necesidades operativas del usuario.

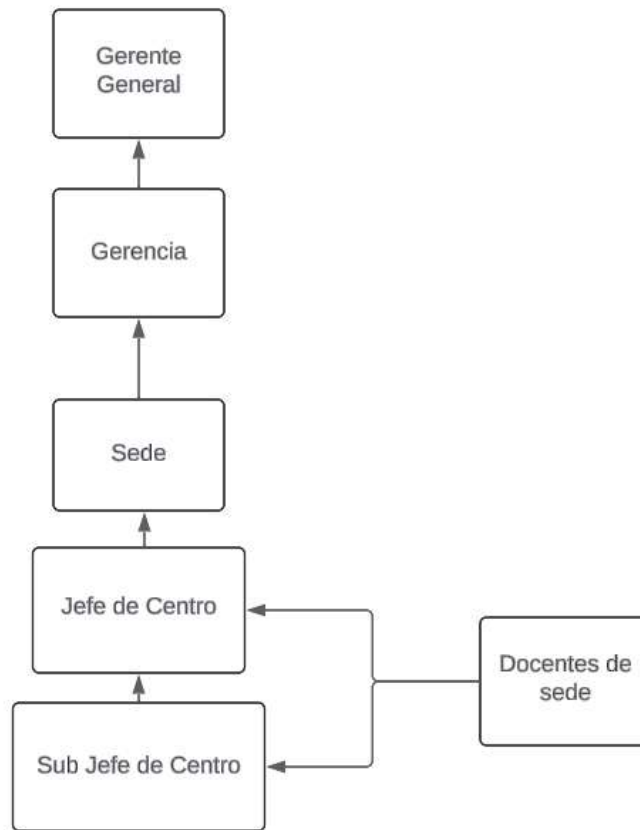
**Incidencias:** Reportes que hacen referencia a problemas inesperados, fallas en el sistema o controles que requieren atención inmediata para restablecer la operatividad normal de los servicios.

**Proyectos:** Peticiones que demandan trabajos de desarrollo más complejos, los cuales requieren un período de tiempo considerable para su análisis, implementación y resolución.

**Figura 1***Organigrama del área de sistemas*

Nota: Elaboración Propia

Por otro lado, dentro del área de educación en el ámbito de la empresa en la enseñanza de Inglés tenemos a la sede quien dentro de esta se encuentran liderando los jefes de centro y subjeses de centro quienes dirigen a los docentes administrando su supervisión de estos a las clases correspondientes, por encima de las sedes se encuentra el área de operaciones, que comanda y da permisos administrativos para cada sede, posteriormente la gerencia y el gerente general administran todo.

**Figura 2***Organigrama de las sedes*

Nota: Elaboración Propia

Dentro de la empresa existen diversas sedes como Arequipa, Comas, Camacho, Lima, Los Jardines, Miraflores, Pueblo Libre, San Borja, San Isidro, San Juan de Lurigancho, San Miguel, Santa Anita, Surco y Trujillo, siendo estas las sedes de enseñanza.

## CAPÍTULO II. MARCO TEÓRICO

Para el desarrollo del proyecto de suficiencia profesional que estuvo enmarcado en el desarrollo de un sistema, se tuvieron en cuenta los conceptos vinculados con:

Las **metodologías de desarrollo de software** hacen referencia a los procedimientos racionales utilizados donde para alcanzar el objetivo se requiere conocimientos y habilidades específicas (Maida & Pacienza, 2015). Estas metodologías establecen procesos, técnicas y herramientas que guían el desarrollo de software de manera organizada, garantizando la calidad del producto final y optimizando los recursos disponibles.

Respecto a **lenguajes de programación**, se utilizaron:

**Java** como lenguaje de programación principal. La elección de Java se fundamentó en que demuestra escalabilidad horizontal y vertical superior con soporte nativo para multithreading y estructuras thread-safe (Java The Complete Reference Ninth Edition, 2014). Esta característica resulta esencial para sistemas que requieren el manejo simultáneo de múltiples usuarios y operaciones concurrentes, garantizando un rendimiento óptimo y estabilidad del sistema.

Asimismo, se utilizó **Play Framework**, que representa una aproximación de desarrollo web basada en cambios visibles inmediatamente durante el desarrollo, proporcionando hot reloading que mejora significativamente la productividad del desarrollador (What is Play? , 2025). Siendo esta una aplicación monolítica, Play Framework se emplea tanto para backend como frontend, permitiendo un desarrollo integrado y cohesivo de todas las funcionalidades del sistema.

Para el testing siendo estas pruebas de software, se realizó a elección del software a utilizar

donde se realizaron pruebas automatizadas a nivel de script, estas pruebas son beneficiosas, pues garantizan un menor margen de error humanos donde con las pruebas de software otorga una mejor eficiencia a través del tiempo (Susnjara & Smalley, 2025), por ello las realizaciones de las pruebas automatizadas son indispensables para verificar el desarrollo y estabilidad correcto del software.

Adicionalmente, el sistema se apoya con otro sistema desarrollado en **Laravel** que funciona como API, donde realiza cálculos de forma conjunta con el sistema de administración de docentes, siendo Laravel una tecnología que permite desarrollar programas escalables siendo este basado en el patrón MVC, teniendo todo modularizado (Aguirre, 2024). En conjunto, ambos sistemas realizan inserciones coordinadas en la base de datos SQL Server, asegurando la integridad y consistencia de la información.

Para la base de datos se utilizó **Microsoft SQL Server**, debido a que es una base de datos madura y altamente consistente donde soporta procedimiento almacenados, una gran escalabilidad, estabilidad y seguridad (Hernández, 2006). Esta elección garantiza la robustez necesaria para el manejo de información crítica y sensible del personal docente, proporcionando características avanzadas de seguridad, respaldo y recuperación de datos.

Paralelamente se implementó **Redis** como base de datos en memoria para obtener información rápida de forma directa aumentando la productividad hasta 8 veces y 80% menos latencia que otras bases de datos (Nielsen, 2023). Esta tecnología permite mejorar significativamente los tiempos de respuesta del sistema al almacenar en memoria los datos de acceso frecuente, reduciendo la carga sobre la base de datos principal.

Todo el sistema fue **dockerizado** para una mejor gestión de recursos, implementando una red interna para todo el sistema. Esta contenerización facilita el despliegue, la escalabilidad y el

mantenimiento del sistema, además de proporcionar un entorno consistente entre desarrollo y producción (Álvaro Iradier, 2021).

Respecto a las normas que rigen el tratamiento de datos en el Perú, se tomó en consideración:

**Ley de Protección de Datos Personales (Ley 29733)**, que entró en vigor desde el 3 de julio de 2011, que garantiza el derecho fundamental a la protección de datos personales mediante el tratamiento adecuado de la información (Congreso de la República, 2011). Esta ley es fundamental para el sistema desarrollado, ya que maneja información sensible del personal docente y debe cumplir con los principios de legalidad, consentimiento, finalidad y seguridad establecidos en la normativa.

Respecto a las **limitaciones**, en el desarrollo del proyecto se encontraron algunas vinculadas con:

El sistema de administración de docentes podría presentar problemas de **rendimiento** al no soportar adecuadamente un exceso de usuarios, ya que es compleja la manipulación simultánea con una gran cantidad de usuarios concurrentes. Esta limitación puede afectar la experiencia del usuario y la eficiencia operativa del sistema.

La generación de reportes internos representa un consumo excesivo de recursos debido a que realiza un procesamiento intensivo de datos, lo que puede impactar el rendimiento general y la disponibilidad para otros usuarios.

### CAPÍTULO III. DESCRIPCIÓN DE LA EXPERIENCIA

Ingresé a la empresa el 1° de diciembre del año 2022 como analista de sistemas junior, y posteriormente el 1° octubre del año 2025 pasé a desempeñar el rol de analista de sistemas.

Dentro del trabajo me dediqué a realizar análisis de los sistemas centrales de la empresa, donde se verificaba el sistema Core, siendo este el ERP educativo PeopleSoft.

#### Figura 3

*Sistema Educativo*



Nota: Obtenida del sistema educativo de la Empresa PeopleSoft

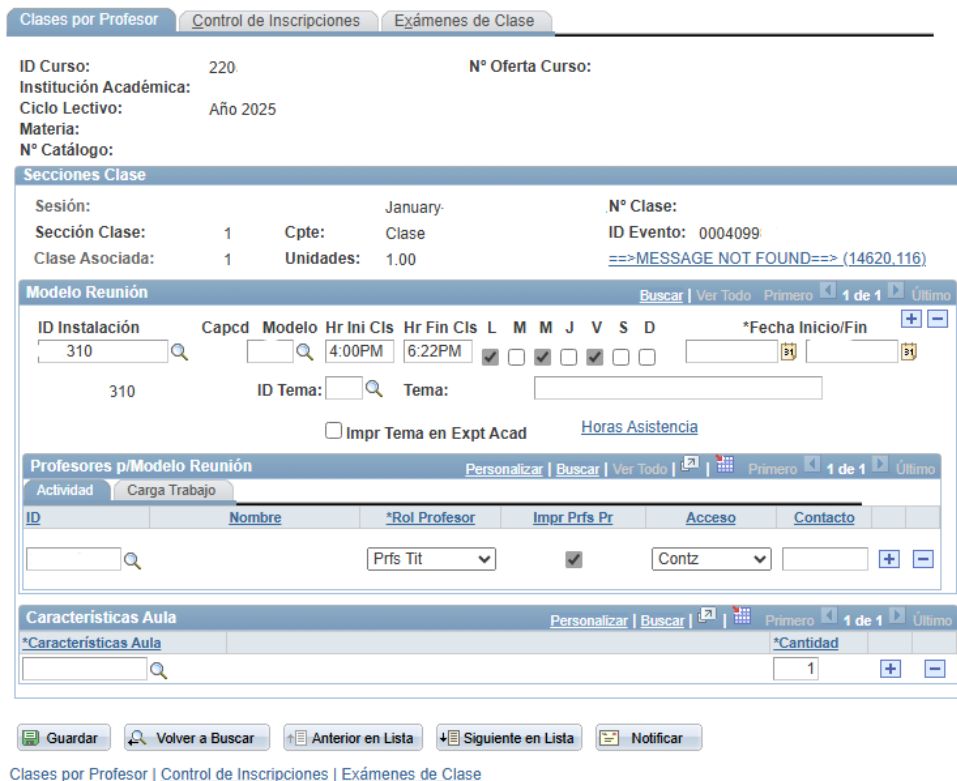
Dentro del sistema PeopleSoft se realiza la administración de todo el Core educativo tal como se muestra en la figura 3, donde se encuentran las creaciones de clase, la creación de los

docentes como usuario dentro de la empresa, las asignaciones de los docentes, la creación de cursos, creación de nuevas sedes, registro de matrículas, registros de notas de alumnado, registro de asistencias y más.

El sistema Core PeopleSoft es una versión del PeopleSoft 9, al ser una versión antigua incluye muchas características como un sistema de Oracle, su interfaz y su comportamiento no es tan amigable con el usuario que se encarga de la administración de las clases, así como la administración de los docentes para su asignación a las clases correspondientes.

