

	UNIVERSIDAD FRANCISCO DE PAULA SANTANDER OCAÑA			
	Documento	Código	Fecha	Revisión
FORMATO HOJA DE RESUMEN PARA TRABAJO DE GRADO		F-AC-DBL-007	08-07-2021	B
Dependencia DIVISIÓN DE BIBLIOTECA		Aprobado		Pág.
		SUBDIRECTOR ACADEMICO		1(1)

RESUMEN – TRABAJO DE GRADO

AUTORES	Edwin Fernando Torrado Arenas		
FACULTAD	Ingenierías		
PLAN DE ESTUDIOS	Ingeniería de sistemas		
DIRECTOR	Byron Cuesta Quintero		
TÍTULO DE LA TESIS	Diseño de una nueva versión del sistema de información académica GENESIS SIA realizado para el proyecto de interconectividad de la Universidad Francisco De Paula Santander Ocaña		
TITULO EN INGLES	Design of a new version of the academic information system GENESIS SIA for the interconnectivity project of the Universidad Francisco De Paula Santander Ocaña.		
RESUMEN (70 palabras)			
<p>El presente documento contiene el diseño de una nueva versión del sistema de información académico Genesis SIA de la Universidad Francisco De Paula Santander Ocaña. El nuevo diseño incluye una nueva arquitectura de software que se adapta a las necesidades de escalabilidad y mantenibilidad del software, también plantea nuevas tecnologías para el desarrollo de este, por último, se diseñaron las interfaces de usuario (UI) con el propósito de mejorar la experiencia de usuario para lograr una interfaz de usuario limpia, agradable y de fácil manejo.</p>			
RESUMEN EN INGLES			
<p>This document contains the design of a new version of the academic information system Genesis SIA of the Universidad Francisco De Paula Santander Ocaña. The new design includes a new software architecture that adapts to the needs of scalability and maintainability of the software, also raises new technologies for the development of this, finally, the user interfaces (UI) were designed in order to improve the user experience to achieve a clean, pleasant and easy to use user interface.</p>			
PALABRAS CLAVES	Arquitectura de software, Diseño UI/UX, Sistema de información Académico, Interfaces de usuario.		
PALABRAS CLAVES EN INGLES	Software Architecture, UI/UX Design, Academic Information System, User Interfaces.		
CARACTERÍSTICAS			
PÁGINAS: 160	PLANOS: 0	ILUSTRACIONES: 63	CD-ROM: 0



Vía Acolsure, Sede el Algodonal, Ocaña, Colombia - Código postal: 546552
 Línea gratuita nacional: 01 8000 121 022 - PBX: (+57) (7) 569 00 88
 atencionalciudadano@ufpso.edu.co - www.ufpso.edu.co

**Diseño de una nueva versión del sistema de información académica GENESIS SIA
realizado para el proyecto de interconectividad de la Universidad Francisco De Paula
Santander Ocaña**

Edwin Fernando Torrado Arenas

Facultad de Ingenierías, Universidad Francisco De Paula Santander Ocaña

Ingeniería De Sistemas

Msc. Byron Cuesta Quintero

12 de agosto de 2023

Agradecimientos

En este trascendental capítulo de mi vida, deseo extender mis más sinceros agradecimientos a Dios, cuya presencia ha iluminado mi camino a lo largo de esta travesía académica.

Asimismo, quiero dedicar un espacio especial de agradecimiento a mis padres, quienes han sido mi faro en las aguas turbulentas de la educación superior. Su amor incondicional, sacrificio y constante apoyo han sido el cimiento sobre el cual he construido este proyecto.

Quiero expresar un agradecimiento especial a la señora Omaira Ortiz Mora y su esposo Leonardo Sánchez Paba, quienes fueron un apoyo invaluable en el inicio de mi carrera. Su confianza, orientación y aliento inicial fueron fundamentales para encaminarme hacia este logro. Su generosidad y apoyo han dejado una huella duradera en mi camino académico y les estoy profundamente agradecido por ser parte esencial de mi trayectoria.

Este logro no solo es mío, sino también de aquellos amigos, profesores y seres queridos que han formado parte de mi trayectoria. A todos ustedes, les agradezco por su aliento, por sus palabras de sabiduría y por ser mi red de apoyo en los momentos de duda. Cada interacción ha dejado una marca indeleble en mi camino hacia la culminación de esta tesis.

Índice

Capítulo 1. Diseño De Una Nueva Versión Del Sistema De Información Académica	
GENESIS SIA Realizado Para El Proyecto De Interconectividad De La Universidad Francisco	
De Paula Santander Ocaña.....	
	11
1.1. Planteamiento Del Problema	11
1.2. Formulación Del Problema.....	12
1.3. Objetivos.....	13
1.3.1. Objetivo General	13
1.3.2. Objetivos Específicos.....	13
1.4. Justificación.....	14
1.5. Delimitaciones	16
1.5.1. Delimitación Operativa	16
1.5.2. Delimitación Conceptual.....	17
1.5.3. Delimitación Geográfica	17
1.5.4. Delimitación Temporal	17
2. Marco Referencial.....	18
2.1. Marco Histórico	18
2.1.1 Antecedentes Internacionales	18
2.1.2 Antecedentes Nacionales.....	19
2.1.3 Antecedentes locales	20
2.2. Marco Contextual	22
2.2.1. Organigrama.....	28

2.2.2.	Mapa de Procesos.....	30
2.2.3.	Sistema de Información Académico (SIA)	31
2.2.4.	Según documentos de las carpetas del Genesis SIA	33
2.3.	Marco Conceptual.....	35
2.4.	Marco Teórico	37
2.4.1.	Teorías de Sistemas de Información Académica	37
2.4.2.	Teorías sobre el desarrollo en PHP	39
2.4.3.	Teorías sobre el desarrollo a través de patrones de diseño	40
2.5.	Marco Legal.....	41
2.5.1.	Ley 1581 de 2012 - Protección de datos personales	41
2.5.2.	Ley 603 de 2000 - Ley de software.....	41
2.5.3.	NTC-ISO/IEC 12207.....	42
2.5.4.	Resolución 1409 de 2012	42
2.5.5.	Política de Tratamiento de Datos Personales de la UFPS	42
2.5.6.	Reglamento Estudiantil de la Universidad Francisco de Paula Santander...	43
2.5.7.	Para el caso de los Sistemas de información académica, se encuentran las siguientes leyes:	43
2.5.8.	Para el caso del desarrollo de software en PHP, se encuentran las siguientes leyes: 44	
2.5.9.	Para el caso de los Patrones de diseño, se encuentran las siguientes leyes:..	45
3.	Diseño Metodológico.....	46
3.1.	Tipo de investigación.....	46
3.2.	Población y muestra.....	47

3.2.1. Población.....	47
3.2.2. Muestra.....	47
4. Resultados.....	49
4.1. Primer Objetivo	49
4.1.1. Entrevista a los desarrolladores del Genesis SIA v3.0.....	50
4.1.2. Reuniones con los directivos de los colegios seleccionados.....	52
4.1.3. Encuestas a los usuarios del sistema Genesis SIA v3.0.....	62
4.2. Segundo objetivo	63
4.2.1. Documentación de tecnologías y arquitecturas de software	63
4.2.2. Analizar las tecnologías y arquitecturas de desarrollo ya documentadas	88
4.2.3. Selección de las tecnologías y arquitecturas que se adaptan al contexto del	107
4.3. Tercer objetivo.....	109
4.3.1. Diseño componentes de arquitectura de software del Genesis SIA versión	110
4.3.2. Diseños de interfaces UI del sistema de información académica Genesis SIA	117
4.4. Conclusiones.....	140
4.5. Recomendaciones	142
4.6. Referencias.....	143
4.7. Apéndices.....	155

Listado de Figuras

Figura 1. <i>Ubicación Geográfica de Ocaña Norte de Santander.</i>	22
Figura 2. <i>Logotipo Universidad Francisco de Paula Santander Ocaña, UFPSO</i>	24
Figura 3. <i>Valores Institucionales UFPS Seccional Ocaña</i>	27
Figura 4. <i>Organigrama Institucional Primera Parte.</i>	28
Figura 5. <i>Organigrama Institucional Segunda Parte.</i>	29
Figura 6. <i>Organigrama Institucional Última Parte</i>	30
Figura 7. <i>Guía Organigrama Institucional.</i>	30
Figura 8. <i>Mapa de Procesos Institucional.</i>	30
Figura 9. <i>Sistema de Información Académico (SIA) - Página Oficial</i>	32
Figura 10. <i>Logotipo de la plataforma Génesis SIA.</i>	33
Figura 11. <i>Módulos de la plataforma Génesis SIA.</i>	34
Figura 12. <i>Resultados de la pregunta ¿Qué tan satisfecho/a está con el uso que le da al Genesis SIA v3?</i>	55
Figura 13. <i>Resultados de la pregunta ¿Cómo calificaría la facilidad de uso del Genesis SIA en su jornada laboral?</i>	56
Figura 14. <i>Resultados de la pregunta ¿Cómo calificaría el soporte que ofrece el Genesis SIA al momento de resolver solicitudes que usted ha realizado?</i>	56
Figura 15. <i>Resultados de la pregunta ¿Qué tan satisfecho/a está con la confiabilidad del sistema Genesis SIA v3?</i>	57
Figura 16. <i>Resultados de la pregunta ¿Qué tan satisfecho/a está con la apariencia del sistema Genesis SIA v3?</i>	58

Figura 17. Resultados de la pregunta <i>¿Con qué frecuencia encuentra errores en el sistema Genesis SIA v3?</i>	59
Figura 18. Resultado de la pregunta <i>¿Cree necesario la creación de un nuevo sistema información académica Genesis SIA?</i>	60
Figura 19. Diagrama de contexto del Genesis SIA v4.0.	111
Figura 20. Diagrama de contenedores de Genesis SIA v4.....	112
Figura 21. Diagrama de componentes	116
Figura 22. Interface de login.....	118
Figura 23. Interface Módulos.....	118
Figura 24. Módulo Institución educativa	119
Figura 25. Interface agregar institución educativa	120
Figura 26. Interface de gestión de sede	120
Figura 27. Interface agregar sede.....	121
Figura 28. Interface gestión de salones	121
Figura 29. Interface de modulo grados y grupos	122
Figura 30. Interface gestión de grados	122
Figura 31. Interface gestión de grupos	123
Figura 32. Interface gestión de personas	123
Figura 33. Interface de gestión de docente	124
Figura 34. Interface gestión de administrativos	124
Figura 35. Interface de área y asignaturas.....	125
Figura 36. Interface de periodos y horarios	125
Figura 37. Interface de gestión de alumnos	126

Figura 38. <i>Interface de gestión de matricula</i>	126		
Figura 39. <i>Interface de módulo concepto evaluativo</i>	127		
Figura 40. <i>Interface de módulo de notas</i>	127		
Figura 41. <i>Interface de módulo de reportes</i>	128		
Figura 42. <i>Interface de cambio contraseña</i>	129	Figura 43. <i>Interface de gestión de mensajes</i>	129
.....	129	Figura 44. <i>Interface de log</i>	
.....	130	Figura 45. <i>Interface de gestión errores</i>	
Módulo de configuración.....	130	Figura 46. <i>Interface de m</i>	
Cambiar contraseña.....	131	Figura 47. <i>Modulo camb</i>	
Carga académica por curso	131	Figura 48. <i>Carga académ</i>	
Búsqueda áreas requeridas	132	Figura 49. <i>Módulo de bú</i>	
Gestión de área	132	Figura 50. <i>Módulo de ge</i>	
Asignación de áreas requeridas	133	Figura 51. <i>Módulo gestio</i>	
Asignación de carga académica por curso	133	Figura 52. <i>Módulo gestio</i>	
Asignación de curso	134	Figura 53. <i>Módulo gestio</i>	
Asignar curso por sede	134	Figura 54. <i>Módulo agreg</i>	
Agregar intensidad horaria	135	Figura 55. <i>Módulo para</i>	
Eliminar intensidad horaria	135	Figura 56. <i>Módulo busca</i>	
Gestión de periodos.....	136	Figura 57. <i>Módulo de ge</i>	
Agregar nuevo alumno.....	136	Figura 58. <i>Módulo para</i>	
Gestión de salones.....	137	Figura 59. <i>Módulo de ge</i>	
Matrícula por grados.....	137	Figura 60. <i>Módulo de m</i>	

Figura 61. <i>Módulo de gestión de eventos</i>	138
Figura 62. <i>Módulo de gestión de mensajes</i>	138
Figura 63. <i>Módulo de prematricula</i>	139

Listado de Tablas

Tabla 1. <i>Cuadro Comparativo para Tecnología de Bases de Datos</i>	88
Tabla 2. <i>Cuadro Comparativo para Tecnología de Lenguajes de Programación</i>	90
Tabla 3. <i>Cuadro Comparativo para Tecnología de Framework</i>	93
Tabla 4. <i>Cuadro Comparativo para Tecnología de Servidores Web</i>	95
Tabla 5. <i>Cuadro Comparativo para Tecnología Frontend</i>	97
Tabla 6. <i>Cuadro Comparativo para Tecnología de Sistemas Operativos</i>	99
Tabla 7. <i>Cuadro Comparativo para Tecnología de Seguridad</i>	101
Tabla 8. <i>Cuadro Comparativo para Arquitecturas de Software</i>	104

Capítulo 1. Diseño De Una Nueva Versión Del Sistema De Información Académica GENESIS SIA Realizado Para El Proyecto De Interconectividad De La Universidad

Francisco De Paula Santander Ocaña

1.1. Planteamiento Del Problema

El avance de las tecnologías y el desarrollo de software a traído como tendencia la adopción de la sistematización de procesos en los sectores empresariales, gubernamentales y educativos permitiendo quitarle un gran peso, ahorro de tiempo y optimización de recursos al momento de ejecutar los procedimientos contemplados en ese proceso. A pesar de esto son muchas las instituciones que no cuentan con estos avances en sus procesos quedando relegadas en el desarrollo productivo estas instituciones siguen con este tipo de problemas generando una serie de dificultades como la pérdida de tiempo, perdida de información, procesos ineficientes, etc. (Larco Guzmán, Geovanny; Navarrete Zurita, 2020).

Para las instituciones educativas debe ser prioridad la inclusión de las tecnologías de la información y las comunicaciones en busca de lograr sobresalir sobre las que trabajan de forma tradicional, la adopción de estos sistemas permite facilitar la innovación y la reducción de costos. Para ello la adopción de sistemas de gestión académica es una herramienta esencial para ser competitivos y estar a la vanguardia mejorando sus servicios y entorno laboral (Jaimes, Juan; García, 2016).

Para satisfacer estas necesidades la Universidad Francisco De Paula Santander Ocaña por medio del proyecto INTERCONECTIVIDAD ha desarrollado el sistema de gestión académica Genesis SIA el cual se puso en marcha dirigido a instituciones de educación básica y media de la región en la cual en el 2016 ya se le daba soporte a 11 instituciones de los departamentos Norte De Santander, Cesar y Sucre brindando los servicios de sistemas información académica (Jaimes, Juan; García, 2016).

Este proyecto lleva funcionando un aproximado de 12 años en los cuales ha brindado soporte a diferentes instituciones de educación en la región, este software fue desarrollo con la tecnología PHP él era muy demandado en el momento de la creación del proyecto, los desarrolladores del ese momento utilizaron prácticas de desarrollo las cuales ya no están vigentes en estos momentos o que se han modificado presentados problemas al momento de darle soporte al software.

El actual Génesis SIA es un monolito de código hecho en PHP el cual presenta dificultades al momento de darle soporte al programa lo cual se ha convertido en un dolor de cabeza al presentarse errores o querer implementar nuevas funcionalidades lo cual es casi imposible por la antigüedad de algunas librerías y de código antiguo el cual es altamente acoplado dificultando la modificación o implementación de nuevas funcionalidades.

1.2. Formulación Del Problema

¿Cómo se deberá realizar el diseño de la nueva versión del Génesis SIA con nuevas tecnologías que brinden practicidad y mantenibilidad para los desarrolladores lo cual permita el crecimiento del proyecto de extensión en la Universidad Francisco De Paula Santander seccional Ocaña?

1.3. Objetivos

1.3.1. Objetivo General

Diseñar la nueva versión del sistema Génesis SIA para la gestión de información académica de los colegios de educación básica y media realizado por la oficina de interconectividad de la Universidad Francisco De Paula Santander seccional Ocaña.