**Figura 4**

Interfaz de Módulo de clase PeopleSoft



The screenshot displays the 'Clases por Profesor' (Classes by Professor) module interface. At the top, there are three tabs: 'Clases por Profesor', 'Control de Inscripciones', and 'Exámenes de Clase'. Below the tabs, the following information is displayed:

- ID Curso: 220
- Institución Académica: Año 2025
- Ciclo Lectivo: Año 2025
- Materia:
- N° Catálogo:
- N° Oferta Curso:

The 'Secciones Clase' (Class Sections) section includes:

- Sesión: January
- Sección Clase: 1
- Cpte: Clase
- Clase Asociada: 1
- Unidades: 1.00
- N° Clase:
- ID Evento: 0004099
- Message: ===MESSAGE NOT FOUND=== (14620.116)

The 'Modelo Reunión' (Meeting Model) section features a search bar and a table with columns: ID Instalación, Capcd, Modelo, Hr Ini Cls, Hr Fin Cls, L, M, M, J, V, S, D, and \*Fecha Inicio/Fin. The table contains one row with values: 310, [empty], [empty], 4:00PM, 6:22PM, [checked], [empty], [checked], [empty], [checked], [empty], [empty], [empty]. Below the table are fields for ID Tema, Tema, and a checkbox for 'Impr Tema en Expt Acad'. A link for 'Horas Asistencia' is also present.

The 'Profesores p/Modelo Reunión' (Professors by Meeting Model) section has a search bar and a table with columns: ID, Nombre, \*Rol Profesor, Impr Prfs Pr, Acceso, and Contacto. The table contains one row with values: [empty], [empty], Prfs Tit, [checked], [empty], [empty].

The 'Características Aula' (Classroom Characteristics) section has a search bar and a table with columns: \*Características Aula and \*Cantidad. The table contains one row with values: [empty], 1.

At the bottom of the interface, there are navigation buttons: 'Guardar', 'Volver a Buscar', 'Anterior en Lista', 'Siguiete en Lista', and 'Notificar'. Below these buttons, the current page is identified as 'Clases por Profesor | Control de Inscripciones | Exámenes de Clase'.

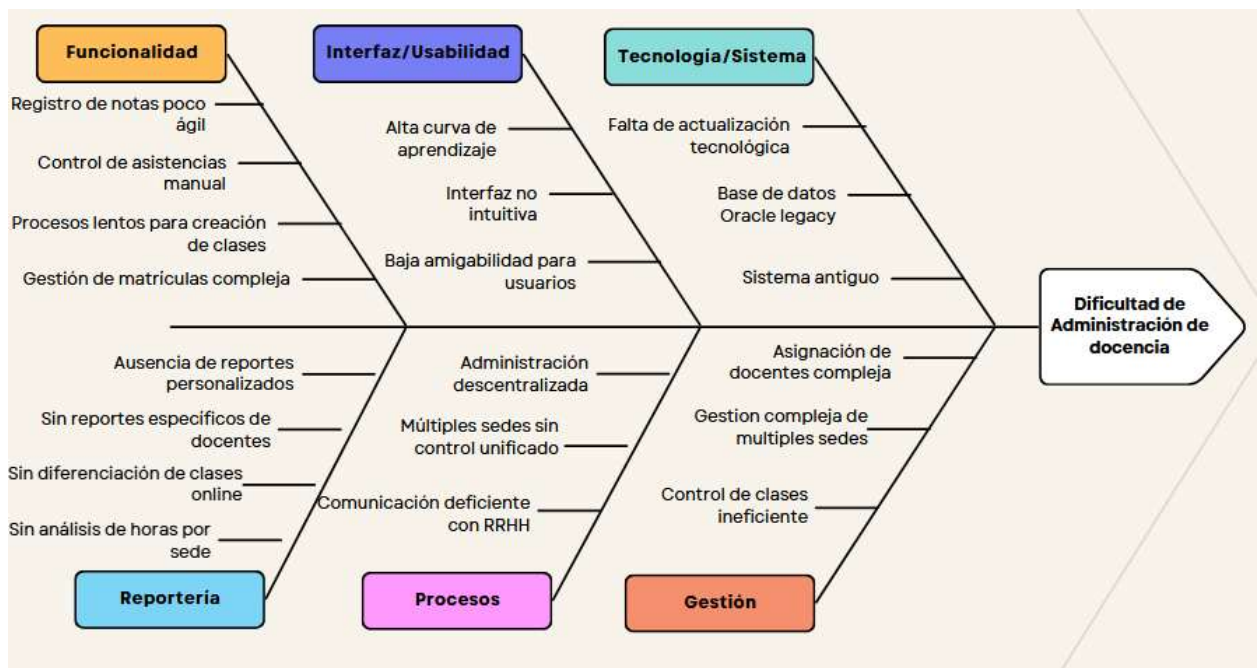
Nota: Imagen obtenida del sistema PeopleSoft

Como se podrá observar en la Figura 4, el principal problema de esta alternativa es la creación de un sistema para poder controlar todo lo enfocado a la administración en las clases y los docentes, donde el jefe y subjefe de sede al ser los encargados de administrar las clases, añadido a esto, se necesita una forma de tener reportes más exclusivos para el sistema , como reportes de docentes, reportes de clases, cantidad de horas por clase en sede, reporte de las clases administradas online y más.

Como se aprecia, existen diversos problemas para los cuales la centralización es clave, así como se visualiza en el siguiente diagrama de Ishikawa

**Figura 5**

*Diagrama de Ishikawa*



Nota: Elaboración Propia

Por ello se creó un sistema integrado en base a la administración del docente, donde internamente se mitigan todos los inconvenientes indicados, como las funcionalidades en registro, control procesos, así como la usabilidad dentro de esta, enfocándose en salvaguardar todos estos procesos y agilizar de manera óptima todas estas.

El desarrollo del sistema se realizó con apoyo de desarrolladores de software, siendo estos, José Bautista, Alexander Ramirez, Daniel Barrientos, Alvaro Barrera, Rodrigo Raymundo y Christopher Huaman, cada uno especializándose en varios apartados del proyecto en conjunto, los miembros mencionados tienen conocimiento de las bases y la lógica de negocio de la empresa, así como conocimiento de las tecnologías a utilizar para el desarrollo de esta.

Siendo sus roles principales y secundarios descritos en la siguiente tabla:

**Tabla 1**

*Roles de miembros del proyecto*

	Rol principal	Roles Secundarios
Christopher Huaman	Encargado de la administración de servidores y comunicaciones internas	Desarrollador de módulos dentro del sistema de Encargado de la conexión con el Api Zoom
Alexander Ramirez	Desarrollador de módulos dentro del sistema	Encargado de gestión del proyecto
Daniel Barrientos	Desarrollador de módulos dentro del sistema	

	Rol principal	Roles Secundarios
Jose Bautista	Encargado de la conexión con sistemas de Recursos Humanos	Desarrollador de módulos dentro del sistema
Rodrigo Raymundo	Desarrollador de módulos dentro del sistema	
Alvaro Barrera	Desarrollador de módulos dentro del sistema	

Dentro de las principales funciones que desarrollé, analicé los requerimientos que surgieron en las diferentes áreas de la institución para crear soluciones tecnológicas que mejoren los procesos académicos y administrativos. También participé en el desarrollo de estas soluciones, trabajando con diferentes tecnologías y herramientas de programación, como: Docker, lenguajes de programación de frontend y backend, administración de servidores Linux, software de testing.

Otra parte importante de mi trabajo fue brindar soporte en la implementación de sistemas de información educativos, lo que me ha dado una visión amplia de cómo funcionan estos procesos en una institución educativa. Además, colaboré en proyectos de mejora continua y participé en el mantenimiento de las plataformas tecnológicas que usa la institución.

También me encargué de documentar los procesos y sistemas, lo cual es fundamental para mantener un buen control de los desarrollos realizados.

Dentro de la elaboración del sistema unas de mis funcionalidades principales fue administración del proyecto general, verificando cambios y las nuevas integraciones en cada parte del proyecto, así como las creaciones o implementaciones dependiendo de la asignación a cargo

en el momento.

En base a la identificación del problema tal como se muestra en la figura 5, se definió el siguiente objetivo general:

- Desarrollar un sistema web integral de administración de docentes que optimice procesos de asignación completos con generación de reportes.

Dentro de los objetivos específicos tenemos los siguientes:

- Crear el sistema de asignación de docentes a las clases respectivas.
- Crear el módulo de seguimientos de matriculados, así como su ubicación y las fechas de inicio y fecha fin.
- Crear un apartado de horas de planillas para hacer la comunicación con el sistema de recursos humano.
- Crear un módulo de mantenimiento, donde se pueda establecer los docentes, las eventualidades, los usuarios, las ubicaciones.
- Crear un módulo de reportes, que minimice el trabajo manual dentro de la empresa.
- Crear automáticamente las reuniones de Zoom por cada clase.

Para la implementación del proyecto se utilizó la metodología en cascada o también llamada Waterfall, esta metodología se usó por el enfoque de gestión de proyectos secuencial y estructurado, donde cada fase se completa antes de iniciar la siguiente. Se distingue por una planificación detallada, documentación precisa y una ejecución ordenada desde el inicio hasta el final del proyecto (Equipo de Adobe para empresas, 2025).

Dentro de esta metodología fue necesario la elaboración de un cronograma de actividades con la duración, fechas y responsables para cada tarea, por ello se adoptó el diagrama de Gantt como herramienta de control y planificación tal como se visualiza en la figura 6.

**Figura 6**

*Diagrama de GANTT*

Modo de tarea	Nombre de tarea	Duración	Comienzo	Pr	% con	Nombres de los recursos	Texto
	<b>GANTT INTERFASES SPRING</b>	250 días?	vie 8/9/24		100%	Alexander Ramirez	
	<b>E01. Carga Proyecto</b>	22 días	mar 1/7/25		100%	Christopher Huaman, Jose Bautista	
	Conexion Base de datos MySQL y Hours	4 días	mar 1/7/25		100%	Christopher Huaman	
	Gestión del Servidor Productivo	7 días	jue 1/23/25		100%	Jose Bautista	
	Rendimiento con Jmeter	8 días	lun 1/27/25		100%	Christopher Huaman	
	<b>E02. Hours</b>	214 días?	mar 1/7/25		100%	Alexander Ramirez	
	<b>Clase</b>	200 días?	lun 1/20/25		100%	Alexander Ramirez	
	<b>Creacion</b>	169 días	lun 1/20/25		100%	Alexander Ramirez	
	<b>S1 - Se construirá el servicio para recibir múltiples clases sin docente</b>	169 días	lun 1/20/25		100%	Alexander Ramirez	
	S1.1 - Homologacion	20 días	lun 3/10/25		100%	Jose Bautista	
	S1.2 - Desarrollo	62 días	lun 1/20/25		100%	Jose Bautista	
	S1.3 - Endpoint	46 días	lun 1/20/25		100%	Jose Bautista	
	S1.3 - Pruebas	105.5 días	jue 4/24/25	11	100%	Alexander Ramirez,Rodrigo Raymundo	El 24/03 se det
	S1.4 - Actualización	86.2 días	mié 5/21/25		100%	Alexander Ramirez,Rodrigo Raymundo	Reproceso de
	<b>S2 - Se construirá un servicio para retornar el ok o error de las clases</b>	61 días	jue 1/23/25		100%	Jose Bautista	
	S2.1 - Homologación	18 días	lun 3/10/25		100%	Jose Bautista	
	S2.2 - Desarrollo	54 días	jue 1/23/25		100%	Jose Bautista,Jose Delgado	
	S2.3 - Endpoint	40 días	vie 1/31/25		100%	Jose Bautista	El 24/03 se det
	S2.3 - Pruebas	9 días	mar 4/15/25	18	100%	Jose Bautista	
	<b>S3 - Partición de Clase</b>	169 días	lun 1/20/25		100%	Christopher Huaman	
	S3.1 - Homologacion	20 días	lun 3/10/25		100%	Christopher Huaman	
	S3.2 - Desarrollo	61 días	lun 1/20/25		100%	Christopher Huaman	
	S3.3 - Endpoint	40 días	vie 1/31/25		100%	Christopher Huaman	El 24/03 se det
	S3.4 - Pruebas	114 días	mar 4/15/25	23	100%	Christopher Huaman,Alvaro Barrera	
	S3.5 - Actualización	88 días	mié 5/21/25		100%	Christopher Huaman,Alvaro Barrera	
	<b>S13 - Eliminación de Partición de Clase</b>	104 días	mar 4/29/25		100%	Christopher Huaman	

Nota: Obtenida del diagrama de Gantt en la realización del proyecto.