1.3.2. Objetivos Específicos

Analizar el contexto, a través del levantamiento de información clave y con ello tener un entendimiento específico de los requisitos del sistema.

Caracterizar tecnologías y aspectos de arquitectura con ello sus atributos y propiedades a través de modelos comparativos.

Diseñar los componentes arquitectónicos a partir del análisis y caracterización previamente realizada proponiendo así la estructura del sistema.

1.4. Justificación

Los sistemas de información académica (SIA) es una solución para facilitar y agilizar todos los procesos de formulación, seguimiento, administración y evaluación de los programas y actividades académicas en las instituciones educativas con el objetivo de contar con la información completa de todos los procesos y actividades realizadas por estas instituciones, este tipo de software viene a solucionar la gestión académica brindando la digitalización de la información y mejora al administración de está permitiendo inclusión, actualización y consulta de la información asociada a los estudiantes, docentes y administrativos de las instituciones

(Elizendo Rodríguez, 2012)

<p>de nivel básico y medio cuentan con una la accesibilidad a la educación que se ha de de que las instituciones de educación tenga entre sus procesos internos para este último vicios de software que mejoran la gestión de tación de notas, estudios curriculares, con el fin de mejorar el proceso brindando</p>	<p>[REDACTED]</p>	<p>En la actualidad gran cantidad de estu mejorando con los añ como objetivo medien los colegios han opta la información acadé prácticas generatié</p>
<p>ta para la mejorar del trabajo docente control de sus cursos (Fernández, 2016).</p>	<p>[REDACTED]</p>	<p>eficiencia en ellos y t facilitando sus activid</p>
<p>Santander seccional Ocaña haciendo uso de SIA v3 el cual es un software de gestión</p>	<p>[REDACTED]</p>	<p>Por esto la Un su programa de exten</p>

académica con el cual brinda este servicio a los centros educativos de la región el cual lleva en funcionamiento 12 años y cuenta con 11 instituciones de los departamentos de Norte De Santander y el Cesar en las cuales este sistema les brinda una gran ventaja al momento de desarrollar sus actividades de administración y gestión de la información de estudiantes, profesores y administrativos de los centros educativos de educación básica y media de la región.

Teniendo en cuenta la longevidad de este sistema en los últimos años se han presentado dificultades en el mantenimiento y soporte del Genesis SIA ya que en su momento de desarrollo se tomaron técnicas de desarrollo de software que hoy en día son obsoletas o ya no se usan porque dificultan el mantenimiento y la escalabilidad del software restringiendo la posibilidad de llegar a un nivel de usuarios más grande, otra problemática presente en este sistema es la dificultad para encontrar, tratar y depurar error en el sistema ya que este es un monolito de código con un alto grado de acoplamiento en todas sus clases lo que hace casi imposible cambiar el código de cierta clase o método porque es imposible saber a qué otras instancias se verán afectadas a estos cambios por lo cual los desarrolladores de la oficina de extensión optan por crear nuevos métodos o clases que den solución a ese error lo que causa que cada vez sea más grande el monolito de código y más difícil depurarlo.

Por tal motivo con este proyecto se plantea la nueva versión de este software tomando en cuenta los errores cometidos en la actual versión y haciendo gran énfasis en el diseño del software tomando en cuenta nuevas tecnologías y patrones de diseño los cuales permitan tener un sistema de gestión académica de fácil mantenimiento, con gran capacidad de escalamiento y mejorando la experiencia de usuario para nuestros clientes, brindando software de calidad que

pueda competir con los demás sistemas de gestión académica del mercado y mostrando la calidad del software desarrollado por la Universidad Francisco De Paula Santander Ocaña.

Y el último aspecto, y no menos importante, es la experiencia a nivel personal que obtendré con el proyecto me ayudarán a seguir creciendo en mi vida profesional, ya que, al realizar el trabajo en una empresa de otro país, me permite tener una posibilidad de proyección internacional, adicional a la universidad le brinda indicadores de internacionalización y posibilita futuros convenios que servirán para la realización de pasantías y prácticas profesionales.

1.5. Delimitaciones

1.5.1. Delimitación Operativa

Para el desarrollo de este proyecto se contará con la asesoría del Ingeniero Byron Cuesta Quintero el cual tiene el papel de director de este proyecto, junto con la asesoría de los Ingenieros Juan Camilo Jaimes Fernández y Luis Eduardo Hernández Suarez por parte de la oficina de Interconectividad para el apoyo en la verificación y aceptación de cada uno de apartados del desarrollo del proyecto.

Como la plataforma maneja alguna información sensible, se podrá tener alguna restricción a la misma.

1.5.2. Delimitación Conceptual

A nivel del proyecto se analizan conceptos como: sistema de información, sistema de información académica, Java, Framework, MVC, Sprint Boot, entre otros.

1.5.3. Delimitación Geográfica

El proyecto será desarrollado en el Municipio de Ocaña, Norte de Santander.

1.5.4. Delimitación Temporal

Para el proyecto se estima una duración de seis (6) meses.

2. Marco Referencial

2.1. Marco Histórico

Los sistemas de información académicos son una herramienta fundamental en la gestión de la información en las instituciones educativas. Su evolución y desarrollo han estado marcados por importantes avances en el campo de la tecnología de la información y la comunicación, así como por los cambios en las necesidades y expectativas de los usuarios. En este marco histórico, se pueden identificar diversas secciones que marcan los hitos más relevantes en el desarrollo de los sistemas de información académicos, a nivel internacional, nacional y regional.

2.1.1 *Antecedentes Internacionales*

El desarrollo de los sistemas de información académicos en los Estados Unidos comenzó en la década de 1960, cuando las instituciones de educación superior comenzaron a utilizar computadoras para gestionar la información académica de los estudiantes (Liao, 2008). En la década de 1970, se crearon sistemas de información académicos más sofisticados que permitían la automatización de tareas administrativas y académicas, como la emisión de certificados y la elaboración de horarios de clases.

En este sentido, se pueden destacar otros antecedentes relevantes. En Europa, uno de los antecedentes más relevantes en el desarrollo de sistemas de información académicos es el Proyecto ERASMUS, creado en 1987 para promover la cooperación entre instituciones de

educación superior en Europa (European Commission, 2021). Como parte de este proyecto, se creó el Sistema de Información ERASMUS (EUSIS), una plataforma que permitía la gestión de la movilidad de estudiantes y profesores entre distintas instituciones europeas. Asimismo, en el Reino Unido, la implementación de sistemas de información académicos se consolidó con la creación del Student Records System (SRS) en la década de 1990, el cual permitía la gestión de datos de estudiantes y la generación de reportes para la toma de decisiones académicas (Jisc, 2017)

En Asia, se destaca el Sistema de Información de la Universidad de Tsukuba (UTIS), implementado en Japón en la década de 1990, el cual permitía la gestión de la información académica de los estudiantes y la emisión de certificados (University of Tsukuba, 2023).

2.1.2 Antecedentes Nacionales

En el ámbito nacional, los sistemas de información académicos han evolucionado significativamente en los últimos años. Su desarrollo ha sido impulsado por la necesidad de las instituciones educativas de contar con herramientas tecnológicas que les permitan una gestión eficiente y efectiva de la información. A continuación, se presentan algunos antecedentes relevantes en el desarrollo de los sistemas de información académicos en Colombia.

Uno de los primeros antecedentes relevantes en la implementación de sistemas de información académicos en Colombia es el Sistema de Información Académica (SIA) de la Universidad Nacional de Colombia, el cual fue creado en 1988 y permitió la automatización de

procesos académicos como la inscripción, la gestión de matrículas y el registro de calificaciones (Universidad Nacional de Colombia, 2017). En la década de 1990, la implementación de sistemas de información académicos se expandió a otras instituciones educativas del país, como la Universidad del Valle, que implementó el Sistema de Información Estudiantil (SIE) en 1996 (Universidad del valle, 2017).

Posteriormente, en 2001, el Ministerio de Educación Nacional creó el Sistema Nacional de Información de la Educación Superior (SNIES), una plataforma que permite la gestión de la información académica de todas las instituciones de educación superior del país (Ministerio de Educación Nacional, 2019). El SNIES ha permitido la estandarización de la información académica en el país y ha facilitado el acceso a datos sobre programas académicos, matrículas y graduados, entre otros aspectos.

En los últimos años, diversas instituciones educativas en Colombia han implementado sistemas de información académicos más avanzados y sofisticados, que les permiten una gestión integral de la información. Un ejemplo de ello es el Sistema de Información Integral de la Universidad de Antioquia (SIIU), implementado en 2010, el cual permite la gestión de la información académica de estudiantes, docentes y programas académicos, así como la gestión financiera y la generación de reportes para la toma de decisiones (Universidad de Antioquia, 2012).

2.1.3 Antecedentes locales

En el ámbito local, el desarrollo de sistemas de información académicos en el departamento de Norte de Santander ha sido un proceso gradual y ha estado influenciado por los antecedentes nacionales en la materia. A continuación, se presentan algunos antecedentes relevantes en el desarrollo de los sistemas de información académicos en Norte de Santander.

En el año 2004, la Universidad de Pamplona implementó el Sistema de Información Académica (SIA), que permitió la gestión de información académica de estudiantes y docentes, así como la generación de reportes para la toma de decisiones (Universidad de Pamplona, 2004). En 2010, la Universidad Francisco de Paula Santander implementó el Sistema de Información de Estudiantes (SIE), que permite la gestión de información académica y financiera de los estudiantes, así como la generación de reportes para la toma de decisiones (Universidad Francisco de Paula Santander, 2018).

En 2016, la Universidad de Pamplona implementó una nueva versión de su sistema de información académica (SIA2), que incluyó nuevas funcionalidades como la gestión de solicitudes de certificados, constancias y diplomas, así como la generación de estadísticas y reportes más detallados (Universidad de pamplona, 2016).

En conclusión, los sistemas de información académicos han sido una herramienta importante en la gestión de la educación superior a nivel mundial. Su implementación ha permitido la automatización de tareas administrativas y la gestión eficiente de la información académica. En Colombia y en el departamento de Norte de Santander, la implementación de estos sistemas ha permitido mejorar la gestión académica y facilitar el acceso a la información

académica por parte de los estudiantes, contribuyendo así al mejoramiento de la calidad de la educación superior.

2.2. Marco Contextual

Figura 1.

Ubicación Geográfica de Ocaña Norte de Santander.



Fuente: Ciudad de Ocaña del departamento Norte De Santander (Alcaldía de Ocaña, 2023).

Según la información proporcionada por: (Alcaldía de Ocaña, 2023). Ocaña, situada en el departamento de Norte de Santander, es una ciudad que destaca por su riqueza histórica, cultural y paisajística. Conocida como la ciudad de los Caro, es un lugar atractivo para visitar gracias a su naturaleza, su gente, su historia y su clima excepcional. Durante la época colonial, Ocaña fue una

de las provincias más importantes del país y jugó un papel relevante en la vida política de la época.

Uno de sus mayores atractivos turísticos y religiosos es el Santuario del Agua de la Virgen de Torcoroma, aunque la gastronomía local también es irresistible, destacando productos únicos como el pan ocañero, las arepas, las barbatuscas y las cocotas. Además, el desfile de los genitores es un evento cultural reconocido como Patrimonio Inmaterial de la Nación.

Fundada el 14 de diciembre de 1570 por el capitán Francisco Fernández de Contreras, Ocaña fue una de las primeras ciudades en contar con cable aéreo. Durante la época de la Independencia, la ciudad adquirió una gran importancia en la culminación de la Campaña del Bajo Magdalena y la Campaña Admirable en 1813. En 1828, fue escogida como escenario de la "Gran Convención" y, en 1857, se erigió como municipio, convirtiéndose desde entonces en un polo de desarrollo (Alcaldía de Ocaña, 2023).

La actividad económica de la ciudad se centra en el comercio y la producción y comercialización de productos agrícolas, correspondiendo el 64.2% del total de la actividad al comercio, el 22.8% al sector servicios y el 1.8% a la industria.

Según las proyecciones del DANE, Ocaña cuenta con una población de aproximadamente cien mil habitantes, lo que la convierte en la segunda ciudad más grande del departamento de Norte de Santander. La ciudad se encuentra ubicada en la zona occidental del departamento sobre la cordillera Oriental, así como se puede evidenciar en la figura 1.

Figura 2.

Logotipo Universidad Francisco de Paula Santander Ocaña, UFPSO



Fuente: Logo de la Universidad Francisco De Paula Santander Ocaña, UFPSO (Universidad Francisco de Paula Santander Ocaña - Reseña Histórica, 2023).

La Universidad Francisco de Paula Santander Seccional Ocaña tiene un papel significativo en el municipio de Ocaña. En noviembre de 1973, se suscribió un contrato para llevar a cabo un estudio de factibilidad que evaluara la viabilidad de establecer un centro de educación superior en la región. El estudio se concluyó exitosamente, sugiriendo la creación de un programa de educación a nivel de tecnología con énfasis en ciencias sociales, matemáticas y física (Universidad Francisco de Paula Santander Ocaña - Reseña Histórica, 2023).

En diciembre de ese mismo año, el rector de la Universidad Francisco de Paula Santander, José Luis Acero Jordán, envió una copia del estudio al ICFES, quien conceptuó que el proyecto era recomendable. En consecuencia, mediante el Acuerdo No. 003 del 18 de julio de 1974, el Consejo Superior de la Universidad Francisco de Paula Santander creó la Universidad Francisco de Paula Santander Seccional Ocaña. La institución se estableció como la máxima expresión cultural y patrimonio de la región, y como una entidad de carácter oficial seccional,

con autonomía administrativa y patrimonio independiente, adscrito al Ministerio de Educación Nacional (Universidad Francisco de Paula Santander Ocaña - Reseña Histórica, 2023).

Según (Universidad Francisco de Paula Santander - Objetivos Institucionales, 2023), actualmente, la UFPSO tiene cinco objetivos principales: fortalecimiento de la cultura de la autoevaluación y aseguramiento de la calidad académica, gestión estudiantil pertinente y con calidad, desarrollo sostenible institucional, investigación y extensión con proyección global, y bienestar universitario y responsabilidad social.

El primer objetivo comprende el desarrollo docente para alcanzar la excelencia académica, el Sistema Interno de Aseguramiento de la calidad, la consolidación de las actividades de visibilidad, internacionalización y bilingüismo, y la virtualización e innovación de los programas académicos con el fin de establecer un campus virtual.

El segundo objetivo se enfoca en fortalecer los servicios académicos, gestionar el currículo para potenciar las competencias de los estudiantes, y promover la oferta académica mediante estrategias locales con enfoque nacional e internacional.

El tercer objetivo busca modernizar la universidad en términos de estructura, arquitectura de procesos y sistemas de información, así como también abogar por la sostenibilidad del campus universitario y gestionar el ciclo del talento humano como pilar del futuro de la institución.

El cuarto objetivo se enfoca en la consolidación de la producción científica, el fortalecimiento del proceso de extensión con pertinencia e impacto social, y el desarrollo de procesos de innovación, emprendimiento y transferencia tecnológica que redunden en beneficios para la institución y sus grupos de valor.

Finalmente, el quinto objetivo busca fortalecer los servicios y consolidar los procesos de bienestar que beneficien el clima y ambiente organizacional, así como también ejercer la responsabilidad social de la universidad con especial énfasis en la educación inclusiva.

En resumen, la Universidad Francisco de Paula Santander Seccional Ocaña cumple un papel importante en el municipio de Ocaña, con objetivos claros y definidos que apuntan al desarrollo académico, sostenibilidad, investigación y bienestar social de la región.

La Universidad Francisco de Paula Santander Ocaña es una institución pública de educación superior que se dedica a la formación de profesionales idóneos en diversas áreas del conocimiento, a través de estrategias pedagógicas innovadoras y el uso de las tecnologías. Como comunidad de aprendizaje y autoevaluación en mejoramiento continuo, su misión es contribuir al desarrollo nacional e internacional con pertinencia y responsabilidad social. Para el año 2025, la visión de la universidad es ser reconocida por la excelencia y eficiencia en el ejercicio de las funciones misionales con enfoque glocal, situando en valor las potencialidades de la comunidad universitaria y participando en los cambios del entorno mediante la transferencia del conocimiento y la innovación. Su objetivo es aportar al desarrollo sostenible de la sociedad y ser

una universidad acreditada de alta calidad (Universidad Francisco de Paula Santander- Mision & Visión, 2023).

Así mismo, cuenta con valores que caracterizan el ser y el actuar de las personas que son altamente deseables con atributos o cualidades, por cuanto posibilitan la construcción de una convivencia gratificante en el marco de la dignidad humana, es así como los valores son acuerdos de comportamiento que regulan la vida y definen la cultura de la Institución, y son promovidos a todos los miembros de la comunidad universitaria (N. . Consejo Superior Universitario UFPSO, 2019).

Figura 3.

Valores Institucionales UFPS Seccional Ocaña



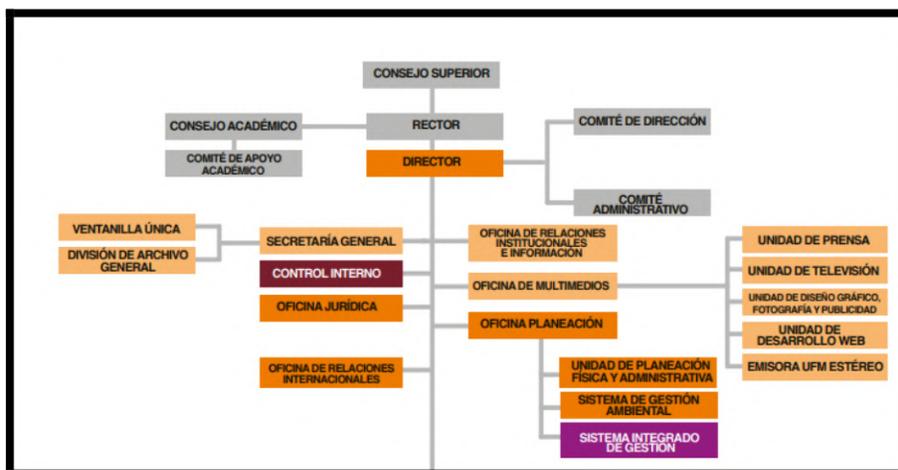
Fuente: Diagrama donde se exponen los valores de la UFPSO (N. . Consejo Superior Universitario UFPSO, 2019).

2.2.1. Organigrama

La Universidad Francisco de Paula Santander Ocaña cuenta con un organigrama y mapa de procesos gracias a la Resolución No. 0233 del 28 de octubre de 2008. Esta resolución adoptó el modelo de operación por procesos y el organigrama para la universidad, siguiendo la Ley 872 de 2003 que promueve el enfoque basado en procesos. Además, el Decreto 1537 del 2001 establece la obligatoriedad para las entidades de elaborar, adoptar y aplicar manuales que documenten y formalicen los procesos y procedimientos. El subsistema de Control Estratégico contiene la Estructura Organizacional, que configura los cargos, funciones, relaciones y niveles de responsabilidad y autoridad en la universidad. El mapa de procesos representa gráficamente los procesos estratégicos, misionales, de apoyo y de evaluación y sus interacciones. La adopción de estos elementos en el sistema de control interno de la universidad permitirá que sea efectivo, eficiente y eficaz en la ejecución de los procesos y actividades de acuerdo con su misión y función constitucional y legal (Universidad Francisco de Paula Santander Ocaña - Organigrama, 2023) (Universidad Francisco de Paula Santander-Ocaña, 2008) .

Figura 4.

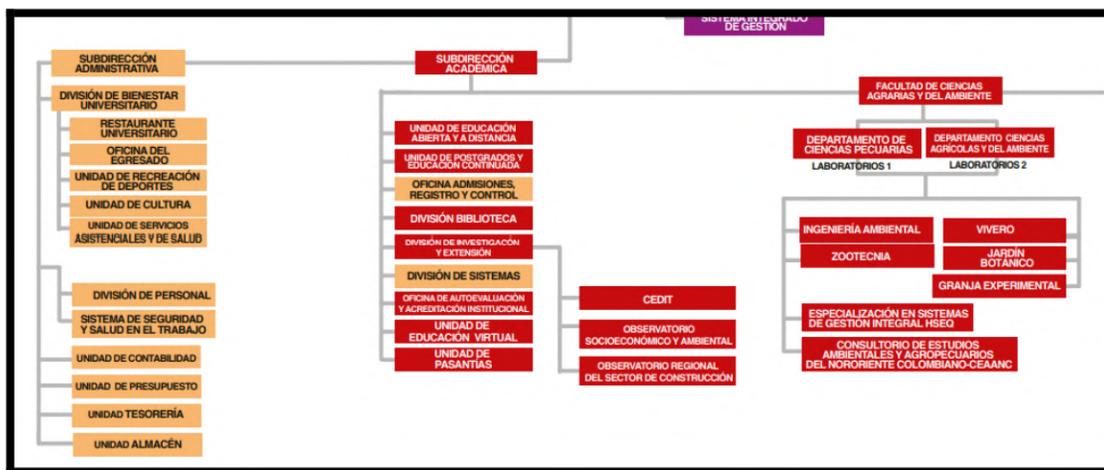
Organigrama Institucional Primera Parte.



Fuente: El gráfico muestra la estructura organización de la UFPSO (Universidad Francisco de Paula Santander Ocaña - Organigrama, 2023).

Figura 5.

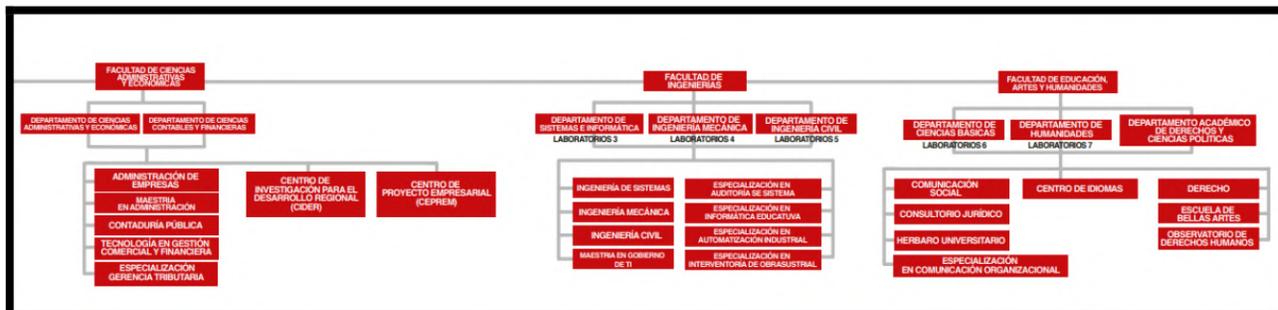
Organigrama Institucional Segunda Parte.



Fuente: El gráfico muestra la estructura organización de la UFPSO (Universidad Francisco de Paula Santander Ocaña - Organigrama, 2023).

Figura 6.

Organigrama Institucional Última Parte.



Fuente: El gráfico muestra la estructura organización de la UFPSO (Universidad Francisco de Paula Santander Ocaña - Organigrama, 2023).

Figura 7.

Guía Organigrama Institucional.

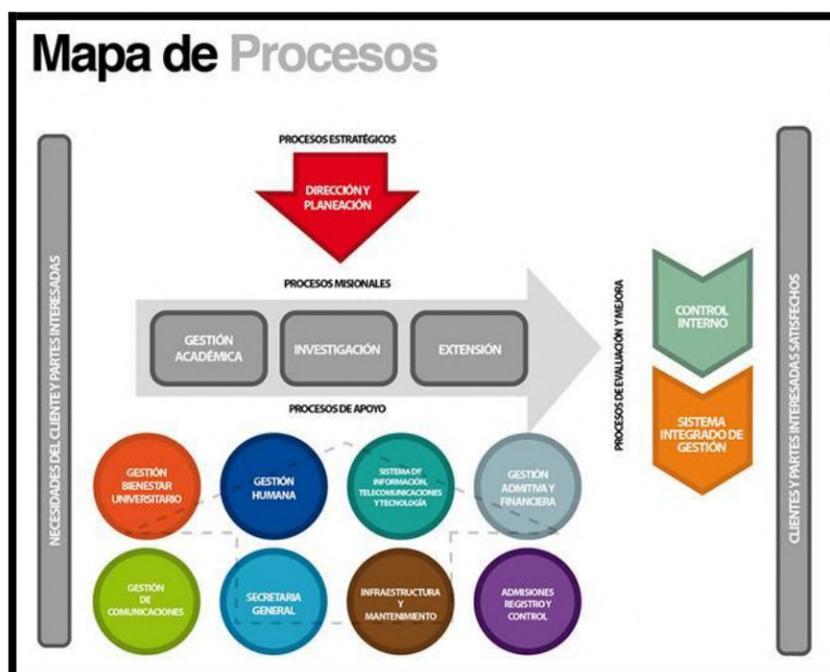


Fuente: Descripción de las unidades que se integran en la institución (Universidad Francisco de Paula Santander Ocaña - Mapa de Procesos, 2023).

2.2.2. Mapa de Procesos

Figura 8.

Mapa de Procesos Institucional.



Fuente: Mapa de Procesos Institucional, donde se encuentran los documentos actuales públicos del Sistema Integrado de Gestión. (Universidad Francisco de Paula Santander Ocaña - Mapa de Procesos, 2023).

2.2.3. Sistema de Información Académico (SIA)

La Universidad Francisco de Paula Santander Ocaña ha realizado un esfuerzo constante para brindar servicios de calidad a su comunidad universitaria, implementando una serie de proyectos de sistemas de información desde el año 2001. El primero de ellos fue el Sistema de Información Académico (SIA), seguido del Sistema de Información Bibliográfico (SIB) y el Sistema de Información Financiero (SIF), entre otros. Cada uno de estos sistemas fue desarrollado con el sistema manejador de base de datos relacional RDBMS ORACLE y

orientado a la web (Cliente/Servidor), lo que permite una eficiente gestión de la información y garantiza su integridad y seguridad.

En particular, el Sistema de Información Académico (SIA) es una aplicación diseñada para facilitar la administración de los diferentes procesos académicos que se llevan a cabo en la universidad. El sistema cuenta con módulos que cubren las funciones de consulta y gestión, permitiendo a los usuarios, según su rol (estudiante, docente, jefe de plan de estudios, secretaria, administrador), llevar a cabo diversas actividades relacionadas con su información personal, financiera y académica. Gracias a esta herramienta, se pueden realizar trámites de manera más eficiente y ágil, lo que se traduce en una mejor experiencia para los usuarios y una mayor eficacia en la gestión académica de la universidad (División-de-sistemas, 2023).

Figura 9.

Sistema de Información Académico (SIA) - Página Oficial



Fuente: Página de inicio del Sistema De Información Académico, SIA de la UFPSO (División-de-sistemas, 2023).

2.2.4. Según documentos de las carpetas del Genesis SIA

El génesis SIA v3 es un sistema de información académica en línea diseñado para ayudar a las instituciones educativas a gestionar eficientemente su información, permitiendo que docentes, estudiantes, administrativos y padres de familia puedan ser líderes en la era de la información mediante el uso de las TIC. Este sistema permite a las instituciones educativas registrar sus sedes, estudiantes, padres de familia, docentes, cursos, áreas, materias y logros, todo basado en el sistema de competencias, fortalezas y debilidades, según lo indica el decreto 1290 de abril de 2009. Además, mantiene un registro anual de la información y genera los informes requeridos para la secretaria de Educación, centralizando toda la información de la institución y permitiendo a los líderes educativos proyectar políticas académicas acordes al desarrollo del sector educativo (García, 2018).

Este sistema resuelve el problema de la falta de sistematización en los procesos de matrícula, evaluación académica y emisión de constancias y certificados, así como la falta de herramientas informáticas apropiadas para crear una cultura informática y promover una mayor interacción entre la comunidad académica. Una de las ventajas del génesis SIA v3 es que se adapta a cualquier institución educativa bajo el decreto 1290 de abril de 2009, y cuenta con un equipo de ingenieros disponibles para solucionar cualquier inquietud o nuevos requerimientos que las instituciones educativas puedan tener (García, 2018).

Figura 10.

Logotipo de la plataforma Génesis SIA.



Fuente: Logo del sistema de información académico Genesis SIA v3, software propietario de la UFPSO (García, 2018).

Figura 11.

Módulos de la plataforma Génesis SIA.



Fuente: Interface de administración del sistema Genesis SIA v3 usada por los colegios para su gestión académica (García, 2018).

2.3. Marco Conceptual

Para la conformación de este marco conceptual se tienen en cuenta las siguientes terminologías:

Sistema de información: Un sistema de información es un conjunto de elementos interconectados que trabajan juntos para procesar y administrar datos e información de manera efectiva. Los sistemas de información se utilizan en una variedad de campos, incluyendo la educación, la industria y el comercio (O'Brien, James; Marakas, 2012).

Sistema de información académica: Un sistema de información académica es un tipo de sistema de información diseñado específicamente para su uso en entornos educativos. Estos sistemas incluyen características como la gestión de matrículas, la administración de cursos, el seguimiento de calificaciones y la programación de horarios (Mabrouk, 2019).

.Java: Java es un lenguaje de programación de alto nivel ampliamente utilizado en la industria del software. Java es conocido por su portabilidad, seguridad y facilidad de uso, y se utiliza comúnmente en el desarrollo de aplicaciones empresariales y de escritorio. ~~creando aplicaciones para dispositivos móviles (O'Brien, 2022).~~

Framework: Un framework, o marco de trabajo, es un conjunto de bibliotecas y convenciones que se utilizan para facilitar el desarrollo de software. Los frameworks proporcionan una estructura básica para la aplicación y ayudan a los desarrolladores

a evitar la reescritura de código común y a mejorar la eficiencia en el desarrollo. (Gamma, Helm

Johnson, & Vlissides, 1995).

MVC: MVC, o Modelo-Vista-Controlador, es un patrón de diseño de software que se

utiliza para separar la lógica de presentación de una aplicación de la lógica de negocios. MVC

divide una aplicación en tres componentes: el modelo que maneja la lógica de negocios y los

datos; la vista, que maneja la presentación

de la información al usuario; y el controlador, que

maneja la interacción del usuario y la lógica

de navegación (Fowler, 2002).

maneja la interacción del usuario y la lógica

framework de Java que se utiliza para simplificar el

Spring Boot: Spring Boot es un framework

Spring Boot proporciona un conjunto de herramientas y

desarrollo de aplicaciones empresariales. Spring

otras tecnologías de Java, como Spring Framework y

bibliotecas que se integran fácilmente con otras

para crear aplicaciones más rápidamente y con menos

Hibernate, lo que permite a los desarrolladores

código (Spring Boot, 2022).

de matrículas es un proceso importante dentro de los

Gestión de matrículas: La gestión de

permite a los estudiantes inscribirse en los cursos que

sistemas de información académica, ya que muchos

verificación de la elegibilidad del estudiante para tomar el

desean tomar. Este proceso incluye la verificación

de horarios (Mabrouk, 2019).

curso, la selección de los cursos y la programación

de horarios es un proceso crítico en los sistemas de

Gestión de horarios: La gestión de

programación de clases y actividades. Los sistemas de

información académica, ya que implica la programación

de estudiantes y profesores a programar y visualizar

información académica pueden ayudar a los

horarios de manera eficiente, evitando conflictos de horarios y problemas de programación (Mabrouk, 2019).

Seguridad de la información: La seguridad de la información es un aspecto importante a considerar en los sistemas de información académica, ya que contienen información confidencial sobre estudiantes, profesores y empleados. Es esencial garantizar que se implementen medidas de seguridad adecuadas para proteger la privacidad y la integridad de la información (Mabrouk, 2019).

2.4. Marco Teórico

2.4.1. Teorías de Sistemas de Información Académica

Los sistemas de información académicos son herramientas tecnológicas diseñadas para ayudar a la gestión y organización de las actividades académicas en una institución educativa. Los sistemas de información académicos permiten la automatización de procesos administrativos y académicos, como el registro de estudiantes, la gestión de notas y calificaciones, la programación de horarios de clases y la gestión de recursos y materiales.

La teoría de la gestión de la información es un enfoque teórico que proporciona una base sólida para el diseño y desarrollo de sistemas de información académicos. Según esta teoría, la gestión de la información implica la creación, el almacenamiento, el procesamiento y la distribución de la información en una organización. Los sistemas de información académicos son

una aplicación práctica de la teoría de la gestión de la información en el ámbito educativo (Israel Núñez, 2018).

Otra teoría relevante en el diseño y desarrollo de sistemas de información académicos es la *teoría de sistemas*. Según esta teoría, un sistema es un conjunto de elementos interdependientes que trabajan juntos para lograr un objetivo común. Los sistemas de información académicos pueden ser considerados como sistemas complejos que involucran múltiples componentes, como la base de datos, el software de gestión, el hardware y los usuarios finales. Una comprensión clara de la teoría de sistemas es esencial para el diseño y desarrollo efectivo de sistemas de información académicos (Burch & Grudnitski, 1989).