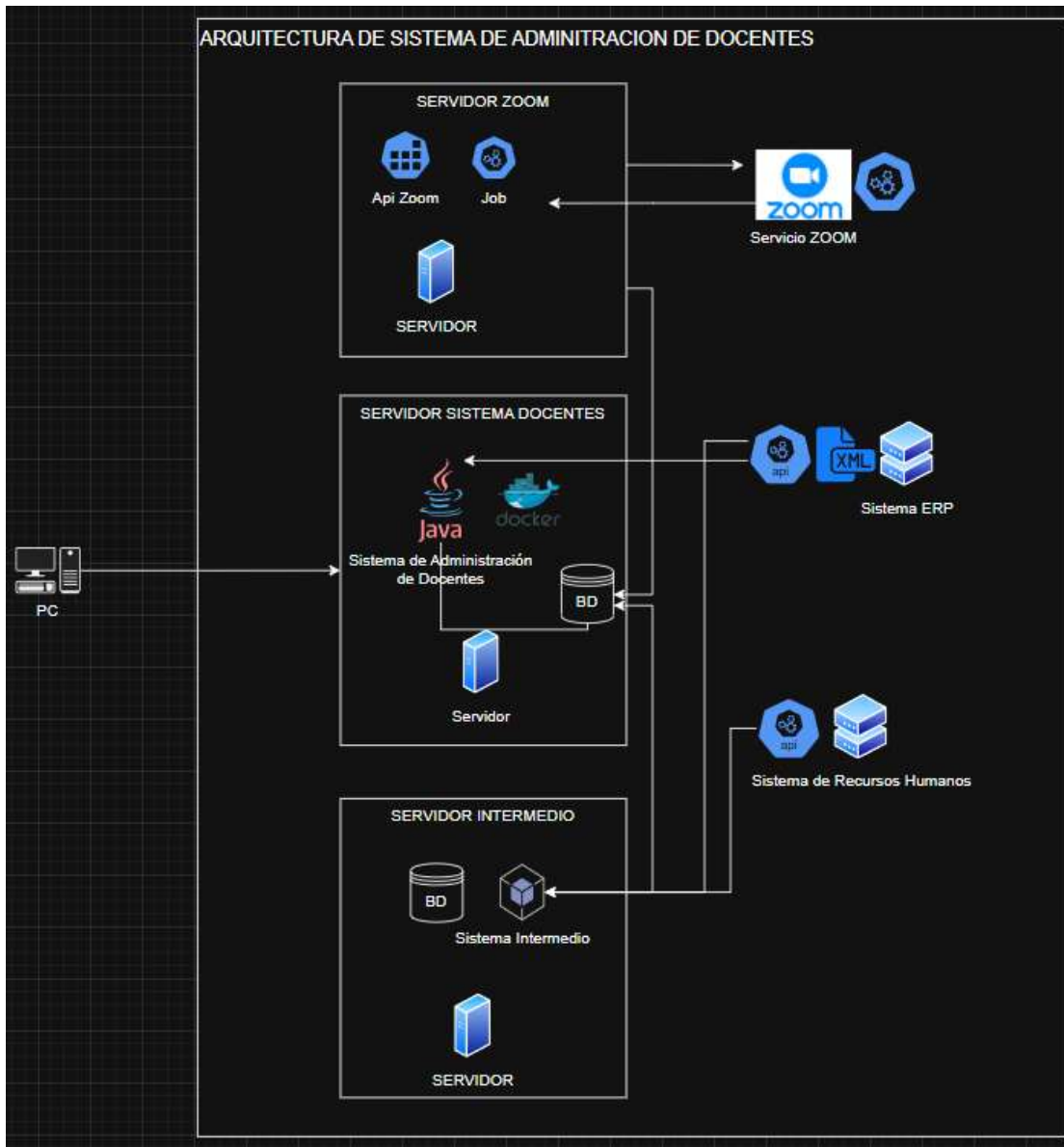
Dentro de la creación del GANTT tenemos varios módulos donde en base a la disposición de cada integrante participante en el desarrollo del proyecto fue ajustada en parte a su especialidad, estas especialidades fueron asignadas según los roles mostrados en la Tabla 1.

Para la generación se basó en una arquitectura de monolítica, acompañada de sistemas de

apoyo, siendo estos pequeños sistemas con comportamientos de microservicios, que apoya al sistema central.

**Figura 7**

*Arquitectura de Software*



Nota: Elaboración Propia

Dentro de la arquitectura del software que se muestra en la figura 7, se realizó en base a separaciones de responsabilidades, teniendo 3 servidores: El primer servidor central es en el que se almacena el sistema de docentes, el cual fue desarrollado en java y dockerizado, almacena el sistema central; dentro del servidor docentes se encuentra la base de datos central donde se encuentran todos los registros necesarios del sistema. Dentro del servidor Zoom se encuentra el sistema Api Zoom, este sistema Api Zoom de igual manera realizado en java se encarga de hacer las consultas en la base de datos del sistema de docentes, observando columnas necesarias para la creación de links Zoom para cada reunión de la clase, por lo cual una vez validado la data necesaria para la creación de nuevas reuniones, realiza la conexión con el Zoom consumiendo su api oficial, para generar links Zoom de forma rápida; este sistema api Zoom también se encarga de cancelar reuniones, eliminándolas. Y, en el servidor Intermedio, se encuentra el sistema Intermedio, este realizado en PHP, el cual se encarga de apoyar tareas complejas al sistema de administración de docentes, como la ejecución de data maestra (categorías, sucursales, docentes, etc), siendo de esta manera ejecuciones directas dentro de este servidor intermedio donde realiza consultas al sistema ERP y al Sistema de Recursos Humanos donde se obtiene la información necesaria para poder hacer las actualizaciones en la base de datos del sistema de Docentes; esto se realiza como apoyo al sistema central y logrando dejar sin tareas recurrentes con complejidad media al sistema de Docentes.

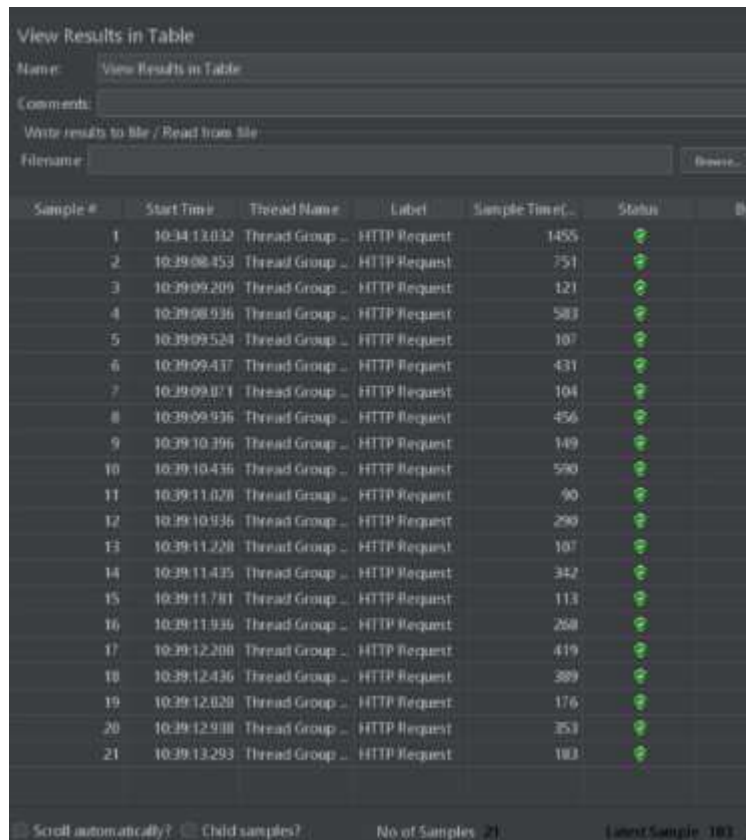
A continuación, se describen las actividades que se muestra en el diagrama de Gantt de la figura 6:

**Carga del proyecto:** Aquí se realizó la configuración inicial del entorno como se muestra en la Figura 8, la conexión con las bases de datos Microsoft SQL SERVER, así como la gestión

para la creación de un servidor con todas las características requeridas para hacer prueba de estrés y lograr observar la estabilidad del sistema dentro de las pruebas de alto rendimiento.