La *teoría de la difusión de innovaciones de Rogers*, también puede ser aplicada al estudio de la adopción de los SIA. Esta teoría propone que la adopción de una innovación se da a través de cinco etapas: conocimiento, persuasión, decisión, implementación y confirmación. Según esta teoría, la velocidad y grado de adopción de una innovación depende de diversos factores, entre los que se incluyen la ventaja relativa de la innovación en comparación con la tecnología previa, la compatibilidad con las necesidades del usuario, la complejidad percibida de la innovación y la observabilidad de sus beneficios (Rogers, 2010).

Uno de los modelos teóricos más utilizados en el estudio de la adopción de los SIA es la *Teoría de Aceptación de la Tecnología (TAM)*. Esta teoría propone que la intención de uso de una tecnología está determinada por la percepción de la utilidad y la facilidad de uso de la misma

(Davis, 1989). Estudios han encontrado que la TAM es un modelo válido y confiable para explicar la adopción de los SIA en diferentes contextos (Sánchez, Hueros, & García, 2017).

Otro modelo teórico que ha sido aplicado al estudio de los SIA es la *Teoría del Comportamiento Planificado (TCP)*. Esta teoría propone que la intención de comportamiento está determinada por la actitud hacia el comportamiento, la norma subjetiva y el control percibido sobre el comportamiento (Ajzen, 2019). Estudios han encontrado que la TCP es una teoría válida para explicar la intención de uso y adopción de los SIA en diferentes contextos.

2.4.2. Teorías sobre el desarrollo en PHP

PHP es un lenguaje de programación de código abierto ampliamente utilizado en el desarrollo de aplicaciones web. El desarrollo en PHP implica la creación de aplicaciones web dinámicas y interactivas utilizando el lenguaje de programación PHP y una variedad de tecnologías web, como HTML, CSS y JavaScript. Las teorías del desarrollo de software proporcionan una base sólida para el diseño y desarrollo de aplicaciones web en PHP.

La teoría de la arquitectura de software. La arquitectura de software se refiere a la estructura y organización de un sistema de software. En el desarrollo de aplicaciones web en PHP, la teoría de la arquitectura de software es esencial para la creación de sistemas escalables, eficientes y fáciles de mantener. Los patrones de diseño, como el patrón Modelo Vista

ectura de software en el

Controlador (MVC), son una aplicación práctica de la teoría de la arquit

desarrollo de aplicaciones web en PHP (Bakken, 2015).

Así mismo, una de las teorías más utilizadas en el desarrollo en PHP es la *Arquitectura Modelo-Vista-Controlador (MVC)*. Esta teoría propone separar la aplicación en tres componentes: el Modelo, que maneja la lógica de la aplicación y la interacción con la base de datos; la Vista, que se encarga de la presentación visual de la aplicación; y el Controlador, que maneja la interacción del usuario con la aplicación y conecta la Vista con el Modelo (Fowler, 2002). La utilización del patrón MVC en el desarrollo en PHP ha sido ampliamente recomendada para lograr una separación clara de las responsabilidades y un código más organizado y mantenible (Zend Technologies, 2016).

Otra teoría importante en el desarrollo en PHP es la *Programación Orientada a Objetos*

(*POO*). La POO es una teoría que propone la organización del código en objetos que encapsulan datos y funciones relacionadas. En PHP, la POO se utiliza para crear clases que representan

objetos con atributos y métodos (Bakken, 2015). La POO en PHP permite una mayor extensión, modularidad y reutilización del código, así como una mayor seguridad y capacidad de

2.4.3. Teorías sobre el desarrollo a través de patrones de diseño

Los patrones de diseño son soluciones comunes a problemas de diseño de software que han sido probadas y documentadas. La *teoría de los patrones de diseño* propone que los patrones comunes pueden mejorar la calidad del software al permitir que los desarrolladores eviten errores y utilicen soluciones probadas (Gamma, Helm, Johnson, & Vlissides, 1995). Además, la aplicación de patrones de diseño puede mejorar la legibilidad, la reutilización y la mantenibilidad del código (Khalib, Ghayyur, & Igbal, 2019). Existen diversas categorías de patrones de

tales como patrones de creación, patrones estructurales y patrones de comportamiento (Gamma et al., 1995).

Los patrones de creación se enfocan en la creación de objetos y se dividen en dos subcategorías: los patrones de creación de clase y los patrones de creación de objeto. Los patrones estructurales se enfocan en la composición de objetos y la formación de estructuras más complejas, mientras que los patrones de comportamiento se enfocan en la interacción entre objetos y la distribución de responsabilidades (Gamma, Helm, Johnson, & Vlissides, 1995).

2.5. Marco Legal

La información Legal relacionada con el desarrollo del proyecto y el objeto de estudio:

2.5.1. Ley 1581 de 2012 - Protección de datos personales

Esta ley establece las reglas para el tratamiento de datos personales en Colombia. En el caso de los sistemas de información académica, se deben tomar medidas para garantizar la privacidad y seguridad de los datos personales de los estudiantes y docentes (Congreso de la República de Colombia, 2023).

2.5.2. Ley 603 de 2000 - Ley de software

Esta ley establece las reglas para el uso y protección del software en Colombia. En el desarrollo de software en PHP, se deben respetar los derechos de autor y propiedad intelectual de terceros (Departamento Administrativo de la Función Pública, 2000).

2.5.3. *NTC-ISO/IEC 12207*

Esta norma establece los estándares para el ciclo de vida del software. En el desarrollo de software en PHP, se pueden seguir estas normas para mejorar la calidad y mantenibilidad del software (Cornejo, 2015).

2.5.4. *Resolución 1409 de 2012*

Esta resolución del Ministerio de Educación Nacional establece las normas para la gestión de la información académica en las instituciones de educación superior en Colombia. La resolución establece las normas para la recolección, almacenamiento, procesamiento y divulgación de la información académica de los estudiantes (Ministerio de trabajo, 2012).

2.5.5. *Política de Tratamiento de Datos Personales de la UFPS*

Esta política establece las normas para el tratamiento de los datos personales de los estudiantes, docentes y personal administrativo de la universidad. La política se basa en la Ley 1581 de 2012 y se aplica en todas las sedes de la universidad, incluyendo la sede de Norte de Santander (Tolosa Martínez, 2011).

2.5.6. Reglamento Estudiantil de la Universidad Francisco de Paula Santander

Este reglamento establece las normas para la gestión académica de los estudiantes de la universidad. El reglamento establece las normas para la recolección, almacenamiento, procesamiento y divulgación de la información académica de los estudiantes, incluyendo los sistemas de información académica (C. Consejo Superior Universitario UFPSO, 2019).

Resulta relevante destacar que se llevó a cabo una exhaustiva investigación acerca de las normativas y regulaciones vigentes en otros territorios en lo referente a los sistemas de información académica, el desarrollo en PHP y los patrones de diseño. Dicha indagación se realizó con el objetivo de ampliar el conocimiento sobre estas temáticas.

2.5.7. Para el caso de los Sistemas de información académica, se encuentran las siguientes leyes:

En relación con los sistemas de información académica, existen leyes y normativas que rigen la protección de datos personales y la privacidad de los estudiantes. En muchos países, se requiere el consentimiento explícito de los estudiantes y se establecen medidas de seguridad para garantizar la confidencialidad y la integridad de los datos académicos (Regulation European Union, 2016).

A nivel Internacional:

Ley de Protección de Datos Personales: Esta ley regula el tratamiento de los datos personales y su protección. En el caso de los sistemas de información académica, se deben tomar medidas para garantizar la privacidad y seguridad de los datos personales de los estudiantes y docentes (Humala Tasso, Ollanta; Jiménez Mayor, Juan; Pedraza Sierra, 2013).

Ley de Protección de Datos de la Unión Europea (GDPR): Esta ley establece una serie de regulaciones en cuanto al procesamiento de datos personales, incluyendo los datos académicos de los estudiantes. La GDPR tiene implicaciones para cualquier organización que procese datos personales de ciudadanos de la UE, incluyendo las universidades y las escuelas (Regulation European Union, 2016).

Ley de Firma Electrónica: Esta ley regula la firma electrónica y su validez legal. En el contexto de los sistemas de información académica, se pueden utilizar firmas electrónicas para validar documentos y trámites (Congreso de la República del Perú, 2008).

Ley de Software Libre: Esta ley promueve el uso y desarrollo de software libre en la administración pública. En el desarrollo de software en PHP, se puede considerar el uso de software libre para reducir costos y fomentar la innovación (Congreso de Chile, 2002).

2.5.8. Para el caso del desarrollo de software en PHP, se encuentran las siguientes leyes:

En cuanto al desarrollo de software en PHP, se deben seguir las leyes y regulaciones en materia de propiedad intelectual y derechos de autor. Es importante asegurarse de que el

software desarrollado no infrinja los derechos de propiedad intelectual de terceros y de que se obtengan las licencias necesarias para utilizar bibliotecas y frameworks de terceros en el desarrollo de aplicaciones (United States Copyright Office, 2023).

A nivel internacional:

Ley de Derechos de Autor del Milenio Digital de Estados Unidos (DMCA): Esta ley establece una serie de regulaciones para proteger los derechos de autor en línea. La ley establece que el software, incluyendo el software en PHP, está protegido por derechos de autor y que el uso no autorizado del software está sujeto a sanciones (Gamma, Helm, Johnson, & Vlissides, 1995).

2.5.9. Para el caso de los Patrones de diseño, se encuentran las siguientes leyes:

A nivel Internacional:

Normativa de Patrones de Diseño: Esta normativa establece los estándares para el desarrollo de patrones de diseño en software. En el desarrollo de software en PHP, se pueden seguir estas normativas para mejorar la calidad y mantenibilidad del software (IEEE Standard, 2009).

3. Diseño Metodológico

3.1. Tipo de investigación

Para el proyecto, se utilizará una investigación de enfoque cualitativo con alcance descriptivo. Según (Hernandez Sampiere; Pilar Baptista; Fernández Collado, 2014), el enfoque cualitativo se centra en comprender y describir los fenómenos sociales desde la perspectiva de los participantes y su contexto, en lugar de medir y cuantificar las variables de manera objetiva. La investigación cualitativa implica la utilización de métodos como la observación, la entrevista y el análisis de documentos para recopilar datos. Estos datos se analizan y se interpretan para identificar patrones, temas y relaciones que ayuden a comprender mejor el fenómeno social estudiado. En este sentido, el enfoque cualitativo permite una exploración más profunda y detallada de los fenómenos sociales, y una comprensión más completa de la perspectiva de los participantes.

Por otro lado, el alcance descriptivo se refiere a la descripción detallada y completa de los aspectos específicos de un fenómeno social. En la investigación con alcance descriptivo, el objetivo principal es describir las características y particularidades de un fenómeno, en lugar de explicar o predecir su comportamiento.

En palabras de (Hernandez Sampiere; Pilar Baptista; Fernández Collado, 2014), "La investigación cualitativa con alcance descriptivo se enfoca en la descripción detallada de las características, cualidades y particularidades de un fenómeno social, y utiliza la observación, la entrevista y el análisis de documentos para recopilar datos que permitan comprender el fenómeno desde la perspectiva de los participantes".

En resumen, para este proyecto se utilizarán métodos de observación, entrevista y análisis de documentos para recopilar datos, y el objetivo principal será describir detalladamente las características y particularidades de un fenómeno social que en este caso son los sistemas de información académica, desde la perspectiva de los participantes.

3.2. Población y muestra

3.2.1. Población

La población objeto de estudio son los once (11) colegios integrados a la plataforma del Génesis SIA. Los colegios son:

Colegio Fray José María Arévalo, La Playa
Institución Educativa Aguas Claras, Corregimiento Aguas Claras, Ocaña
Institución Educativa Carlos Julio Torrado Peñaranda, Ábrego
Institución Educativa Colegio Agustina Ferro, Ocaña
Institución Educativa Colegio Santa Barbara, Ábrego
Institución Educativa Escuela Normal Superior De Corozal
Instituto Técnico Alfonso López, Ocaña
Institución Educativa Colegio La Salle, Ocaña
Institución Educativa Francisco Fernández de Contreras, Ocaña
Institución Educativa Colegio Integrado Gilberto Claro Lozano
Institución Educativa José Mejía Uribe, La Gloria, Cesar

3.2.2. Muestra

Teniendo en cuenta la población objeto de estudio tomada, la muestra está conformada por cuatro colegios que serán seleccionadas en el transcurso del proyecto. Esto corresponde al 36,36% de la población seleccionada.

Para el desarrollo del proyecto, se utilizarán técnicas de recolección de información que incluyen fuentes primarias y secundarias. Las fuentes primarias incluyen entrevistas con los

desarrolladores de la plataforma, reuniones con los directivos del colegio y encuestas a los usuarios del sistema Génesis SIA V.3.0. Estas técnicas permitirán un análisis detallado del contexto y una comprensión precisa de los requisitos del sistema.

Además, se utilizarán fuentes secundarias, como la documentación existente del sistema de información Génesis SIA en versiones anteriores y la actual (v.3.0), para identificar áreas de mejora. También se investigarán tecnologías y arquitecturas utilizadas en el desarrollo de sistemas de información académico para seleccionar las mejores tecnologías que satisfagan las necesidades del proyecto.

4. Resultados

4.1. Primer Objetivo

Analizar el contexto, a través del levantamiento de información clave y con ello tener un entendimiento específico de los requisitos del sistema.

1. Realizar entrevistas programadas con los desarrolladores del Génesis SIA v3.0.
2. Organizar reuniones con los directivos de los colegios seleccionados.
3. Aplicar encuestas a los usuarios del sistema Genesis SIA v3.0.

Para cumplir con el primer objetivo de analizar el contexto, a través del levantamiento de información clave y con ello tener un entendimiento específico de los requisitos del sistema se realizaron los siguientes pasos:

Se contactó a los desarrolladores del Génesis SIA v3.0 a los cuales se le hicieron preguntas claves sobre cómo fue desarrollado el Genesis SIA en ese momento con el propósito de obtener un contexto técnico de cómo fue desarrollado el software.

También se contó con la opinión de los directivos de los colegios con el propósito de obtener una perspectiva de la funcionalidad y adaptabilidad que tiene el software en cada colegio, además tomar recomendaciones de mejora que ellos consideren de utilidad para la administración y uso del software en sus instituciones.

Por último, se contó con una encuesta de satisfacción de los usuarios del Genesis SIA para análisis de la aceptación y conformidad de los usuarios finales con el software y así detectar falencias y oportunidades de mejora a tener en cuenta en la nueva versión de Genesis SIA.

Esta serie de pasos nos permitieron obtener un contexto lo más completo posible en el cual obtenemos información de todos los agentes que están involucrados en el software Genesis SIA, toda esta información al ser analizada nos permitió tener una visión en la cual podemos definir las falencias y debilidades que tiene el software. Gracias a esta información se encontraron las opciones de mejora que serán aplicadas en la nueva versión para solventar estas desventajas y así construir un software más robusto y competitivo en el mercado.

4.1.1. Entrevista a los desarrolladores del Genesis SIA v3.0

Al momento de diseñar sistemas informáticos es fundamental tener un conocimiento técnico y experiencia de los expertos involucrados en el caso de estudio que estas abordando teniendo en cuenta esto se entrevistó a dos desarrolladores que fueron parte del planeación, ejecución, despliegue, soporte y mantenimiento del Genesis SIA v3.0.

Para la realización de la entrevista fue usado un formato el cual cuenta con preguntas clave que permiten obtener información clave del cómo fue construido el sistema de información, las técnicas y tecnologías que utilizaron en ese momento, Luego de obtener toda la información relevante se obtuvieron los siguientes resultados:

El Genesis SIA fue un producto que salió a producción en el año 2014 por el proyecto de interconectividad de la UFPSO en el cual participaron grupo de ingenieros de UFPSO los cuales desarrollaron el sistema de información académica orientado a la web, debido a esto se utilizaron tecnologías como HTML, CSS y JS para el Frontend, PHP para el Backend y PostgreSQL para la gestión de base de datos, en la parte de arquitectura de software al momento de planeación de sistema no se estableció ningún patrón de arquitectura lo cual generó que el software sea poco escalable en tiempo dificultando el soporte y mantenimiento.

El Genesis SIA v3.0 al no contar con una arquitectura definida se creó un monolito de código el cual presenta dificultades al momento de agregar o modificar nuevas funcionalidades ya que no cumple con los principios SOLID el software es poco escalable limitando la aplicación de funcionalidades y alcance que puede tener el sistema.

Esto genera que el soporte y mantenimiento sea tedioso ya que al momento de resolver un incidente o nuevo requerimiento debes adaptarlo al código antiguo dificultando la implementación ya que no cuenta con un marco de trabajo que permita la integración de nuevas soluciones al código, esto debido al alto grado de acoplamiento con el que cuenta el software lo cual genera que el desarrollador no tenga noción de cual es el alcance real de ese código que desean modificar generando que implementen nuevas funciones y dejando código obsoleto dentro del software.

Complemento de lo anteriormente dicho es que el software no cuenta con sistema de versionado ni de repositorio de código lo cual genera que toda modificación sea realizada en

producción generando la caída de los servicios del Genesis SIA durante el tiempo que sea necesario para solucionar los problemas presentados generando intermitencia en los servicios prestados a las instituciones educativas.

~~Otra de las problemáticas planteadas es que el software se personaliza según las reglas~~

para cada institución

de diseño y arquitectura en la

la plataforma no puedan ser

contactar con la oficina de

incidentes directamente en

s.

instituciones Colegio Fray José

s expone la idea de desarrollar

es educativas estos son los

plataforma que necesita ser

que ofrecen una mejora

en y administración de todos sus

impuestas por cada institución generando una versión modificada

educativa, esto combinado con la falta de aplicación de patrones de

plataforma hace que muchos de los incidentes que se presentan en

resueltos de forma administrativa generando la necesidad de contar

interconectividad para que los ingenieros a cargo resuelvan estos in

código.

4.1.2. Reuniones con los directivos de los colegios seleccionados.

Durante la investigación se habló con los Rectores de las in

María Arévalo y Escuela Normal Superior De Corozal donde se le

un nuevo génesis SIA para suplir las necesidades de las instituciones

resultados.

Los rectores concuerdan que el Genesis SIA v3.0 es una pl

actualizada ya que en el mercado hay alternativas más modernas q

sustancial en los procesos de las instituciones facilitando la gestión

procesos.

Es necesario que la plataforma ofrezca diferentes sistemas de evaluación lo cual permita que la institución educativa tenga la flexibilidad de elegir cual se adapta mejor los procesos que se llevan a cabo en la institución y estos permita la mejora continua de la calidad de la educación ofrecida por el plantel educativo.

También se hizo referencia a mejora del diseño del software nos comentaron los rectores que les gustaría un sitio moderno y con el énfasis en la usabilidad del sistema que permita mayor independencia de las instituciones al soporte brindado por el proyecto de interconectividad, esto enfocado que los usuarios del sistema tengan la capacidad de resolver problemas de manera administrativa dentro del software sin la intervención de los ingenieros del proyecto.

Además de expresar la necesidad de un nuevo sistema Genesis SIA, los usuarios también destacan la importancia de ofrecer capacitaciones periódicas al personal de la institución. Estas capacitaciones son consideradas esenciales para garantizar un uso efectivo y maximizar el potencial del sistema. Al proporcionar una formación regular, se busca mejorar la comprensión y el dominio de las herramientas y funcionalidades del sistema, lo que a su vez contribuirá a una mejor implementación y aprovechamiento de todas sus características.

Por último, se trató el tema de nuevas funcionalidades que les gustaría ver en sistema a lo cual nos expresan que les gustaría que los estudiantes tuvieran acceso a sus módulos de forma móvil, que la navegabilidad por la aplicación sea más sencilla, que la plataforma cuente con

tutoriales guiados para los diferentes módulos y la capacidad de personalizar el sistema a la forma con trabaja cada institución de forma individual.

4.1.3. Encuestas a los usuarios del sistema Genesis SIA v3.0

En esta sección, se presentan los resultados de una encuesta diseñada para medir el nivel de satisfacción de los usuarios del Genesis SIA, un sistema de información académico utilizado en instituciones educativas. El objetivo principal de la encuesta fue evaluar la satisfacción de los usuarios con respecto a diversos aspectos del sistema, incluyendo la facilidad de uso, la calidad de los datos, la velocidad de procesamiento y la eficacia de las funciones y características del sistema.

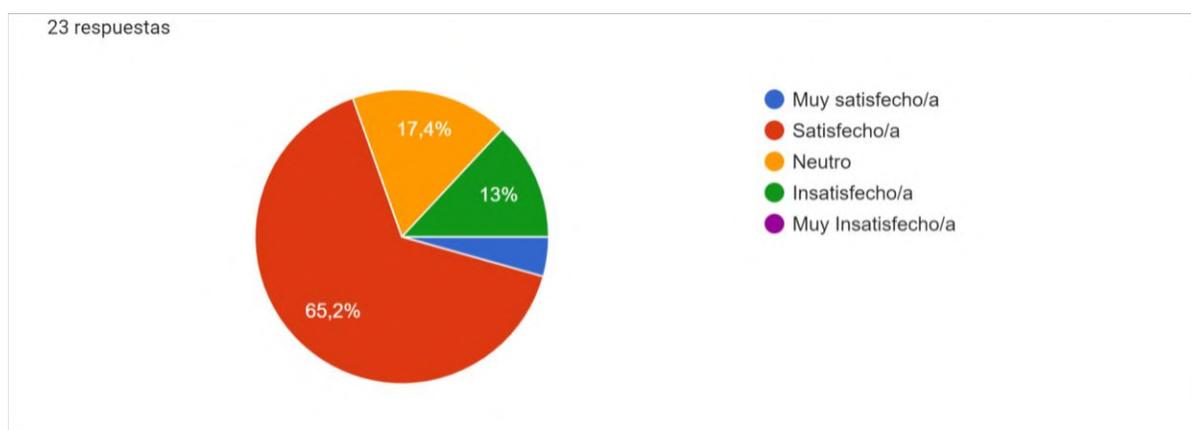
La encuesta fue enviada a una muestra de usuarios del Genesis SIA que pertenecen a diferentes instituciones educativas, y se recibieron respuestas completas de un número significativo de usuarios. En esta sección, se presentan los resultados detallados de la encuesta, incluyendo el análisis estadístico de las respuestas y la presentación de gráficos y tablas para facilitar la visualización de los resultados.

La información obtenida a través de esta encuesta será de gran importancia para las instituciones educativas que utilizan el sistema, ya que permitirá identificar los puntos fuertes y débiles del Genesis SIA y mejorar la calidad del servicio ofrecido a los usuarios. Además, los resultados de la encuesta pueden ser útiles para otras instituciones educativas que deseen evaluar la satisfacción de los usuarios de sus propios sistemas de información académico.

En total se recibieron 23 repuestas de las diferentes instituciones en donde fueron respondida por los diferentes usuarios del Genesis SIA v3.0 a continuación los resultados de cada pregunta:

Figura 12.

Resultados de la pregunta ¿Qué tan satisfecho/a está con el uso que le da al Genesis SIA v3?

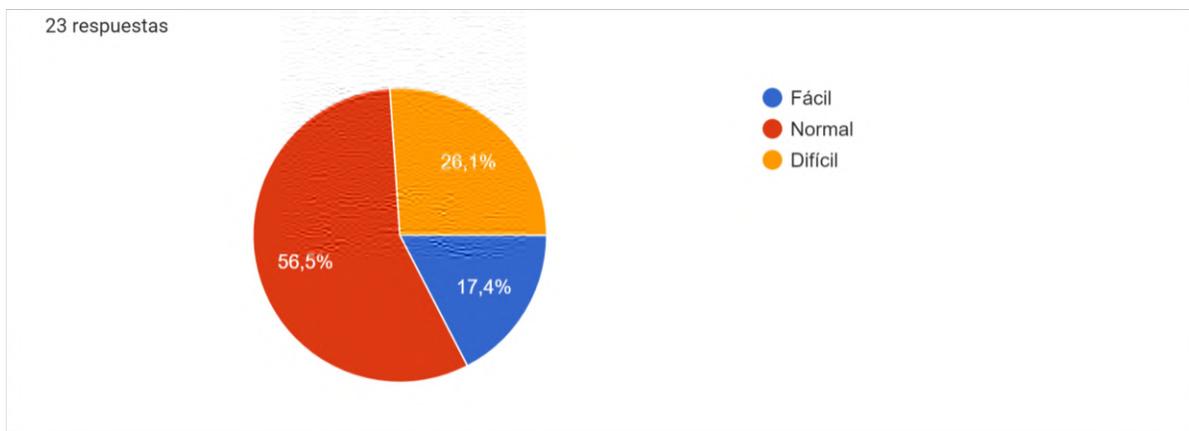


Fuente: El Grafico muestra los resultados obtenidos en la primera pregunta de la encuesta.

La anterior gráfica muestra la distribución de las respuestas de los usuarios del Genesis SIA en cuanto a su nivel de satisfacción con el uso que ellos le dan al sistema. El 65,2% de los usuarios se encuentra en el grupo de satisfecho, mientras que un 4,3% se encuentra en el grupo de muy satisfecho. Por otro lado, el 13% de los usuarios reporta estar insatisfecho con el sistema, y un 17,4% se encuentra en el grupo neutro, es decir, no expresó una opinión positiva o negativa sobre su experiencia con el Genesis SIA.

Figura 13.

Resultados de la pregunta ¿Cómo calificaría la facilidad de uso del Genesis SIA en su jornada laboral?

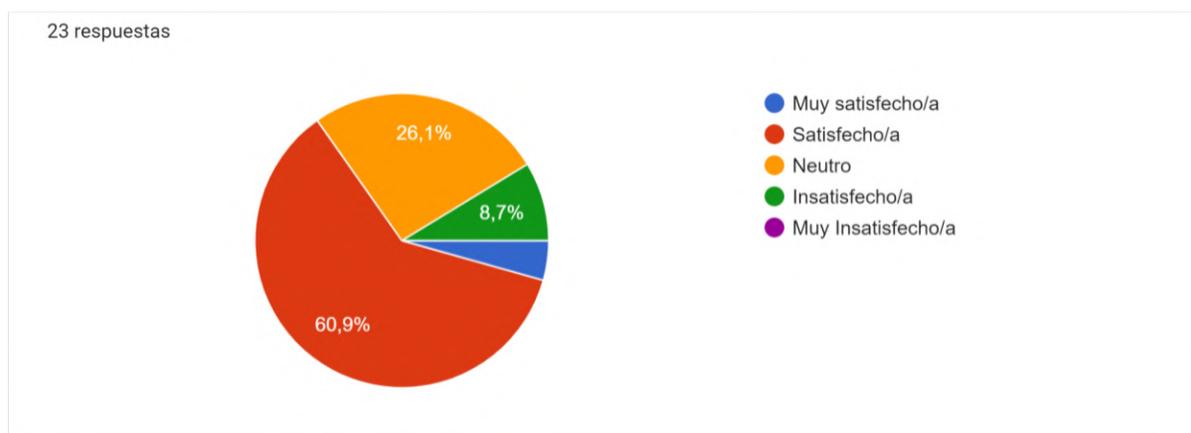


Fuente: El Grafico muestra los resultados obtenidos en la segunda pregunta de la encuesta.

Los resultados de la encuesta muestran que el nivel de facilidad de uso del software Genesis SIA es percibido de manera heterogénea por los usuarios. El 56,5% de los usuarios lo considera normal, lo que indica que no es especialmente fácil o difícil de utilizar. Sin embargo, un 26,1% de los usuarios lo considera difícil y solo el 17,4% lo considera fácil.

Figura 14.

Resultados de la pregunta ¿Cómo calificaría el soporte que ofrece el Genesis SIA al momento de resolver solicitudes que usted ha realizado?



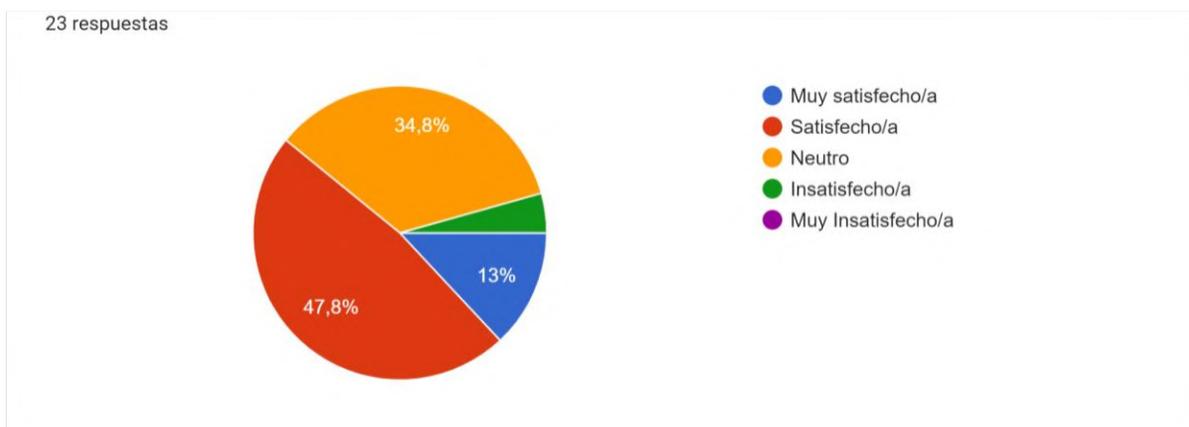
Fuente: El Gráfico muestra los resultados obtenidos en la tercera pregunta de la encuesta.

La gráfica presenta la evaluación del soporte brindado por el Genesis SIA al momento de resolver las solicitudes de los usuarios. Según los resultados obtenidos, el 60,9% de los usuarios se encuentra satisfecho con el soporte recibido, mientras que un 4,3% está muy satisfecho. Por otro lado, el 26,1% de los usuarios se encuentra en la categoría neutro, lo que indica que no expresaron una opinión clara sobre la calidad del soporte. Sin embargo, el 8,7% de los usuarios se muestra insatisfecho con el soporte brindado.

Figura 15.

Resultados de la pregunta ¿Qué tan satisfecho/a está con la confiabilidad del sistema Genesis

SIA v3?

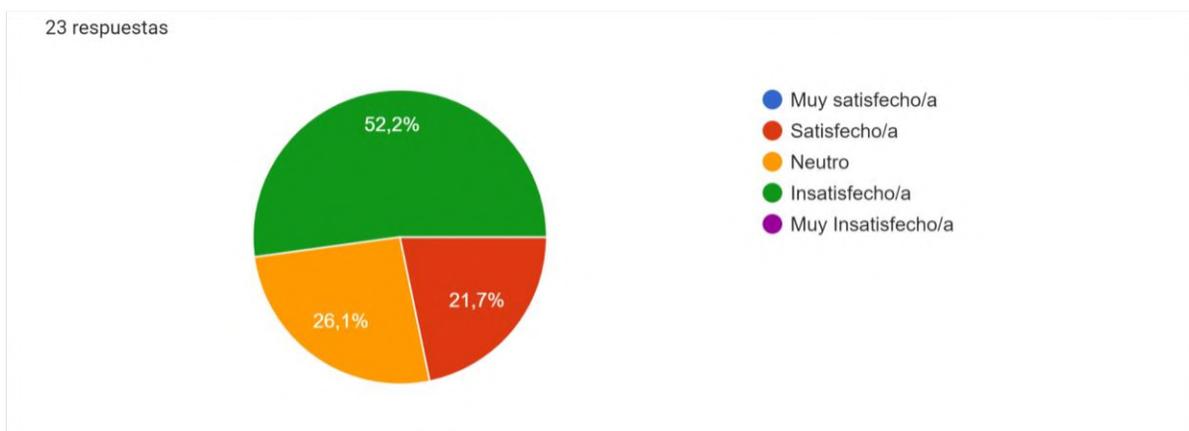


Fuente: El Gráfico muestra los resultados obtenidos en la cuarta pregunta de la encuesta.

La gráfica refleja la evaluación de la confiabilidad del sistema Genesis SIA por parte de los usuarios. Según los resultados, un 47,8% de los usuarios se encuentra satisfecho con la confiabilidad del sistema, mientras que un 13% muestra un alto nivel de satisfacción, calificándolo como muy satisfactorio. Por otro lado, un 34,8% de los usuarios se posiciona en la categoría neutro, indicando una falta de opinión clara sobre la confiabilidad del sistema. Sin embargo, un 4,3% de los usuarios se muestra insatisfecho con la confiabilidad del Genesis SIA.