**Figura 8**

*Conexión de base de datos y prueba de estrés*



Sample #	Start Time	Thread Name	Label	Sample Time	Status	Bytes
1	10:39:13.032	Thread Group	HTTP Request	1455	Success	
2	10:39:08.453	Thread Group	HTTP Request	751	Success	
3	10:39:09.209	Thread Group	HTTP Request	121	Success	
4	10:39:08.936	Thread Group	HTTP Request	583	Success	
5	10:39:09.524	Thread Group	HTTP Request	107	Success	
6	10:39:09.417	Thread Group	HTTP Request	431	Success	
7	10:39:09.071	Thread Group	HTTP Request	104	Success	
8	10:39:09.936	Thread Group	HTTP Request	456	Success	
9	10:39:10.396	Thread Group	HTTP Request	149	Success	
10	10:39:10.436	Thread Group	HTTP Request	590	Success	
11	10:39:11.028	Thread Group	HTTP Request	90	Success	
12	10:39:10.936	Thread Group	HTTP Request	290	Success	
13	10:39:11.228	Thread Group	HTTP Request	107	Success	
14	10:39:11.435	Thread Group	HTTP Request	342	Success	
15	10:39:11.781	Thread Group	HTTP Request	113	Success	
16	10:39:11.936	Thread Group	HTTP Request	268	Success	
17	10:39:12.208	Thread Group	HTTP Request	419	Success	
18	10:39:12.436	Thread Group	HTTP Request	389	Success	
19	10:39:12.028	Thread Group	HTTP Request	176	Success	
20	10:39:12.938	Thread Group	HTTP Request	353	Success	
21	10:39:13.293	Thread Group	HTTP Request	183	Success	

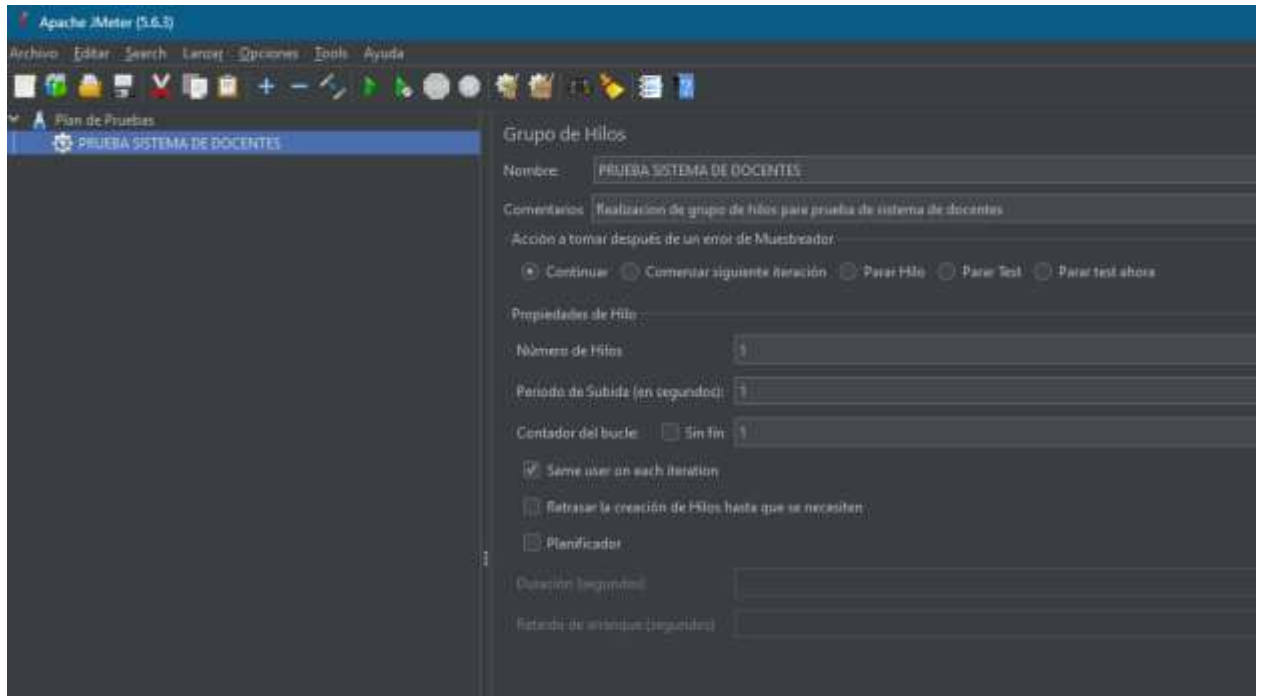
Nota: Elaboración propia

La carga del proyecto se realizó con el software de pruebas Jmeter, usando la opción de HTTP Request, se logró utilizar opciones para realizar peticiones y observar el consumo del servidor, terminado la revisión de la cantidad de consumo de recursos; para la realización de la prueba de estrés se elaboró un plan de pruebas y posteriormente un grupo de hilos para su

ejecución, como se observa en la figura 9.

## Figura 9

### *Grupo de Hilos*



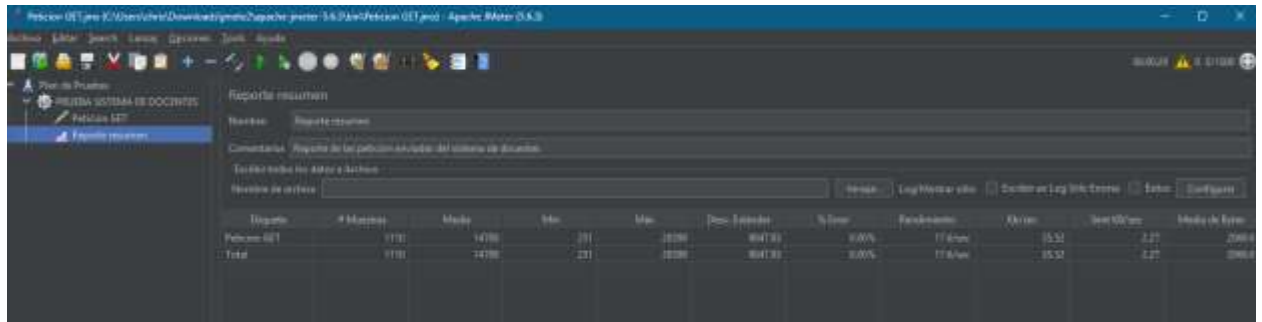
Nota: Elaboración propia

En la figura 9, se muestra características importantes como: los números de hilos, el periodo de subida (en segundos) y un contador de bucle, cada una de estas 3 opciones son importantes ya que permiten simular una muestra en el estrés al sistema instalado en un servidor de pruebas con características de servidor productivo. A continuación, se creó la petición HTTP para registrar las características del consumo dado, tal como se muestra en la figura 10, donde se indica el protocolo, el nombre del servidor o IP, el puerto, el protocolo de la petición y la ruta.



**Figura 12**

*Trata de data de Reporte Resumen*



Objeto	# Matriculados	Medias	Min.	Max.	Peticiones	% Error	Funcionarios	Alumnos	Semanas	Medios de Examen
Peticiones GET	1131	1478	231	2030	84133	0.00%	716767	10532	227	25904
Total	1131	1478	231	2030	84133	0.00%	716767	10532	227	25904

Nota: Elaboración propia

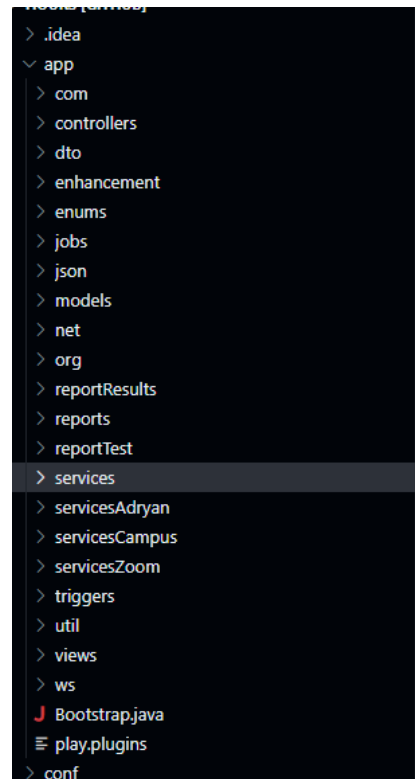
**Creación de sistema de administración de docentes:** Una vez realizadas las pruebas de rendimiento para la primera instancia se procedió al desarrollo del software teniendo en consideración los requerimientos registrados. Estos requerimientos consistieron en: crear un sistema que permita a los usuarios tener un mejor control en la asignación de docentes a las clases respectivas, permitir al sistema obtener el aproximado de matriculados por clase, así como su ubicación y las fechas de inicio y fin, crear un apartado de gestión de horas de planillas para comunicarse con el sistema de recursos humanos y así controlar la cantidad de horas trabajadas por cada docente, implementar un módulo de mantenimiento para gestionar docentes, eventualidades, usuarios y ubicaciones, desarrollar un módulo de reportes que minimice el trabajo manual dentro de la empresa, y generar automáticamente las salas virtuales (Zoom) para cada clase, creando las reuniones correspondientes de forma automática.

Se inició el desarrollo del software con 3 módulos internos para cada sección, al estar usando PlayFramework aplicamos MVC, donde la estructura se basa en un modelo de carpetas que permite tener un buen manejo en el desarrollo del software, tal como se muestra en la figura

13.

### Figura 13

*Estructura MVC del Sistema*



Nota: Obtenida de la estructura del proyecto

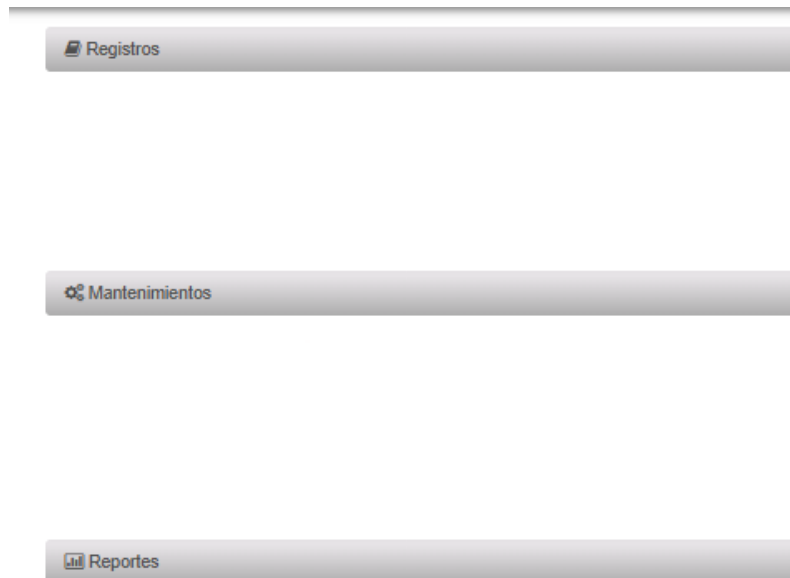
En la carpeta *controllers* se colocan las clases que reciben las peticiones y devuelven respuestas; en *models* están las entidades y el acceso a la base de datos; en *views*, las plantillas que generan la salida (HTML u otros formatos); en *services* y las variantes contienen la lógica de negocio que usan los controladores; *dto* tiene objetos sólo para transporte de datos; *enums* guarda valores constantes del sistema; *json* contiene clases para convertir objetos a json y viceversa; *util* contiene las utilidades y funciones comunes; *jobs* incluye las tareas programadas en segundo plano; *reports*, *reportresults* y *reportest* son carpetas utilizadas para la generación y manejo de

reportes; *ws* se utiliza para websockets o comunicación asíncrona; *enhancement*, *net* y *org* son sólo estructuras de organización para código adicional o externo; *bootstrap.java* se ejecuta cuando arranca la aplicación; y, *play.plugins* registra plugins de Play Framework.

El sistema tuvo 3 grandes módulos: registros, mantenimientos y reportes, como se muestra en la figura 14.

### **Figura 14**

*Módulos principales del sistema*



Nota: Obtenido y modificado del sistema de administración de Docentes.

Dentro del **módulo de registros** se encuentran 4 accesos principales: Clases, Clases (Visualización), Horas para planilla y Eventualidades, tal como se observa en la figura 15.

Todas las vistas internas del sistema se realizaron con HTML en conjunto con el motor de

plantillas de Play Framework “Groovy”, el cual permite extender layouts, insertar variables del backend, hacer condicionales, cargar recursos, rutas, imágenes y más.

## Figura 15

### *Módulo de registros*



Nota: Obtenida del sistema de administración de Docentes.

En el menú Clases, tal como se observa en la figura 16, se tomó en consideración las opciones principales para hacer una búsqueda más directa sin alterar la dificultad de la búsqueda para el usuario final. Los criterios frecuentemente utilizados son: Periodo, Ubicación, Materia, Curso, Frecuencia, Horario, Estado de la clase y el docente.

## Figura 16

### Interfaz de Clases



Nota: Obtenido y modificado de la interfaz de administración de Docentes.

Dentro de la Búsqueda se observa opciones importantes como el Periodo, donde se determina el intervalo de las clases que empiezan en el mes y año; por otro lado, se tiene la ubicación, dentro de esta, están todos los centros físicos donde se dictan las clases de forma presencial y el centro online donde se dictan de forma virtual; dentro de las clases se visualiza la materia; el curso donde se puede elegir los cursos a dictar dependiendo de la clase; la frecuencia donde es la constancia de cada clase a dictar; el horario se basa en el tiempo que empieza y acaba la clase; el estado de la clase siendo esta: abierta, finalizada, cancelada y anulada. Dentro del menú de Clases se observa la tabla de vista previa de las clases, tal como se visualiza en la Figura 17.

**Figura 17**

*Vista previa de clases*

Clase	Periodo	Materia	Frecuencia	Estado de la Clase	Ubicación	Curso	Horario	Docente
1	202510	Matemática	1	Abierta	Presencial	Matemática I	08:00 - 09:00	Dr. Juan Pérez
2	202510	Matemática	1	Abierta	Virtual	Matemática I	09:00 - 10:00	Dr. María Gómez
3	202510	Matemática	1	Abierta	Presencial	Matemática I	10:00 - 11:00	Dr. Carlos Ruiz
4	202510	Matemática	1	Abierta	Virtual	Matemática I	11:00 - 12:00	Dr. Ana López

Nota: obtenido y modificado del sistema de administración de docentes.

Donde hay una vista previa de las características de las clases, dentro de todas las características de las clases, hay 4 opciones que son reuniones, reasignar, sustituir y auditoria como se observa en la figura 18.

### Figura 18

#### *Opciones de clases*



Nota: Obtenido del sistema de administración de docentes.

En la vista previa de clases se encuentra la opción Reuniones; es decir, las reuniones programadas dependiendo de la frecuencia asignada, respetando cada día de la semana la reunión relacionada con su frecuencia, y respetando el horario asignado a la clase. Cada reunión tiene un id característico, dentro de las reuniones, existe el docente a cargo, el cual se encarga de dar la clase en cada reunión. También existe el docente sustituto el cual se encarga de tomar la reunión, en caso por alguna eventualidad de parte del docente principal de la clase no pueda dictar el día pueda sustituirlo, tal como se observa en la figura 19.

**Figura 19**  
*Interfaz de Reuniones*

Periodo	N. Clase	Curso	Org. Acad.	Grado Acad.	Frec.	L	M	M	J	V	S	Horario	Horas	Aula	Empresa	Estado	Docente	F.I.	F.F.	Matriculados	Secund
202	31'		OACENTROS	GENT	Presencial Mixta	X	X		X			18:00 - 21:22	368.48					02/10	24/10	0	

R. Dividida	M	M Padre	Docente Asignado	Docente Sustituto	Even. Sort.	Fecha	H.L. Prog.	Hora Ingreso	H.F. Prog.	Hora Salida	Eventualidades	Ingreso	Dividi	Zona
	5902974					02/10/2024	19:00		21:22		▲	↻	↑	🗑️
	5902975					04/10/2024	19:00		21:22		▲	↻	↑	🗑️
	5902976					07/10/2024	19:00		21:22		▲	↻	↑	🗑️
	5902977					09/10/2024	19:00		21:22		▲	↻	↑	🗑️
	5902978					11/10/2024	19:00		21:22		▲	↻	↑	🗑️
	5902979					14/10/2024	19:00		21:22		▲	↻	↑	🗑️
	5902980					16/10/2024	19:00		21:22		▲	↻	↑	🗑️

Nota: Obtenido y modificado del sistema de administración de docentes.

Dentro de la reunión existe las eventualidades donde es se puede indicar la eventualidad surgida dentro de la reunión y otorga la sustitución por parte del docente en la reunión como se observa en la figura 20.

**Figura 20**  
*Interfaz de Eventualidades*

### Eventualidades

Horario:

Tipo Eventualidad:

Nombre Docente:

Observaciones:

	Eventualidad	Docente	Observaciones
<input type="button" value="Nuevo"/> <input type="button" value="Actualizar"/> <input type="button" value="Eliminar"/>			

Nota: Obtenido y modificado del sistema de administración de docentes.

La otra opción dentro de la vista de reuniones es la de Ingreso como se observa en la figura 21, aquí se determina la hora de ingreso por parte del docente, aquí se registrará el ingreso del docente y su hora de salida.

**Figura 21**  
*Interfaz de Ingreso*



Docente Asignado	35
Fecha De Sesion	12/06/2025
Duracion Programada	15:30 - 17:00
Hora De Ingreso	
Hora De Salida	
Eventualidad	Ninguna
Observaciones	

Nota: Obtenido y modificado del sistema de administración de docentes.

La siguiente opción es la dividir, que se observa en la figura 22, donde cada reunión se puede dividir en intervalos más pequeños, dentro del horario general puede existir 2 reuniones que abarquen la reunión original respetando el horario de inicio y horario final de esta. El propósito de la división de la reunión es en caso exista una eventualidad por parte del docente que no pueda finalizar la clase o empezar la clase a la hora acordada. Para la opción de dividir se utilizaron procedimientos almacenados, estos ayudaron a tener mayor lógica interna en la base de datos, y su manipulación de la data fue de manera más directa.

**Figura 22**  
*Interfaz de Dividir*



The screenshot shows a web form titled "Dividir" with a close button (X) in the top right corner. The form contains the following fields:

- Docente Asignado:** A text input field containing the value "33".
- Fecha:** A date input field containing the value "12/06/2025".
- Hora Inicio Primera Sesión:** A time input field containing the value "15:30".
- Hora Fin Primera Sesión / Hora Inicio Segunda Sesión:** An empty text input field.
- Hora Fin Segunda Sesión:** A time input field containing the value "17:00".

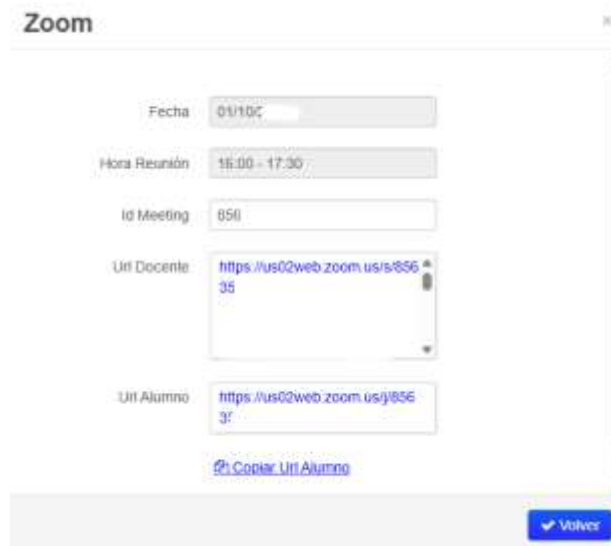
At the bottom right of the form, there are two buttons: a blue "Guardar" button with a checkmark icon and a grey "Cancelar" button with an X icon.

Nota: Obtenido y modificado del sistema de administración de docentes.

La última opción es la de Zoom, aquí se podrá ver los enlaces Zoom de la clase, esta opción sólo está disponible para las clases online, ya que las clases que son presenciales tienen esta opción deshabilitada. Los campos de Zoom como se muestra en la figura 23, tiene 3 atributos importantes: el id de las reuniones, el link del docente y link del estudiante, que son generados por el api Zoom.

**Figura 23**

*Interfaz de Zoom*



Nota: Obtenido y modificado del sistema de administración de docentes.

En el módulo de Mantenimientos como se muestra en la figura 24 se encontrarán diversos módulos esenciales para el sistema donde en su mayoría son tablas maestras para el sistema

**Figura 24**

*Módulo de Mantenimiento*



Nota: Obtenido y modificado del sistema de administración de docentes.

Dentro de los mantenimientos están el Modelo Horario como se visualiza en la figura 25, que es una tabla maestra donde están las horas de inicio y las horas de fin en relación el periodo, estas siendo una tabla maestra puede agregarse o actualizarse, son esenciales para la administración y agrupamiento de las clases.

**Figura 25**

*Interfaz de Modelo Horario*



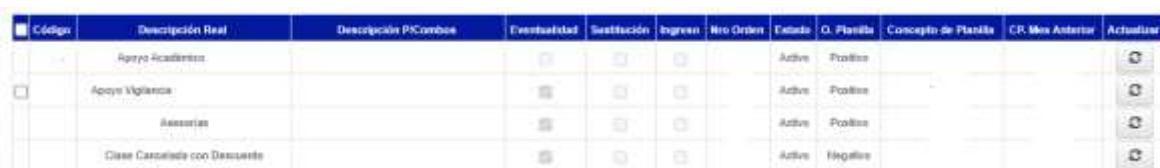
ID	Hora de Inicio	Hora de Fin	Sábado	Activo	Vigente desde	Vigente hasta	Actualizar	URL Usuario	URL Fecha
326	15:00	19:00	<input checked="" type="checkbox"/>	<input type="checkbox"/>	20	20	<input type="button" value="Actualizar"/>	40	12986
325	14:15	17:30	<input type="checkbox"/>	<input checked="" type="checkbox"/>	20	26	<input type="button" value="Actualizar"/>	42	16950
334	18:30	20:00	<input type="checkbox"/>	<input checked="" type="checkbox"/>	20	21	<input type="button" value="Actualizar"/>	56	25942
335	11:25	13:25	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	20	26	<input type="button" value="Actualizar"/>	56	25944

Nota: Obtenido y modificado del sistema de administración de docentes.

Tipo de eventualidad, aquí se visualizan todos los tipos de eventualidades que puedan surgir en cada reunión tal como se ve en la figura 26, indicando si la eventualidad afecta a la planilla o no, dependiendo si la eventualidad es totalmente justificable o no.

**Figura 26**

*Interfaz de Tipo de eventualidad*



Código	Descripción Real	Descripción P/Combo	Eventualidad	Justificación	Ingreso	No Orden	Estado	D. Planilla	Concepto de Planilla	CP. Max Anterior	Actualizar
-	Apoyo Académico		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		Activo	Positivo			<input type="button" value="Actualizar"/>
<input type="checkbox"/>	Apoyo Vigilancia		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		Activo	Positivo			<input type="button" value="Actualizar"/>
	Asesorías		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		Activo	Positivo			<input type="button" value="Actualizar"/>
	Clase Cancelada con Descuento		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		Activo	Negativo			<input type="button" value="Actualizar"/>

Nota: Obtenido y modificado del sistema de administración de docentes.