Figura 16.

Resultados de la pregunta *¿Qué tan satisfecho/a está con la apariencia del sistema Genesis SIA v3?*

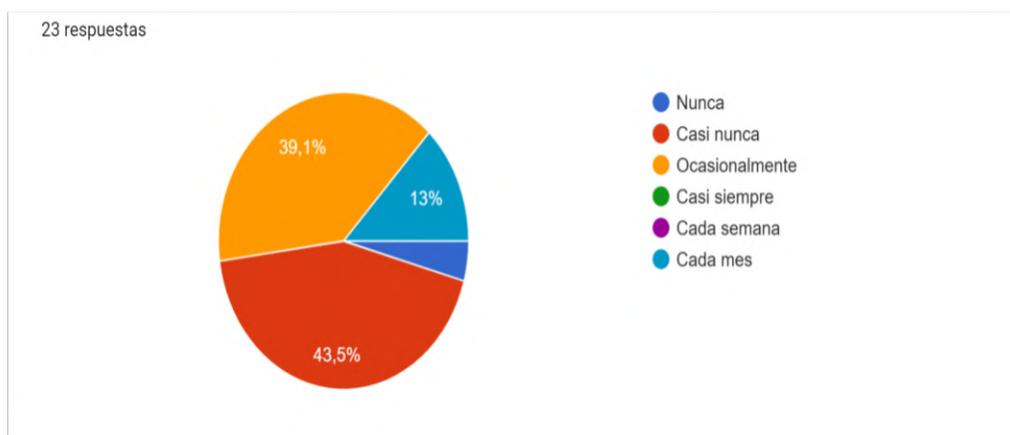


Fuente: El Gráfico muestra los resultados obtenidos en la quinta pregunta de la encuesta.

La gráfica presenta la evaluación de la apariencia y las interfaces gráficas del sistema Genesis SIA, según la opinión de los usuarios. Los resultados revelan que un 21,7% de los usuarios se encuentra satisfecho con la apariencia y las interfaces gráficas del sistema, mientras que un 26,1% se ubica en la categoría neutro, lo que indica una falta de opinión clara al respecto. Sin embargo, la mayoría de los usuarios, un 52,2%, se muestra insatisfecho con la apariencia y las interfaces gráficas del Genesis SIA.

Figura 17.

Resultados de la pregunta *¿Con qué frecuencia encuentra errores en el sistema Genesis SIA v3?*

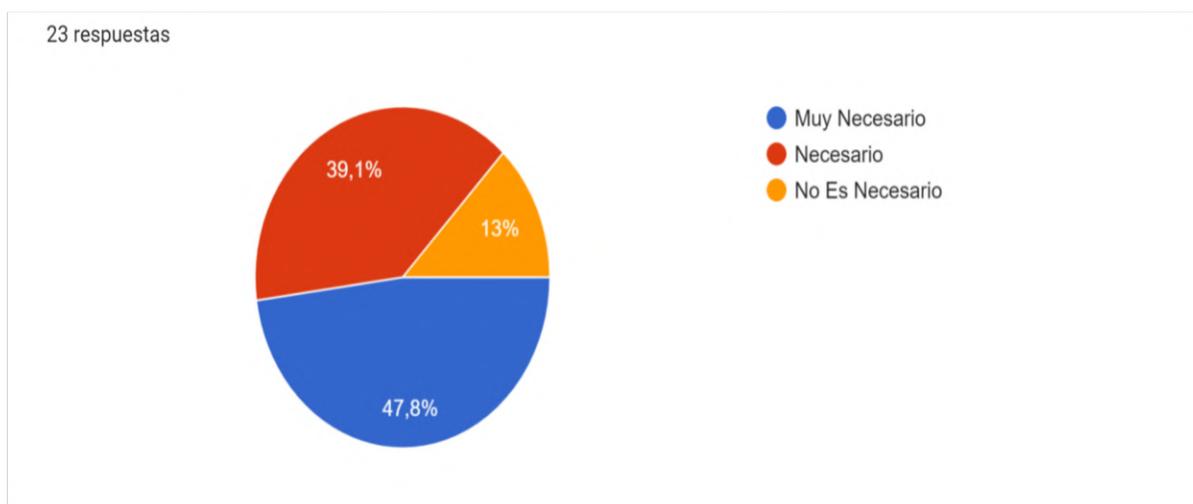


Fuente: El Gráfico muestra los resultados obtenidos en la sexta pregunta de la encuesta.

La gráfica muestra la frecuencia con la que los usuarios del sistema Genesis SIA encuentran errores en su funcionamiento. Los resultados revelan que un pequeño porcentaje, el 4,3%, afirma que nunca encuentra errores en el sistema, mientras que el 43,5% indica que casi nunca encuentra errores. Por otro lado, un 39,1% de los usuarios menciona que ocasionalmente encuentra errores en el sistema. Sin embargo, es importante destacar que un 13% de los usuarios reporta encontrar errores en el sistema cada mes.

Figura 18.

Resultado de la pregunta ¿Cree necesario la creación de un nuevo sistema información académica Genesis SIA?



Fuente: El Gráfico muestra los resultados obtenidos en la septima pregunta de la encuesta.

La gráfica muestra la opinión de los usuarios sobre la necesidad de crear un nuevo sistema Genesis SIA. Según los resultados, un 39,1% de los usuarios considera que es "necesario" la creación de un nuevo sistema, mientras que un 47,8% lo califica como "muy necesario". Sin embargo, un 13% de los usuarios opina que no es necesario.

Estos datos reflejan una clara demanda por parte de los usuarios de un nuevo sistema Genesis SIA. La mayoría de los usuarios enfatiza la necesidad o incluso la alta necesidad de contar con una nueva versión del sistema.

Los resultados de la octava pregunta muestran que los usuarios del sistema de información académico Genesis SIA solicitan la inclusión de videos de ayuda para facilitar el uso del sistema. Además, los usuarios sugieren que el sistema debe mejorar su experiencia de uso para ser más amigable y fácil de utilizar. En este sentido, las plataformas como WebColegios

son consideradas un buen ejemplo a seguir, ya que proporcionan herramientas útiles y accesibles para los usuarios. En resumen, los usuarios del Genesis SIA buscan una plataforma de gestión académica intuitiva, fácil de usar y con recursos de ayuda adecuados para mejorar su experiencia de uso.

4.2. Segundo objetivo

Caracterizar tecnologías y aspectos de arquitectura con ello sus atributos y propiedades a través de modelos comparativos.

Recopilar documentación sobre tecnologías y arquitecturas utilizadas en desarrollo de sistemas de información académico.

Analizar las tecnologías y arquitecturas de desarrollo ya documentadas.

Seleccionar las tecnologías y arquitecturas que se adaptan al contexto del Genesis SIA versión 4.0.

Para cumplir con el segundo objetivo de caracterizar tecnologías y aspectos de arquitectura a través de modelos comparativos, identificando sus atributos y propiedades se

realizaron los siguientes pasos:

Se llevó a cabo una recopilación de documentación sobre las tecnologías y arquitecturas utilizadas en el desarrollo de sistemas de información académicos.

Se analizó la documentación recopilada sobre las tecnologías y arquitecturas de desarrollo.

Se seleccionaron las tecnologías y arquitecturas que se adecuan al contexto del Genesis SIA versión 4.0.

Estos pasos permitieron obtener una visión más clara de las tecnologías y arquitecturas que se pueden utilizar en el desarrollo del Genesis SIA versión 4.0. La recopilación de documentación y el análisis de las tecnologías y arquitecturas existentes proporcionaron información valiosa sobre los atributos y propiedades de cada una de ellas. La selección de las tecnologías y arquitecturas adecuadas garantizará un diseño de arquitectura eficiente y efectivo para el Genesis SIA versión 4.0.

4.2.1. Documentación de tecnologías y arquitecturas de software

El diseño de un sistema de gestión de información académica requiere del uso de varias tecnologías para su correcto funcionamiento y seguridad. Entre las tecnologías más comunes se encuentran la base de datos, lenguajes de programación, frameworks, servidores web, herramientas de frontend, sistemas operativos y tecnologías de seguridad. Todas estas tecnologías trabajan en conjunto para proporcionar una plataforma eficiente, rápida y segura para el manejo de la información académica. En el siguiente texto se explican de manera detallada de cada una de estas tecnologías (Conolly & Carolyn, 2015).

Bases de Datos

La tecnología de base de datos es una herramienta fundamental para el diseño de un sistema de gestión de información académica (SGIA). Una base de datos es un conjunto de

información estructurada y organizada que se almacena en un sistema informático, y se utiliza para gestionar grandes volúmenes de datos (Coronel & Morris, 2017). En el contexto de un SGIA, una base de datos permite almacenar y administrar información sobre estudiantes, profesores, cursos, calificaciones, entre otros.

Existen diferentes tipos de bases de datos, como las *bases de datos relacionales* y las *bases de datos no relacionales*. Las bases de datos relacionales son las más utilizadas en el desarrollo de SGIA, debido a su capacidad para establecer relaciones entre diferentes entidades de datos y garantizar la integridad de los datos. En una base de datos relacional, los datos se organizan en tablas, y las tablas se relacionan entre sí mediante claves primarias y foráneas (Rendón, 2019).

Uno de los sistemas de gestión de bases de datos relacionales más populares debido a su rapidez, facilidad de uso y escalabilidad en la industria es *MySQL*, sistema de gestión de bases de datos relacional de código abierto, que se utiliza en una gran cantidad de aplicaciones web y móviles (MySQL, 2023) (Robledano, 2019).

Así también, se encuentra *Oracle*, que es un sistema de gestión de bases de datos relacional de propósito general, que se utiliza en aplicaciones empresariales y en la gestión de grandes volúmenes de datos. Es conocido por su alto rendimiento, escalabilidad y seguridad (Oracle, 2023).

Otro sistema de gestión de bases de datos relacionales muy utilizado en el desarrollo de SGIA es *PostgreSQL*. PostgreSQL es un sistema de gestión de bases de datos relacional de

código abierto, que se utiliza en aplicaciones web y móviles. PostgreSQL es conocido por su estabilidad, seguridad y capacidad para manejar grandes volúmenes de datos (PostgreSQL, 2023).

Es importante tener en cuenta que el diseño de la base de datos es crucial para el éxito de un SGIA. Un buen diseño de la base de datos debe tener en cuenta la integridad de los datos, la eficiencia en la consulta de datos y la escalabilidad del sistema (Conolly & Carolyn, 2015). Para lograr un buen diseño de la base de datos, es recomendable seguir buenas prácticas de diseño, como la normalización de la base de datos y la utilización de índices para acelerar las consultas (Giménez, 2019).

Existen muchos proyectos y casos de estudio en los que se ha utilizado la tecnología de base de datos para el diseño de un sistema de gestión de información académica. A continuación, se presentan algunos ejemplos destacados:

Universidad de California, Berkeley: En la Universidad de California, Berkeley, se implementó un sistema de gestión de información académica basado en una base de datos relacional. El sistema permitía a los estudiantes y profesores acceder a la información académica, incluyendo notas, horarios de clases y registros de asistencia. La base de datos utilizada fue MySQL, y se utilizó el lenguaje de programación PHP para desarrollar la aplicación (UC Berkeley, 2023).

Universidad de Sao Paulo: En la Universidad de Sao Paulo, Brasil, se desarrolló un sistema de gestión de información académica basado en una base de datos NoSQL. El sistema

permitía a los estudiantes y profesores acceder a la información académica, incluyendo notas, horarios de clases y registros de asistencia. La base de datos utilizada fue MongoDB, y se utilizó el framework Ruby on Rails para desarrollar la aplicación (Oliveira & Ferreira, 2016).

Universidad de Granada: En la Universidad de Granada, España, se implementó un sistema de gestión de información académica basado en una base de datos relacional. El sistema permitía a los estudiantes y profesores acceder a la información académica, incluyendo notas, horarios de clases y registros de asistencia. La base de datos utilizada fue Oracle, y se utilizó el lenguaje de programación Java para desarrollar la aplicación (Vázquez-Poletti, Salas, & Vaquero, 2013).

Sistema de gestión de información académica de la Universidad Tecnológica de Panamá (UTP): este sistema utiliza una base de datos Oracle y permite la gestión de información académica de estudiantes, profesores y cursos (Fernández & García, 2012).

Sistema de información académica de la Universidad Tecnológica de Pereira (UTP): este sistema utiliza una base de datos MySQL y permite la gestión de información académica de estudiantes, profesores y cursos, así como la generación de informes y estadísticas (Arroyave & López, 2013).

Sistema de gestión académica de la Universidad Autónoma de Bucaramanga (UNAB): este sistema utiliza una base de datos Microsoft SQL Server y permite la gestión de información

académica de estudiantes, profesores y cursos, así como la generación de informes y estadísticas (Salazar & Linares, 2016).

En todos estos proyectos y casos de estudio, se utilizó la tecnología de base de datos para almacenar y gestionar la información académica de manera eficiente y segura. La elección de la base de datos específica dependió de los requisitos del sistema y las necesidades del negocio.

Lenguajes de Programación

Los lenguajes de programación son un conjunto de instrucciones que permiten a los programadores escribir código que la computadora puede entender y ejecutar. En el contexto de un SGIA, los lenguajes de programación se utilizan para desarrollar las aplicaciones que permiten a los usuarios interactuar con la base de datos y realizar tareas específicas, como ingresar información, consultar datos y generar informes (Mendoza, 2020).

Existen diferentes tipos de lenguajes de programación, como los lenguajes de programación orientados a objetos, los lenguajes de programación de scripting y los lenguajes de programación de bajo nivel (Martin, 2018). Cada tipo de lenguaje de programación tiene sus propias ventajas y desventajas, y la elección del lenguaje de programación dependerá de las necesidades específicas del SGIA.

Dentro de los lenguajes más utilizados en el desarrollo de aplicaciones se encuentran:

Java: lenguaje de programación orientado a objetos de propósito general, que se utiliza en una gran cantidad de aplicaciones web y móviles. Java es muy popular debido a su capacidad para desarrollar aplicaciones escalables y seguras (Deitel & Harvey, 2018).

PHP: es un lenguaje de programación de scripting de código abierto, que se utiliza en una gran cantidad de aplicaciones web y en el desarrollo de SGIA. PHP es muy popular debido a su facilidad de uso y a la gran cantidad de recursos disponibles en línea para aprender y desarrollar aplicaciones (Welling & Thomson, 2016).

Python: es un lenguaje de programación de propósito general, que se utiliza en una gran cantidad de aplicaciones web y en el desarrollo de SGIA. Python es muy popular debido a su facilidad de uso y a la gran cantidad de bibliotecas y frameworks disponibles para el desarrollo de aplicaciones (Downey, 2012).

Según (Enriquez, 2016), dice que “es importante tener en cuenta que el lenguaje de programación es solo una parte del desarrollo de un SGIA. Para lograr un buen diseño de la aplicación, es necesario tener en cuenta otros aspectos como la arquitectura de la aplicación, la usabilidad de la interfaz y la seguridad de la aplicación”.

Existen proyectos y casos de estudio donde han utilizado lenguajes de programación como: Java, Python y C#. Algunos de estos proyectos fueron:

Sistema de Gestión de Información Académica de la Universidad Nacional Autónoma de México (UNAM) - Java: El sistema de gestión de información académica de la UNAM es una plataforma integral que ofrece servicios y recursos en línea a los estudiantes, profesores y administrativos de la universidad. El sistema fue desarrollado utilizando el lenguaje de programación Java y la plataforma de desarrollo Spring Framework. El proyecto fue desarrollado por el Centro de Enseñanza de Lenguas Extranjeras (CELE) de la UNAM y se encuentra en producción desde el 2012 (Universidad Nacional Autónoma de México, 2021) (Coordinación General de Planeación y Simplificación de la Gestión Institucional, 2023)

Sistema de Gestión de Notas de la Universidad de California, Berkeley - Python: El sistema de gestión de notas de la Universidad de California, Berkeley es un proyecto de código abierto que permite a los estudiantes y profesores de la universidad gestionar notas, tareas, calificaciones y asistencia en línea. El sistema fue desarrollado utilizando el lenguaje de programación Python y la plataforma de desarrollo Django. El proyecto fue desarrollado por el Centro de Tecnología de la Información de la universidad y se encuentra en producción desde el 2010 (Eastern Daylight Time, 2020).