Docentes, aquí se puede verificar los docentes registrados, así como toda su descripción

completa, sus estados activos o cesados, así como su ubicación del centro a donde pertenecen.

### Figura 27

#### Interfaz de Docentes



Nota: Obtenido y modificado del sistema de administración de docentes.

Usuarios, aquí se pueden administrar los usuarios que tengan acceso al sistema de administración de docentes, otorgándoles permisos para ver las características límites por parte del usuario con el rol descrito.

Esta data es de sólo consultas, que es llenada gracias al sistema intermedio, obteniendo la data por medio de los servicios de recursos humanos, realiza inserciones de forma directa en las tablas de docentes dentro del sistema.

### Figura 28

#### Interfaz de Usuarios



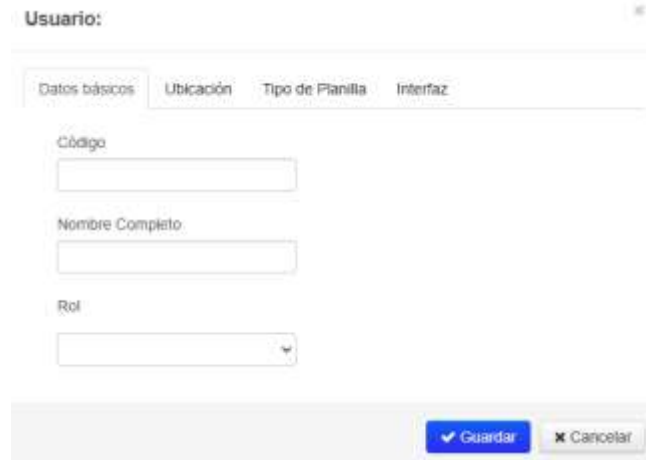
Nota: Obtenido y modificado del sistema de administración de docentes.

Se podrá registrar datos básicos, ubicación, tipo de planilla y la interfaz la cual le da la

potestad de tener límites dentro del sistema.

### Figura 29

#### *Interfaz de Usuario*



Nota: Obtenido y modificado del sistema de administración de docentes.

Ubicaciones, aquí se podrá dar la verificación de cada ubicación o centro por que se maneja dentro del sistema, así como se puede verificar la opción si participa en pase de planilla o no.

### Figura 30

#### *Interfaz de Ubicaciones*

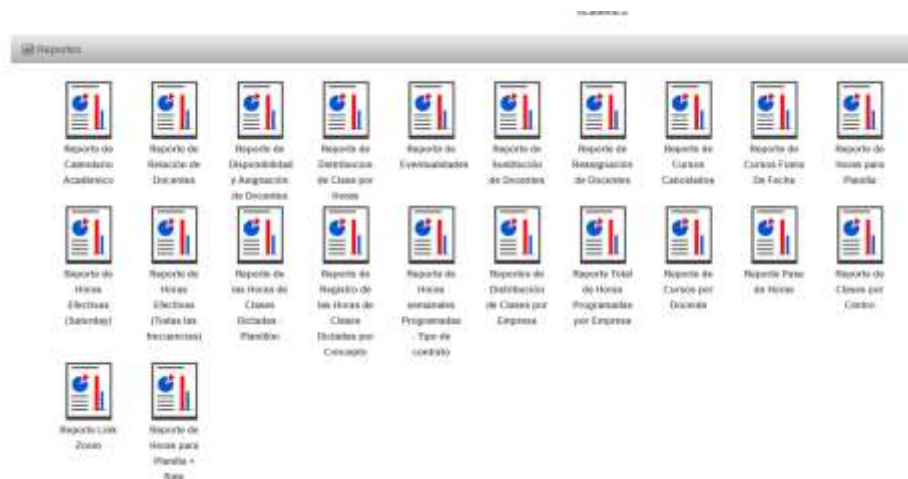
Código	Descripción	Participa en Pase Planilla	Acción	Ult. Usuario	Ult. Fecha
		<input type="checkbox"/>			
		<input type="checkbox"/>			
		<input type="checkbox"/>			
		<input type="checkbox"/>			
		<input type="checkbox"/>			

Nota: Obtenido y modificado del sistema de administración de docentes.

Dentro del último módulo llamado reportes, aquí se encuentran todos los reportes que son necesarios para tener una mejor observación por parte del usuario. De igual manera esta data es llenada por el sistema intermedio obteniendo la data del ERP y luego actualizando de forma directa en la base de datos del sistema.

**Figura 31**

*Menú de reportes*



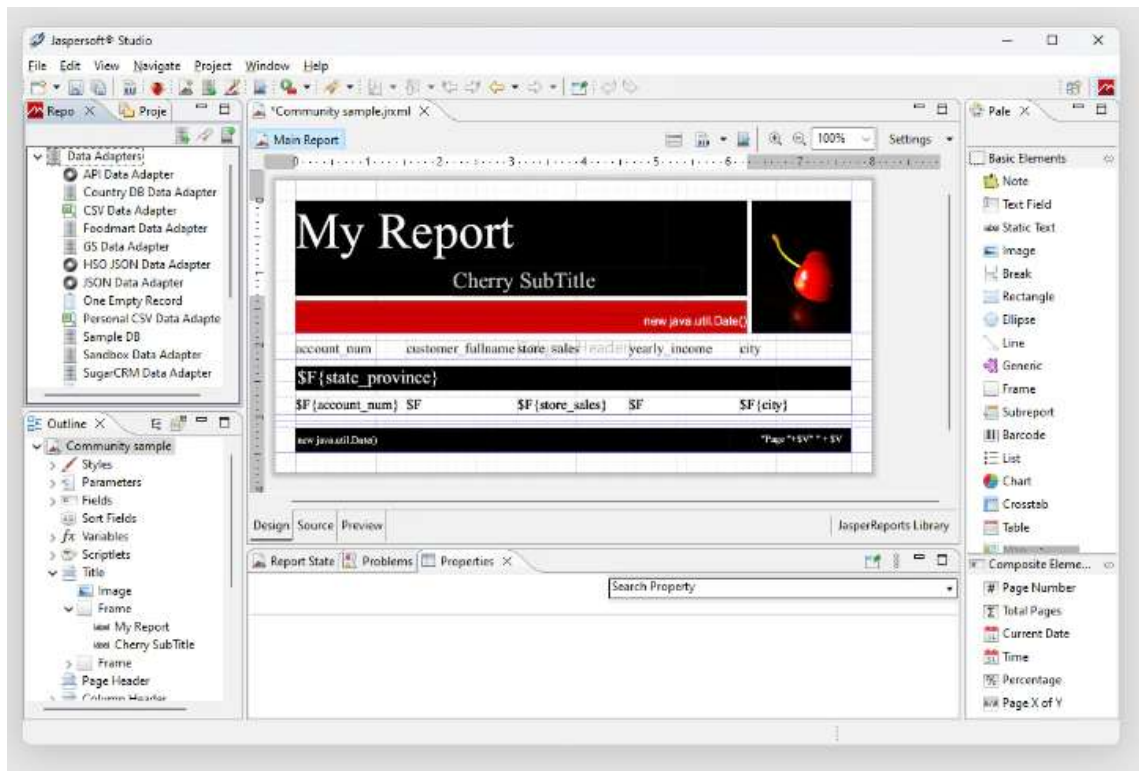
Nota: Obtenido y modificado del sistema de administración de docentes.

Cada reporte tiene una característica importante, todos los reportes pueden ser exportados en Excel y PDF; los reportes funcionan con parámetros como: centro, ubicación, periodo y más. Esto fue logrado gracias a la librería para generar reportes (JasperReports) en base a los objetos internos obtenidos dentro del Framework, logrando hacer las gestiones para generar los reportes en base a los requerimientos; para la generación de todos los reportes se utilizó JasperSoft Studio para así poder tener el modelo del reporte tal como se observa en la figura 32; gracias a ello se

pudo gestionar fácilmente todos los reportes requeridos.

**Figura 32**

*JasperSoft Studio*



Nota: Obtenido de la página oficial de JasperSoft

Reporte de calendario a académico: Aquí se puede observar el calendario académico de cada sede a elegir, en conjunción con el año elegido para verificar y observar el reporte seleccionado.

Reporte de relación de docentes: Aquí se observa los docentes en una vista previa, se observa todas las características de todos los docentes su nombre, fechas de ingreso, teléfono, celular, ubicación y más.

Reporte de Disponibilidad y asignación de Docentes: Uno de los reportes más usados donde se podrá observar la disponibilidad del docente, entre todas las clases, aquí es esencial hacer un filtro ya que para los docentes se podrá elegir dependiendo de la ubicación un mejor filtraje para observar su disponibilidad de acuerdo con sus clases llevadas.

Reporte de distribución de clase Por Horas: Aquí se visualiza las horas y clases distribuidas en grupo para tener una mejor opción de vista para las clases en su agrupamiento, dentro de este reporte está disponible con varios filtrajes como: la ubicación, las frecuencias, el estado del aula, el periodo y más.

Reporte de eventualidades: En este reporte Se visualiza las eventualidad y licencias dadas a los docentes, donde se visualiza el total de horas las eventualidad o licencias y observaciones extras.

Reporte de Sustitución de Docentes: Se podrá ver las sustituciones de los docentes, la fecha, el número de clase, así como la eventualidad toda relacionada a la sustitución del docente.

Reporte de Reasignación de Docentes: Al igual que el reporte de sustitución de docente, en este reporte se verifica la reasignación del docente, donde la reasignación consiste en que el docente toma el control de la clase como docente principal de esta.

Reportes de Cursos Cancelados: El reporte se observa todos los cursos cancelados, dando filtros del docente a cargo y la fecha y horario de la clase, así como su ubicación.

Reportes de Cursos Fuera de Fecha: las clases fuera de fecha son aquellas clases que por alguna excepción superan la fecha estimada por alguna situación única donde todas las clases donde surja este detalle se podrá visualizar.

Reporte de Horas para planilla: Uno de los reportes más utilizado y más importantes donde se visualizan las horas realizadas por cada docente en una vista previa, se puede visualizar con las opciones convenientes en el filtro dependiendo de la ubicación el tipo de planilla, el docente el periodo, la fecha de inicio y fin

Reporte de Horas efectivas (Sábados): Aquí se podrá ver los docentes donde sus horas enseñadas los sábados siendo estos días donde hay más horas son mostradas con una sumatoria total de horas.

Reporte de horas efectivas (Todas las frecuencias): Al igual que en el reporte de hora efectivas, sirve para las demás frecuencias dadas, donde esta las horas, así como el total de horas.

Reporte de Registro de las Horas de Clases Dictadas: Aquí se visualiza las horas normales, así como los descuentos, con cada fecha del periodo, dando un total de horas indicando el descuento con las eventualidades.

Reporte de Registro de las Horas de Clases Dictadas por Concepto: En este reporte se visualiza las horas dictadas para cada docente, así como varias indicaciones de descuento, tardanzas, Apoyo Exámenes de Clasificación y más.

Reporte de Horas semanales Programadas: Dentro de este reporte se encuentra la lista de docente donde se filtra por el tipo de planilla, donde se visualiza las semanas y las horas.