Sistema de Gestión de Exámenes de la Universidad de Illinois - C#: El sistema de gestión de exámenes de la Universidad de Illinois es un proyecto de código abierto que permite a los profesores de la universidad crear y gestionar exámenes en línea. El sistema fue desarrollado utilizando el lenguaje de programación C# y la plataforma de desarrollo ASP.NET. El proyecto fue desarrollado por el Centro de Tecnología de la Información de la universidad y se encuentra en producción desde el 2012 (University of Illinois, 2023).

Frameworks

Los frameworks permiten a los desarrolladores crear aplicaciones de manera más rápida y eficiente, sin tener que escribir todo el código desde cero. Siendo un conjunto de herramientas y bibliotecas que proporcionan una estructura para el desarrollo de aplicaciones y facilitan la integración con otras tecnologías (Ortega, Guevara, & Benavides, 2017) .

Existen muchos frameworks que se utilizan en el desarrollo de aplicaciones web y móviles, y la elección del framework dependerá de las necesidades específicas del SGIA. Algunos de los frameworks más populares son:

Django: Es un framework de código abierto en Python, que se utiliza en la creación de aplicaciones web complejas y escalables. Django es muy popular en el desarrollo de SGIA debido a su facilidad de uso y su capacidad para manejar grandes cantidades de datos (Holvaty & Kaplan-Moss, 2009).

Ruby on Rails: Es un framework de código abierto en Ruby, que se utiliza en la creación de aplicaciones web. Ruby on Rails es muy popular debido a su rapidez y eficiencia en el desarrollo de aplicaciones y su capacidad para integrarse con otras tecnologías (Rails Core Team, 2016).

Laravel: Es un framework de código abierto en PHP, que se utiliza en la creación de aplicaciones web. Laravel es muy popular debido a su facilidad de uso y su capacidad para manejar grandes cantidades de datos (Stauffer, 2019).

React: Es un framework de código abierto en JavaScript, que se utiliza en la creación de aplicaciones web y móviles. React es muy popular debido a su capacidad para crear interfaces de usuario interactivas y dinámicas (Lebensold, 2018).

Spring Boot: es un framework de lenguaje de programación de Java que se utiliza comúnmente para desarrollar aplicaciones de alta calidad y escalables en poco tiempo (VMware Tanzu, 2023). Fue creado en 2012 por Pivotal Software, Inc. y se ha vuelto muy popular entre los desarrolladores debido a su facilidad de uso y su capacidad para crear aplicaciones web complejas en poco tiempo (Webb, Syer, & Long, 2019). En el diseño de un sistema de gestión de información académica, Spring Boot puede ser una herramienta muy útil debido a su capacidad para integrar fácilmente con otras tecnologías, como bases de datos, servicios web y herramientas de frontend (Villagómez & Fernando, 2020).

La elección de un framework específico dependerá de los requisitos del sistema y las preferencias del equipo de desarrollo.

Servidores Web

Los servidores web son una tecnología fundamental para el diseño de un sistema de gestión de información académica (SGIA). Un servidor web es un software que se ejecuta en un servidor y que se encarga de recibir y responder a las solicitudes de los clientes a través de internet. El servidor web es el responsable de servir los archivos y recursos web que conforman

una aplicación web y de manejar las conexiones entre el cliente y el servidor (Chavarria-Neira & Gudiño, 2017).

Existen muchos servidores web disponibles, y la elección del servidor dependerá de las necesidades específicas del SGIA. Algunos de los servidores web más populares son:

Apache: Es uno de los servidores web más populares y ampliamente utilizados en el mundo. Es de código abierto y se ejecuta en una gran variedad de sistemas operativos, incluyendo Linux, Windows y macOS (The Apache Software Foundation., 2023) (Palma Pérez, 2020).

Nginx: Es un servidor web ligero y de alto rendimiento que se ha vuelto muy popular en los últimos años. Es de código abierto y se ejecuta en sistemas operativos como Linux, macOS y Windows (Palma Pérez, 2020).

Microsoft IIS: Es el servidor web de Microsoft y se ejecuta en sistemas operativos Windows. Es ampliamente utilizado en entornos empresariales y ofrece una buena integración con otras tecnologías de Microsoft (Microsoft Corporation, 2021).

Tomcat: Es un servidor web de código abierto que se utiliza principalmente en la ejecución de aplicaciones web Java. Se ejecuta en sistemas operativos como Linux, Windows y macOS (Apache Software Foundation, 2021).

Es importante tener en cuenta que los servidores web son una parte importante de una infraestructura completa de aplicaciones web, que incluye también bases de datos, servidores de aplicaciones y otros componentes (Chavarría-Neira & Gudiño, 2017).

Los servidores web son esenciales en el diseño de sistemas de gestión de información académica, ya que permiten que los usuarios accedan a la información a través de una interfaz web. A continuación, se presentan algunos proyectos y casos de estudio donde se ha utilizado la tecnología de servidores web:

Moodle: Moodle es un sistema de gestión de aprendizaje de código abierto que utiliza tecnologías web para permitir a los profesores y estudiantes interactuar y compartir información. Está diseñado para ser utilizado en entornos académicos y cuenta con una gran cantidad de funciones y características para facilitar la gestión de la información académica. Moodle utiliza Apache como servidor web y PHP como lenguaje de programación (Moodle, 2022).

Blackboard: Blackboard es otro sistema de gestión de aprendizaje que se utiliza ampliamente en entornos académicos. Proporciona herramientas para la gestión del contenido, la colaboración y la evaluación, y se integra con una amplia variedad de herramientas y tecnologías para proporcionar una experiencia de aprendizaje integrada. Blackboard utiliza el servidor web Tomcat y el lenguaje de programación Java (Aliaga Meléndez & Dávila Rojas, 2021).

Canvas: Canvas es un sistema de gestión de aprendizaje basado en la nube que se utiliza en muchas instituciones educativas. Proporciona una variedad de herramientas y características para la gestión del contenido, la colaboración y la evaluación, y es conocido por su facilidad de

uso y flexibilidad. Canvas utiliza Nginx como servidor web y Ruby como lenguaje de programación (Flores Huacho, 2021).

Herramientas Frontend

Las herramientas de frontend son fundamentales para el diseño de un sistema de gestión de información académica (SGIA), ya que permiten crear la interfaz de usuario y hacer que la aplicación sea accesible para los usuarios finales. Según (Pérez Pérez), “para realizar una implementación adecuada y eficiente se utilizan las siguientes herramientas”:

HTML: Es el lenguaje de marcado estándar para la creación de páginas web. HTML se utiliza para definir la estructura y contenido de una página web. Permite crear elementos como títulos, párrafos, encabezados, enlaces, imágenes y tablas (Mozilla Developer Network, HTML: Hypertext Markup Language, 2021).

CSS: Las hojas de estilo en cascada (CSS) se utilizan para definir la apariencia y el diseño de una página web. CSS se utiliza para definir los estilos de los elementos HTML, como el color, la fuente, el tamaño, el espaciado y la disposición (Mozilla Developer Network, CSS: Cascading Style Sheets, 2021).

JavaScript: Es un lenguaje de programación que se utiliza para crear interactividad en una página web. JavaScript se utiliza para agregar efectos visuales, validación de formularios y

otros comportamientos dinámicos en una página web (Mozilla Developer Network, JavaScript, 2021).

Frameworks de frontend: Son conjuntos de herramientas y bibliotecas que se utilizan para acelerar el desarrollo de una aplicación web. Algunos de los frameworks más populares son: React, Angular & Vue (Hernández, 2019).

La tecnología de frontend es un conjunto de herramientas que se utilizan para diseñar y desarrollar la interfaz de usuario de una aplicación web. En el caso de un sistema de gestión de información académica, es importante contar con una interfaz intuitiva y fácil de usar que permita a los usuarios acceder y gestionar la información de manera eficiente (Gómez, 2019).

A continuación, se presentan algunos proyectos y casos de estudio donde se ha utilizado tecnología de frontend para el diseño de sistemas de gestión de información académica:

SISAC - Sistema de Información Académico de la Universidad de Chile: SISAC es un sistema de gestión académica que utiliza AngularJS como framework de frontend. Permite a los estudiantes consultar sus notas, inscribirse en cursos, solicitar certificados y realizar otras acciones relacionadas con su vida académica. El uso de AngularJS permite una rápida carga de la página y una experiencia de usuario fluida (Maturana & Méndez, 2018).

SIGA - Sistema de Información para la Gestión Académica de la Universidad de los Andes: SIGA es un sistema de gestión académica que utiliza ReactJS como framework de frontend. Permite a los estudiantes gestionar su información académica, como notas, horarios de

clase y materiales de estudio. El uso de ReactJS permite una rápida actualización de la página y una experiencia de usuario dinámica (Mendoza M. J., 2020).

ACAD-Web - Sistema de gestión académica de la Universidad Nacional de Colombia:
ACAD-Web es un sistema de gestión académica que utiliza Bootstrap como framework de frontend. Permite a los estudiantes acceder a su información académica, como notas y horarios de clase, y realizar trámites académicos en línea. El uso de Bootstrap permite una interfaz de usuario atractiva y adaptable a diferentes dispositivos (González & Giraldo, 2016).

Sistemas Operativos

Los sistemas operativos son esenciales para el diseño de un sistema de gestión de información académica (SGIA), ya que proporcionan una plataforma para ejecutar el software necesario para la gestión de datos y la presentación de información. Los sistemas operativos son programas que administran los recursos del hardware y proporcionan servicios a las aplicaciones de software (Tanenbaum & Woodhull, 2015). En el contexto de un SGIA, los sistemas operativos pueden ser utilizados para administrar servidores, almacenar y acceder a datos, y proporcionar una interfaz de usuario para los usuarios finales (Silberschatz, Galvin, & Gagne, 2018).

Hay varios sistemas operativos que se utilizan comúnmente en la creación de aplicaciones de software, incluyendo:

Windows: Windows es un sistema operativo de Microsoft que es popular en entornos de escritorio y empresariales. Windows proporciona una interfaz gráfica de usuario y una amplia gama de herramientas y aplicaciones que pueden ser utilizadas para la gestión de datos y la presentación de información (De la Cruz Barboza, 2018).

MacOS: MacOS es un sistema operativo desarrollado por Apple que se utiliza en los ordenadores de la marca. MacOS también proporciona una interfaz gráfica de usuario y una amplia gama de herramientas y aplicaciones que pueden ser utilizadas para la gestión de datos y la presentación de información (De la Cruz Barboza, 2018).

Linux: Linux es un sistema operativo de código abierto que es popular en entornos empresariales y de servidor. Linux es conocido por ser robusto y escalable, lo que lo hace una opción popular para la gestión de datos y la presentación de información en un SGIA (De la Cruz Barboza, 2018).

Unix: Unix es un sistema operativo de línea de comandos que se utiliza comúnmente en entornos empresariales y de servidor. Unix es conocido por ser seguro y escalable, lo que lo hace una opción popular para la gestión de datos y la presentación de información en un SGIA (De la Cruz Barboza, 2018).

Los sistemas operativos son una parte fundamental en el diseño de cualquier sistema informático, incluyendo los sistemas de gestión de información académica. A continuación, se

presentan algunos proyectos y casos de estudio donde se ha utilizado tecnología de sistemas operativos para el diseño de sistemas de gestión de información académica:

Sistema de gestión académica basado en Linux: Este proyecto desarrollado en la Universidad de Zulia, en Venezuela, implementó un sistema de gestión académica utilizando Linux como sistema operativo base. El sistema incluye módulos para la gestión de estudiantes, profesores, cursos y calificaciones, así como para la generación de informes y estadísticas. El proyecto se centró en la seguridad y la facilidad de uso para los usuarios finales (Viloria, Valles, & Morales, 2013).

Sistema de información académica basado en Windows Server: En la Universidad de La Sabana, en Colombia, se implementó un sistema de información académica utilizando Windows Server como sistema operativo base. El sistema incluye módulos para la gestión de estudiantes, profesores, cursos, notas y horarios, así como para la generación de informes y estadísticas. El proyecto se centró en la integración con otros sistemas de la universidad y la escalabilidad (Arias & Murillo, 2016).

Seguridad

La seguridad es un aspecto fundamental en el diseño de cualquier sistema de gestión de información académica (SGIA), ya que estos sistemas almacenan datos sensibles de los estudiantes, profesores y personal administrativo. Las tecnologías de seguridad se utilizan para proteger los datos de accesos no autorizados, asegurar la privacidad de los usuarios y garantizar

la integridad de los datos (Stallings, 2017) . A continuación, se describen algunas de las tecnologías de seguridad más comunes utilizadas en los SGIA:

Criptografía: La criptografía es el estudio de los algoritmos que se utilizan para proteger la información confidencial. Los algoritmos de criptografía se utilizan para cifrar los datos y protegerlos de accesos no autorizados. Además, la criptografía también se utiliza para autenticar la identidad de los usuarios y garantizar la integridad de los datos (Cheswick & Bellovin, 2003).

Firewalls: Los firewalls son dispositivos que se utilizan para proteger la red de accesos no autorizados. Los firewalls bloquean el tráfico de red no deseado y permiten que solo los usuarios autorizados accedan a la red del SGIA (Cheswick & Bellovin, 2003).

Sistemas de autenticación: Los sistemas de autenticación se utilizan para verificar la identidad de los usuarios antes de permitirles acceder al SGIA. Los sistemas de autenticación pueden incluir la verificación de credenciales, como nombres de usuario y contraseñas, y pueden ser complementados con otras formas de autenticación, como la autenticación biométrica (Stajano, 2011) (Kizza, 2016).

Auditorías de seguridad: Las auditorías de seguridad se utilizan para evaluar la seguridad del SGIA y detectar posibles vulnerabilidades. Las auditorías de seguridad pueden incluir la revisión de los registros de actividad, la realización de pruebas de penetración y la evaluación de la conformidad con los estándares de seguridad (ISO/IEC, 2013).

A continuación, se presentan algunos proyectos y casos de estudio que destacan la importancia de la tecnología de seguridad en el diseño de sistemas de gestión de información académica.

Proyecto SISRA: El proyecto SISRA (Sistema de Información y Seguimiento de la Responsabilidad Académica) es un sistema de gestión de información académica desarrollado por la Universidad Nacional de Colombia. El sistema incluye medidas de seguridad, como autenticación de usuarios, permisos de acceso y encriptación de datos, para proteger la información confidencial (Rueda & Orozco, 2014).

Caso de estudio de la Universidad de Illinois en Urbana-Champaign: La Universidad de Illinois en Urbana-Champaign implementó una serie de medidas de seguridad para proteger la información académica en su sistema de gestión de información académica. Estas medidas incluyen autenticación de usuarios, cifrado de datos, y monitoreo continuo de la red para detectar posibles amenazas de seguridad (Ho, 2017).

Proyecto SIAF: El proyecto SIAF (Sistema de Información Académica y Financiera) es un sistema de gestión de información académica y financiera utilizado por varias universidades en México. El sistema incluye medidas de seguridad, como autenticación de usuarios, cifrado de datos y permisos de acceso, para proteger la información confidencial (Martínez & Mendoza, 2016).

La seguridad es una consideración fundamental en el diseño de sistemas de gestión de información académica y debe abordarse mediante la implementación de medidas de seguridad adecuadas, como autenticación de usuarios, cifrado de datos y permisos de acceso. Los proyectos y casos de estudio mencionados anteriormente proporcionan ejemplos concretos de cómo las universidades han abordado la seguridad en sus sistemas de gestión de información académica.

Una arquitectura de software es esencial para el diseño de un sistema de gestión de información académica. Existen varias arquitecturas comunes, como la basada en capas, la orientada a servicios (SOA), la orientada a microservicios y la basada en eventos, cada una con sus propias ventajas y desventajas.

Arquitectura Basada En Capas

La arquitectura basada en capas es un enfoque de diseño de software que organiza una aplicación en capas lógicas, donde cada capa tiene una responsabilidad específica y se comunica solo con capas adyacentes. Esta arquitectura es muy utilizada en el diseño de sistemas de gestión de información académica debido a su modularidad, escalabilidad y facilidad de mantenimiento (Fowler, 2002) (Richards & Ford, 2015).

La arquitectura basada en capas se divide típicamente en tres capas principales: la capa de presentación (también conocida como capa de interfaz de usuario), la capa de lógica de negocio y la capa de acceso a datos. La capa de presentación maneja la interacción del usuario con el sistema y muestra la información. La capa de lógica de negocio contiene la lógica de la

aplicación y procesa las solicitudes del usuario. La capa de acceso a datos se encarga de la interacción con la base de datos y gestiona el almacenamiento y recuperación de la información (Richards & Ford, 2015).