Reporte Total de Horas Programadas por Empresa: Aquí se visualiza las horas dadas en uso de enseñanza por los docentes a empresas específicas

Reporte de Cursos por Docente: Aquí se visualizan los docentes donde se verifican los cursos por cada docente a nivel histórico por periodo, frecuencia, Horario y más.

Reporte Pase de Horas: En este reporte se visualiza el pase de Horas hacia el sistema de recursos humanos donde se visualiza el total de Horas pasadas.

Reporte de Clases por Centro: Se visualiza las clases, las aulas físicas u online, en base al horario, se puede tener más precisión al realizar el filtro por periodo, ubicación, frecuencia, horario y estado de la clase.

Reportes de Link Zoom: Este reporte es esencial en caso de emergencias para obtener el enlace Zoom de cada clase para poder acceder de manera directa, se filtran por el centro y el estado de clase, docente, así como su hora de inicio y hora final.

Reporte de Horas para Planilla + Rate: Aquí se visualiza el centro de costo, el tipo de contrato, el docente, así como el total general de horas pagadas y el tarifario por hora de las horas trabajadas por docente.

**Sistema de administración de docentes con conexión con el sistema de colaboradores de recursos humanos:** Se realizó la conexión con el sistema de recursos humanos para tener un mejor manejo de la planilla por parte de los docentes, ya que el sistema tiene un control donde existen las eventualidades o la cantidad de horas dictadas por los docentes, siendo esta data fácilmente obtenida por el sistema, él envió de la data hacia el sistema de recursos humanos se facilita de manera rápida y segura a través de los servicios de conexión, esta data que se envía a través del sistema, viaja al sistema intermedio y posteriormente a través de esta realiza la conexión hacia el sistema de recursos humanos y de manera viceversa. La trata de esta data se realizo de esta manera para tener la separación de la trata demandante en el servicio intermedio donde se puede almacenar y tratar la data antes de enviarla de manera segura y eficaz.

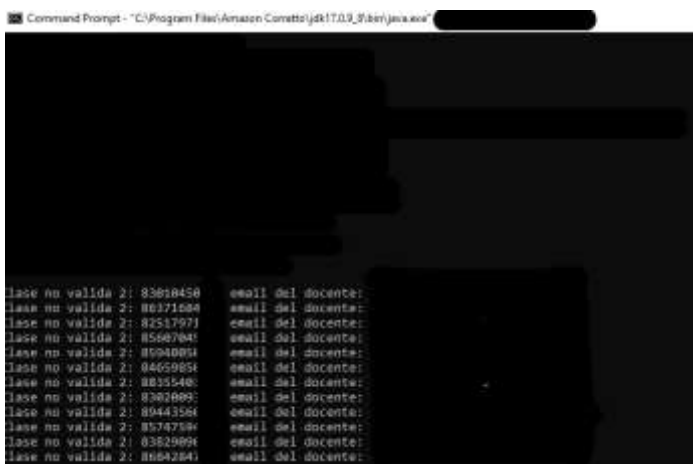
**Figura 33**  
*Interfaz de Horas para planilla*



Nota: Obtenido y modificado del sistema de administración de docentes.

**Api Zoom:** Este módulo fue construido en un servidor apartado del sistema de administración de docentes, siendo un apoyo indispensable. Este job es el necesario para la creación de links Zoom, el cual se conecta directamente a la base de datos del sistema y verifica las tablas principales donde las reuniones son creadas y se observan aquellas reuniones donde no hay links Zoom creados. Para la creación de links se hace uso de la data de las horas, de igual manera verifica la data de las clases canceladas y de las reuniones partidas, generando reuniones o eliminando aquellas que no se utilizarán.

**Figura 34**  
*Job de Api Zoom*



Nota: Obtenido y modificado del sistema de administración de docentes.

La ejecución del api Zoom se realizó tal como se muestra en la imagen de diagrama de software de la figura 7; el api Zoom realizado en java tiene 3 enfoques importantes: creación, cancelación y eliminación de duplicidad.

Para la generación de nuevas clases, se decidió hacer uso de lectura mediante un job que revisa constantemente la columna dentro de las tablas de reuniones, esta tabla está enlazada con la clase que contiene la lógica para la creación de reuniones, siempre que la columna de reunión esté vacía o sin valor, el api Zoom lee esta columna y automáticamente genera links Zoom y luego se hace la inserción con el nuevo link de la reunión.

Para las cancelaciones, sólo se observa que no haya una reunión asignada, ya que el sistema realiza un cambio de estado para las reuniones canceladas por alguna eventualidad, en este caso el api Zoom también lee el estado y las finaliza.

En el caso de las reuniones duplicadas, hace una verificación en las reuniones del docente, en la casuística de que el docente tenga una reunión múltiple asignada por algún motivo, esta se encarga de eliminar las reuniones que no pertenezcan a las oficiales que fueron creadas y estén conectadas al sistema.

**Pruebas de rendimiento de servidores,** las pruebas de rendimiento se realizaron usando software dedicado para cada apartado, estas pruebas se encuentran realizadas para cada módulo, de igual manera se realizaron varias elevaciones al servidor donde no se encontraron anomalías y resistiendo todas las pruebas dadas, dando indicaciones de soportar altas concurrencias.

**Figura 35**

*Prueba de rendimiento del servidor*



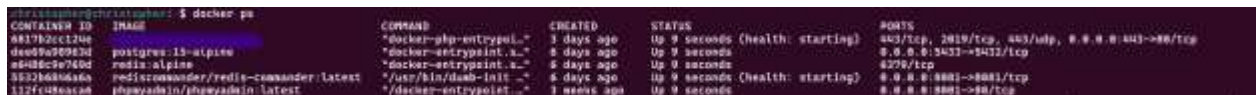
Nota: Porcentaje obtenido del servidor, imagen referencial

**Deploying en Docker:** Aquí se realizó el despliegue en Docker donde la creación de un dockerfile adecuado permitió su deploying fácilmente y en minutos sin dificultad.

El deploying fue generado verificando la liberación de puertos, así como el consumo del servidor, para la facilitación del deploying se utilizaron comandos rápidos proporcionados por Docker, obteniendo la imagen del sistema y la creación del contenedor como se observa en la figura 36.

**Figura 36**

*Vista de contenedores*



Nota: Obtenido y modificado del sistema de administración de docentes internamente en local.

**Actividades Generales:** Se generaron diversos documentos y guías para el uso correcto del sistema, así como videos de enseñanzas para cada módulo y opciones dentro del sistema, estas

guías fueron creadas tanto a nivel de usuario como a nivel técnico.

**Conexión con ERP:** La verificación en la salida a producción con la comunicación con ERP. Incluye procesos automáticos de creación de personas, alumnos, clientes y proveedores, así como la actualización de datos contables y de trabajadores donde fue de suma importancia un análisis en profundidad ante cualquier anomalía surgida.

Respecto a las consideraciones éticas que se han seguido para el desarrollo del proyecto, en base a lo propuesto por (Floridi, 2013), se tuvo la confidencialidad en la trata de la data personal, así como la privacidad en todo momento, donde se realizó implementaciones de seguridad y de control para el flujo sensible de información garantizando la ley 29733 de Protección de datos personales del Perú (Congreso de la República, 2011).

## CAPÍTULO IV. RESULTADOS

Con el desarrollo del proyecto se lograron cumplir los objetivos deseados:

Respecto al **objetivo general** “Desarrollar un sistema web integral de administración de docentes que optimice procesos de asignación completos con generación de reportes”, se logró implementar exitosamente, donde se logró centralizar todos los módulos incluyendo la generación de reportes, los procesos de asignación completos siendo estas creaciones de clases virtuales vinculadas dando la mejora de la gestión de la administración de docentes. Pasando de ciertos episodios de errores de asignaciones de salas a cero. Todos estos tiempos estimados en mejora se observan en la Tabla 2.

**Tabla 2**

*Tiempo estimados en la administración de docentes*

<b>Indicador</b>	<b>Antes</b>	<b>Después</b>	<b>Mejora</b>
Tiempo de asignación completa por sede	5 horas	2 horas	73% reducción
Errores de asignación de salas	12-18 errores/mes	0 errores/mes	100% eliminación
Docentes gestionados	250-300 docentes	250-300 docentes	Proceso optimizado
Clases gestionadas por ciclo	1000 a más clases	1000 a más clases	Gestión centralizada

Respecto al **objetivo específico 1** “Crear el sistema de asignación de docentes a las clases respectivas”, se logró implementar un módulo de asignación para las 14 sedes, logrando permitir una gestión centralizada en tiempo real. Se logró la reducción del tiempo en la asignación en base a la cantidad de consumo lentitud en tráfico masivo por motivo de separación de responsabilidades en servidores de apoyo, gracias a las nuevas tecnologías usadas. Se logró una disponibilidad del 99% mediante una arquitectura en Docker. Se implementaron validaciones automáticas para evitar

conflicto entre horarios de los docentes al asignarlos. Se logró una integración con base de datos SQL Server para mantener la integridad estándar. Todas estas tecnologías implementadas se observan en comparativa dentro de la Tabla 3.

**Tabla 3**

*Comparativa de Tecnologías implementadas.*

Aspecto Tecnológico	Antes	Después
Infraestructura	Servidor único	Arquitectura Docker (3-4 contenedores)
Sistema de caché	Sin implementación	Redis implementado
Validaciones	Manuales	Automáticas en tiempo real
Base de datos	SQL Server sin optimización	SQL Server optimizado con índices

Respecto al **objetivo específico 2** “Crear el módulo de seguimientos de matriculados, así como su ubicación y las fechas de inicio y fecha fin”, se logró el desarrollo de un dashboard donde se muestra en tiempo real la cantidad de matriculados por cada clase de manera directa y precisa, teniendo una distribución por sede y programa. Se logró una implementación de filtros por ubicación referenciando a las sedes. Se realizó un sistema de visualización de fechas de inicio y fin para cada clase o programa. Las consultas rápidas frecuentes se redujeron significativamente a un 80% menos de latencia gracias a las tecnológicas implementada al realizar uso de cache como se observa en la Tabla 4.

**Tabla 4**

*Métricas de consultas y rendimiento*

Indicador	Antes	Después	Mejora
Latencia en consultas	5 segundos	1 segundo	80% reducción
Consultas realizadas diariamente	400-600 consultas	400-600 consultas	Optimizadas

Indicador	Antes	Después	Mejora
Tiempo de actualización de datos	Por actualización	Tiempo real	Instantáneo
Precisión de datos	~95%	99.50%	4.5% incremento
Filtros disponibles	1-2 filtros básicos	4 filtros (sede, programa, fecha, estado)	200% incremento

En el **objetivo específico 3** “Crear un apartado de horas de planillas para hacer la comunicación con el sistema de recursos humanos”, se logró la creación de módulos de sincronización con el sistema de recursos humanos mediante API. Se realizó la automatización de cálculo de horas trabajadas por docente y el registro automático de horas lectivas basado en la asistencia a clases virtuales. Se logró la creación de reporte de horas para el proceso de planilla mensual, reduciendo errores manuales de cálculo de horas al realizarse de manera automática, tal como se observa en la tabla 5.

**Tabla 5**

*Métricas de Procesamiento de Planillas*

Indicador	Antes	Después	Mejora
Errores en cálculo de horas	12-15%	<1%	93% reducción
Tiempo de proceso mensual	2-3 días	3-4 horas	90% reducción
Tiempo de verificación manual	6-8 horas	1-2 horas	75% reducción
Reclamos por errores	25-30/mes	3-5/mes	85% reducción

En el **objetivo específico 4** “Crear un módulo de mantenimiento, donde se pueda establecer los docentes, las eventualidades, los usuarios, las ubicaciones”, se realizaron las implementaciones en base a CRUD, que permitió la gestión de docentes, registro de eventualidades, administración

de usuarios del sistema con roles y permisos, configuración de ubicaciones con sus características de las ubicaciones. Se realizó el sistema de auditoría para rastrear los cambios de datos, control que anteriormente no existía. Estas implementaciones dieron mayor rango de mejora como se observa en la tabla 6.

**Tabla 6**

*Gestión de datos maestros*

<b>Indicador</b>	<b>Antes</b>	<b>Después</b>	<b>Mejora</b>
Tiempo de actualización de docente	5-8 minutos	30-45 segundos	90% reducción
Registros de auditoría	0 (sin control)	2,000-3,000/mes	Control total
Tiempo de registro de eventualidad	3-5 minutos	1-2 minutos	60% reducción

En el **objetivo específico 5** “Crear un módulo de reportes, que minimice el trabajo manual dentro de la empresa”, se logró la realización de los reportes automatizados con la reducción de tiempo dedicado a la generación manual de reportes y permitiendo su exportación en formatos PDF y Excel. Ahora todos los reportes están parametrizados por sede, periodo, docente, programa, lo que brinda acceso a personalización por cada área. Gracias a las nuevas tecnologías implementadas generan un mayor margen de mejora como se observa en la tabla 7.

**Tabla 7**

*Eficiencia en generación de reportes*

<b>Indicador</b>	<b>Antes</b>	<b>Después</b>	<b>Mejora</b>
Tiempo de generación	2-4 horas	2-5 minutos	95% reducción
Errores en reportes	10-15%	<2%	87% reducción
Reportes generados mensualmente	150-200	150-200	Automatizados
Formatos disponibles	1 (manual)	2 (PDF y Excel)	100% incremento

En el **objetivo específico 6** “Crear automáticamente las reuniones de Zoom por cada clase”, se logró la integración exitosa con API de Zoom para creación automática de reuniones dando enlaces únicos para cada sesión de clase. Se logró la implementación del módulo de limpieza de reuniones duplicadas y el envío automático de credenciales de acceso a docentes y estudiantes, reduciendo ampliamente los errores manuales. Donde las configuraciones creadas dieron un mayor margen de mejora como se observa en la tabla 8, así como las métricas de satisfacción por parte de los usuarios como se observa en la tabla 9.

**Tabla 8**

*Métricas de automatización Zoom*

<b>Indicador</b>	<b>Antes</b>	<b>Después</b>	<b>Mejora</b>
Reuniones creadas/mes	800-1,000 (manual)	800-1,000 (automático)	100% automatizado
Errores de configuración	25-35/mes	1-3/mes	92% reducción
Tiempo de configuración/clase	5-7 minutos	0 minutos (automático)	100% reducción
Reuniones duplicadas	15-20/mes	0-2/mes	90% reducción
Tiempo ahorrado mensual	0	4,000-7,000 minutos	~66-116 horas

**Tabla 9**

*Métricas de automatización Zoom*

<b>Grupo de Usuarios</b>	<b>Satisfacción Antes</b>	<b>Satisfacción Después</b>	<b>Incremento</b>
Docentes	60%	95%	35%
Estudiantes	65%	92%	27%
Jefes de sede	55%	90%	35%
Personal administrativo	50%	95%	45%

## CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES

En el desarrollo del proyecto puse en práctica diferentes competencias descritas en el perfil de egreso, que permitieron responder a cada uno de los objetivos planteados.

Tal es así que, se proyecto logró cumplir satisfactoriamente el **objetivo general**, orientado al desarrollo de un sistema web integral para la administración de docentes que permitiera optimizar procesos críticos de asignación, seguimiento académico, gestión de horas de planilla, manejo de datos maestros, generación de reportes e integración automática con plataformas externas. La implementación de una arquitectura dockerizada, una base de datos optimizada en SQL Server, Redis como sistema de caché, un sistema complementario en Laravel y la integración con la API de Zoom permitió construir una solución escalable, segura y alineada con las mejores prácticas de la ingeniería de software. Este conjunto de decisiones técnicas evidencia el dominio de herramientas modernas para la solución de problemas reales, muchas de las cuales consolidé gracias al aprendizaje continuo a lo largo del proyecto y del ejercicio profesional.

Además, el proceso de diseño y desarrollo permitió aplicar principios sólidos del campo profesional, en coherencia con la competencia “El profesional y el mundo”, al analizar el impacto de la solución en la eficiencia institucional, la seguridad de la información, la sostenibilidad operativa y la calidad del servicio educativo. El cumplimiento de la normativa de protección de datos (Ley 29733) y la implementación de mecanismos de seguridad reforzaron el compromiso con “Ética”, asegurando un tratamiento adecuado, responsable y legalmente pertinente de la información sensible.

En relación con el **objetivo específico 1**, la creación del módulo de asignación docente permitió reducir en un 75% el tiempo asociado al proceso y eliminar completamente los errores de asignación. Esta mejora se logró mediante validaciones automáticas y una arquitectura distribuida con disponibilidad del 99%, lo cual refleja el dominio de la competencia “Diseño y desarrollo de soluciones” y “Uso de herramientas”, al emplear tecnologías como Docker, Redis, Play Framework y SQL Server optimizado para resolver un problema complejo con criterios de eficiencia y escalabilidad.

El **objetivo específico 2**, relacionado con el seguimiento de matriculados, se cumplió mediante un dashboard de monitoreo en tiempo real que redujo la latencia en un 80% gracias al uso de caché. Esto evidenció la aplicación de las competencias “Análisis de Problemas” y “Conocimientos de ingeniería”, al analizar el comportamiento del sistema, identificar cuellos de botella y diseñar estrategias de optimización basadas en datos.

El cumplimiento del **objetivo específico 3**, consistente en automatizar la gestión de horas de planilla y su integración con Recursos Humanos, permitió disminuir los errores de cálculo a menos del 1 % y reducir los tiempos de procesamiento de varios días a pocas horas.

En cuanto al **objetivo específico 4**, orientado a la gestión de datos maestros, la implementación de funcionalidades CRUD y un sistema de auditoría con miles de registros mensuales garantizó integridad, trazabilidad y eficiencia. La reducción de más del 90% del tiempo de actualización evidencia la práctica articulada “Conocimientos de ingeniería” y “Uso de herramientas”, al diseñar estructuras de datos consistentes y aplicar técnicas modernas de gestión de información.

El **objetivo específico 5**, referido a la automatización del módulo de reportes, permitió

reducir el tiempo de generación de información de varias horas a sólo minutos, disminuyendo los errores a menos del 2% y permitiendo la exportación en formatos estándar como PDF y Excel. Este logro demuestra la competencia “Comunicación”, al estructurar información compleja de manera clara, precisa y accesible para diversos usuarios institucionales.

Respecto al **objetivo específico 6**, la integración automática con Zoom permitió generar entre 800 y 1,000 reuniones mensuales sin intervención humana y reducir en un 92% los errores de configuración. Este resultado articula competencias como “Aprendizaje a lo largo de la vida”, al incorporar tecnologías emergentes y adaptarse a entornos cambiantes, y “Diseño y desarrollo de soluciones”, al plantear procesos automatizados robustos basados en APIs.

Finalmente, la magnitud del proyecto, la incorporación de nuevas tecnologías, la resolución de problemas complejos y la capacidad de adaptación continua reflejan la práctica real de “Aprendizaje a lo largo de la vida”, competencia esencial para el ejercicio profesional del ingeniero de sistemas en entornos altamente dinámicos.

En conjunto, el proyecto no sólo alcanzó sus objetivos funcionales, sino que fortaleció el desarrollo integral del perfil profesional, al integrar de manera concreta todas las competencias en un caso real, complejo y de alto impacto organizacional, demostrando rigor técnico, responsabilidad ética, pensamiento crítico y capacidad para diseñar soluciones de ingeniería alineadas con estándares actuales de la disciplina.

### **Recomendaciones**

Se recomienda implementar cargas de pruebas más exhaustivas, así como un escalamiento horizontal, implementando balanceadores de carga y realizando optimización de consultas a través de índices estratégicos para mejorar el rendimiento del sistema.

Así también, es importante sugerir implementar una migración gradual de la arquitectura

monolítica central del sistema a una arquitectura de microservicios para mejorar la mantenibilidad y escalabilidad, de esta forma se podrá tener un mejor control de cada sector por componente del sistema, donde se podrá gestionar actualizaciones dependiendo de las necesidades del módulo.

Finalmente, se recomienda implementar la integración de herramientas de monitoreo para observar el rendimiento en tiempo real y así poder lograr la detección temprana de anomalías y análisis de logs.

## REFERENCIAS

- Aguirre, S. (2024). *Laravel Curso Completo*. Bogota: Ra-Ma.
- Álvaro Iradier, I. M. (2021). *Docker para DevOps*. -: Tecnología espublico.
- Congreso de la República. (2011). *Ley N.º 29733: Ley de Protección de Datos Personales*. Lima, Perú: Diario Oficial El Peruano.
- Diario Oficial El Peruano. (27 de 09 de 2023). *Data Science en Perú: demanda laboral crece en 34%, superando la tasa global del 28%*. Obtenido de <https://www.elperuano.pe/noticia/223991-data-science-en-peru-demanda-laboral-crece-en-34-superando-la-tasa-global-del-28>
- Equipo de Adobe para empresas. (23 de 05 de 2025). *business adobe*. Obtenido de business adobe: <https://business.adobe.com/blog/basics/waterfall>
- Floridi, L. F. (2013). *The Ethics of Information*. United Kingdom: Oxford University Press (OUP).
- Hernández, J. S. (2006). *SQL SERVER VS MySQL*. Ecuador: UNIVERSIDAD TÉCNICA DE COTOPAXI.
- huaman, C. (2010). *el poder oculto*. Peru: Santillana.
- Java The Complete Reference Ninth Edition. (2014). En H. Schildt, *Java The Complete Reference Ninth Edition Comprehensive Coverage of the Java Language* (pág. 233). Estados Unidos: McGraw- Hill Education.
- Maida, E. G., & Pacienza, J. (2015). *Metodologías de desarrollo de software*. Argentina: Cátedra Seminario de Sistemas , Biblioteca digital de la universidad Católica Argentina.
- Nielsen, D. (2023). *Usos populares de Redis para principiantes*. 700 E El Camino Real, Suite 250 Mountain View, CA 94040: redislabs.

Susnjara, S., & Smalley, I. (27 de 06 de 2025). *IBM*. Obtenido de ¿Qué son las pruebas de software?: <https://www.ibm.com/es-es/think/topics/software-testing>

*What is Play?* . (2025). Obtenido de playframework: <https://www.playframework.com/>