Según el libro "Software Architecture Patterns" de Mark Richards y Neal Ford, la arquitectura basada en capas es una de las arquitecturas más utilizadas en la industria del software debido a su simplicidad y flexibilidad. Además, permite la separación de preocupaciones y la reutilización de componentes (Richards & Ford, 2015).

En el caso de un sistema de gestión de información académica, la capa de presentación podría incluir una interfaz de usuario para que los estudiantes y profesores interactúen con el sistema, la capa de lógica de negocio podría incluir procesamiento de solicitudes de inscripción y gestión de horarios, y la capa de acceso a datos podría incluir la interacción con una base de datos para almacenar información de estudiantes y profesores (Richards & Ford, 2015).

Un ejemplo de aplicación de esta arquitectura en el diseño de un sistema de gestión de información académica se puede encontrar en el trabajo de investigación realizado por (Sahay & Patwardhan, 2015). En este estudio, se describe la implementación de una arquitectura basada en capas para un sistema de gestión de información académica en una institución educativa de la India. En este sistema, la capa de datos se compone de una base de datos relacional que almacena información de los estudiantes, profesores, cursos y calificaciones. La capa intermedia se compone de un servidor de aplicaciones que maneja la lógica de negocio y la gestión de procesos. La capa superior se compone de una interfaz de usuario web que permite a los usuarios acceder a la información del sistema.

Otro ejemplo de aplicación de la arquitectura basada en capas en un sistema de gestión de información académica se puede encontrar en el trabajo de investigación realizado por (Li, Chen, Zhang, & Liu, 2016). En este estudio, se describe la implementación de una arquitectura basada en capas para un sistema de gestión de información académica en una universidad china. En este sistema, la capa de datos se compone de una base de datos relacional que almacena información de los estudiantes, profesores, cursos y calificaciones. La capa intermedia se compone de un servidor de aplicaciones que maneja la lógica de negocio y la gestión de procesos. La capa superior se compone de una interfaz de usuario web que permite a los usuarios acceder a la información del sistema. Además, se incorporó una capa adicional para el acceso móvil al sistema.

En conclusión, la arquitectura basada en capas es una opción común y efectiva para el diseño de sistemas de gestión de información académica. La división de responsabilidades y la clara separación entre las capas permite una fácil mantenibilidad y escalabilidad del sistema.

Arquitectura Orientada a Servicios (SOA)

La Arquitectura Orientada a Servicios (SOA, por sus siglas en inglés) es una técnica de diseño de software que se enfoca en la creación de servicios reutilizables y modulares, que pueden ser combinados para construir aplicaciones de software complejas. En un sistema SOA, los servicios son unidades de funcionalidad que se ofrecen a través de una red, y que pueden ser utilizados por otros servicios o aplicaciones (Papazoglou, 2016).

Para el diseño de un sistema de gestión de información académica, la arquitectura SOA puede ser una opción viable, ya que permite la integración de diferentes sistemas y aplicaciones, lo que puede mejorar la interoperabilidad y la eficiencia de los procesos. Por ejemplo, mediante la utilización de servicios de autenticación y autorización, se puede asegurar el acceso adecuado a la información y los recursos, lo que es especialmente importante en el contexto académico (Erl, 2005).

La arquitectura SOA consta de tres componentes principales: el proveedor de servicios, el consumidor de servicios y el registro de servicios. El proveedor de servicios es responsable de ofrecer servicios que pueden ser consumidos por otros sistemas o aplicaciones. El consumidor de servicios es un sistema o aplicación que utiliza los servicios ofrecidos por el proveedor de servicios. El registro de servicios es un repositorio que almacena información sobre los servicios disponibles, como su ubicación, tipo, descripción y requisitos de acceso (Papazoglou, 2016) (IBM, 2004).

Algunas ventajas de la arquitectura SOA incluyen la flexibilidad, la escalabilidad y la reutilización de los servicios. Sin embargo, también existen desafíos asociados con la implementación de SOA, como la complejidad de la gestión de servicios y la necesidad de una planificación y coordinación adecuadas entre los diferentes equipos involucrados (Chappell, 2004).

Un ejemplo de aplicación de SOA en un sistema de gestión de información académica es el proyecto de la Universidad de Jaén en España. En este proyecto, se utilizó SOA para crear una arquitectura modular y escalable para el sistema de gestión académica de la universidad. La arquitectura se dividió en varios servicios, cada uno responsable de una función específica, como la gestión de cursos, la gestión de calificaciones y la gestión de usuarios. Estos servicios se diseñaron para ser independientes y reutilizables, lo que permitió una mayor flexibilidad y escalabilidad en el sistema (Cruz, Sanchez, & Guerrero, 2013).

Otro ejemplo es el proyecto de la Universidad de Guadalajara en México, donde se utilizó SOA para desarrollar un sistema de gestión académica que integra varias funcionalidades, como la inscripción en línea, la consulta de calificaciones, la gestión de horarios y la consulta de planes de estudio. El sistema se diseñó utilizando una arquitectura basada en servicios, donde cada servicio se implementó como una función independiente y se integró en el sistema a través de interfaces de servicio (Gutiérrez & Pérez, 2014).

Arquitectura Orientada A Microservicios

La arquitectura orientada a microservicios es un enfoque de diseño de software que se enfoca en construir aplicaciones como un conjunto de pequeños servicios independientes que se comunican entre sí a través de interfaces bien definidas. Cada servicio está diseñado para realizar una tarea específica y puede ser desplegado y escalado de manera independiente, lo que permite una mayor flexibilidad y agilidad en el desarrollo y mantenimiento de la aplicación (Newman, 2015).

En el contexto de un sistema de gestión de información académica, la arquitectura orientada a microservicios puede ser útil para construir diferentes servicios para la gestión de información académica, tales como la gestión de notas, la gestión de horarios, la gestión de profesores y la gestión de estudiantes. Cada servicio puede ser desarrollado y desplegado de manera independiente, lo que permite una mayor escalabilidad y flexibilidad en la gestión del sistema (Balalaie, Heydarnoori, & Jamshidi, 2016).

Uno de los principales beneficios de la arquitectura orientada a microservicios es que permite la modularización de la aplicación en servicios independientes, lo que facilita el mantenimiento y la evolución de la aplicación. Además, esta arquitectura también permite una mayor flexibilidad en cuanto a la elección de tecnologías y herramientas para cada servicio, lo que permite utilizar la tecnología más adecuada para cada tarea específica (Newman, 2015) (Balalaie, Heydarnoori, & Jamshidi, 2016).

Un ejemplo de proyecto que utilizó la arquitectura de microservicios para la gestión de información académica es el sistema de información de la Universidad de Carolina del Norte. En este proyecto, la arquitectura de microservicios se utilizó para crear un sistema de gestión de cursos que se integró con otros sistemas de la universidad, como el sistema de gestión de la biblioteca y el sistema de gestión de la investigación. La arquitectura de microservicios permitió a los desarrolladores integrar fácilmente estos sistemas sin afectar el rendimiento general del sistema de gestión de información académica (Ahmed, Hossain, & Alam, 2020).

Otro ejemplo es el sistema de gestión de información académica de la Universidad de Colorado, que utiliza una arquitectura de microservicios para la gestión de cursos, calificaciones y asesoramiento académico. Esta arquitectura permitió a la universidad agregar nuevas funcionalidades y servicios sin afectar la estabilidad del sistema en general (Kamal, Shah, & Hussain, 2019).

Arquitectura Basada En Eventos

La arquitectura basada en eventos es una forma de diseño de software que se enfoca en la comunicación y procesamiento de eventos. En esta arquitectura, los componentes de software son diseñados para interactuar a través de eventos y sus respectivos manejadores. Un evento puede ser cualquier tipo de acción o cambio en el estado del sistema, como una solicitud de un usuario, una actualización de una base de datos o una respuesta de un servicio externo (IBM., 2021).

Una de las principales ventajas de la arquitectura basada en eventos es su capacidad para manejar grandes volúmenes de eventos y procesarlos en tiempo real. Esto se logra a través del uso de tecnologías como colas de mensajes y buses de eventos, que permiten que los eventos sean enrutados y procesados de manera eficiente. Además, la arquitectura basada en eventos facilita la integración con sistemas externos, ya que los eventos pueden ser consumidos y producidos por servicios externos (IBM., 2021) (Han, Lee, & Yoo, 2017).

Un ejemplo de aplicación de la arquitectura basada en eventos en un sistema de gestión de información académica es el trabajo de investigación realizado por Han y sus colegas en la Universidad de Pusan en Corea del Sur. En su estudio, los autores diseñaron un sistema de gestión de información académica basado en eventos para manejar la inscripción de estudiantes y la actualización de notas. El sistema utilizó una arquitectura basada en eventos para garantizar la escalabilidad y la eficiencia del procesamiento de los eventos (Han, Lee, & Yoo, 2017).

En resumen, la arquitectura basada en eventos es una forma efectiva de diseñar un sistema de gestión de información académica que puede manejar grandes volúmenes de eventos y procesarlos en tiempo real. Esta arquitectura también facilita la integración con sistemas externos y permite la implementación de componentes de software de forma independiente.

4.2.2. Analizar las tecnologías y arquitecturas de desarrollo ya documentadas

Para llevar a cabo el análisis, se crearon cuadros comparativos para cada tecnología y arquitectura de software. Estos cuadros permitieron obtener una visión clara y detallada de las propiedades, ventajas y desventajas de cada opción, lo que resulta crucial para seleccionar las tecnologías y arquitecturas más adecuadas para el diseño de arquitectura efectivo y eficiente del Genesis SIA versión 4.0.

Tabla 1.

Cuadro Comparativo para Tecnología de Bases de Datos

Tecnología de Bases de Datos	Propiedades	Ventajas	Desventajas
Relacionales	Estructuradas en tablas Modelo de datos basado en relaciones Utilizan SQL como lenguaje de consulta Permiten el establecimiento de relaciones entre tablas	Facilidad de uso Flexibilidad Capacidad para manejar grandes cantidades de datos Buena escalabilidad	No son eficientes para trabajar con datos no estructurados Pueden tener problemas de rendimiento con operaciones complejas Requieren conocimientos técnicos para su configuración
NoSQL	No utilizan SQL como lenguaje de consulta Pueden manejar grandes cantidades de datos no estructurados Modelos de datos flexibles	Rendimiento muy rápido para grandes volúmenes de datos Flexibilidad en el modelo de datos Escalabilidad horizontal fácil de implementar	Menor soporte para consultas complejas Puede haber problemas de consistencia en datos altamente distribuidos No existe un estándar para la implementación
Orientadas a objetos	Utilizan modelos de datos basados en objetos Permiten el almacenamiento de objetos y sus relaciones Soportan herencia y polimorfismo	Permiten la representación de objetos y relaciones complejas de forma natural Buena escalabilidad Integración con lenguajes de programación orientados a objetos	Falta de estándares en la implementación Requieren conocimientos técnicos para su configuración Problemas de rendimiento con grandes volúmenes de datos
Grafos	Almacenan datos en forma de nodos y relaciones Permiten la representación de relaciones complejas de forma natural Utilizan lenguajes de consulta específicos para grafos	Rendimiento rápido para operaciones de consulta de grafos complejas Flexibilidad para modelar relaciones complejas Integración con lenguajes de programación orientados a grafos	No son adecuados para operaciones que no involucren grafos Requieren conocimientos técnicos para su configuración No existen estándares ampliamente aceptados para la implementación

Fuente: Comparación de los diferentes paradigmas de las bases de datos.

Resultado del Análisis

Después de analizar el cuadro comparativo de las tecnologías de bases de datos, se puede concluir que la mejor opción para un proyecto de sistema de gestión de información académica es la tecnología de bases de datos relacionales.

Aunque las bases de datos NoSQL y Orientadas a objetos ofrecen ventajas como el rendimiento rápido y la capacidad de almacenar datos complejos de manera natural, la estructura de tablas y el modelo de datos basado en relaciones de las bases de datos relacionales se adaptan mejor a la naturaleza de los datos en un sistema de gestión de información académica.

Dentro de los tipos de las bases de datos relacionales se seleccionó la base de datos relacional PostgreSQL, ya que es la mejor opción debido a sus ventajas y capacidades específicas. Además, PostgreSQL utiliza SQL como lenguaje de consulta, lo que facilita su uso y manejo, tiene un mejor soporte para consultas complejas y ofrece una mayor eficiencia para trabajar con datos estructurados. También tiene una amplia comunidad de usuarios y desarrolladores que brindan soporte y recursos adicionales.

Tabla 2.

Cuadro Comparativo para Tecnología de Lenguajes de Programación

Lenguaje de Programación	Propiedades	Ventajas	Desventajas
Java	Lenguaje de programación orientado	Portabilidad entre diferentes sistemas	Requiere más recursos de memoria que otros

Lenguaje de Programación	Propiedades	Ventajas	Desventajas
	a objetos Compilado en bytecode para ser ejecutado en una JVM	operativos Amplia disponibilidad de bibliotecas y herramientas de	lenguajes Algunos desarrolladores pueden encontrarlo verboso
Lenguaje de programación multiplataforma		concurrentes	
Velocidad de ejecución más lenta en comparación con otros lenguajes Problemas de rendimiento en aplicaciones altamente concurrentes Menor capacidad para manejar tareas de bajo nivel en comparación con otros lenguajes	Python	Lenguaje de programación interpretado Fácil de aprender y utilizar Amplia biblioteca estándar Sintaxis clara y legible	Sintaxis clara y legible Gran comunidad de desarrolladores y amplia disponibilidad de bibliotecas y herramientas de terceros Versatilidad en diferentes ámbitos, desde aplicaciones web hasta ciencia de datos
Dependencia de bibliotecas y marcos de trabajo para tareas avanzadas Problemas de seguridad en aplicaciones web mal diseñadas Problemas de rendimiento en aplicaciones altamente concurrentes	JavaScript	Lenguaje de programación interpretado Ampliamente utilizado en el desarrollo web Compatible con todos los navegadores modernos Sintaxis clara y legible	Capacidad para construir aplicaciones interactivas y dinámicas en el navegador Gran comunidad de desarrolladores y amplia disponibilidad de bibliotecas y herramientas de terceros Compatible con todas las plataformas y navegadores modernos
Mayor complejidad de programación en comparación con otros lenguajes Poca seguridad en operaciones de bajo	C++	Lenguaje de programación de bajo nivel Compilado para ser ejecutado directamente en la máquina	Muy alto rendimiento en comparación con otros lenguajes Potencial de uso en sistemas operativos y hardware de bajo

Lenguaje de Programación	Propiedades	Ventajas	Desventajas
	Potencial de rendimiento muy alto Capacidad de utilizar punteros y operaciones de bajo nivel	nivel Capacidad de controlar la memoria de forma precisa	nivel en comparación con otros lenguajes Requiere más tiempo y esfuerzo para la programación y el mantenimiento
PHP	Lenguaje de programación interpretado Ampliamente utilizado en el desarrollo web Sintaxis clara y sencilla Gran comunidad de desarrolladores y amplia disponibilidad de bibliotecas y herramientas de terceros	Fácil de aprender y utilizar Gran disponibilidad de recursos y herramientas para el desarrollo web Capacidad para integrarse con bases de datos y otros sistemas Buena capacidad de escalabilidad	Rendimiento inferior en comparación con otros lenguajes de programación Poca seguridad en aplicaciones mal diseñadas Puede ser menos adecuado para aplicaciones de alto rendimiento

Fuente: Comparación de los lenguajes de programación con sus ventajas y desventajas

Resultado del Análisis

El cuadro presenta una comparación de los principales lenguajes de programación que se pueden utilizar en el desarrollo de un sistema de gestión de información académica. Según la información proporcionada, Java parece ser la mejor opción para este tipo de proyecto.

Java es un lenguaje de programación orientado a objetos que es compilado en bytecode y se ejecuta en una JVM (Java Virtual Machine). Esto significa que es altamente portable y puede ser utilizado en diferentes sistemas operativos sin necesidad de recompilar el código fuente.

Además, tiene una amplia biblioteca estándar y hay muchas bibliotecas y herramientas de terceros disponibles que pueden ser utilizadas para desarrollar el sistema.

Otra ventaja de Java es su seguridad y robustez, lo que lo hace adecuado para sistemas críticos como un sistema de gestión de información académica.

Tabla 3.

Cuadro Comparativo para Tecnología de Framework

Framework	Propiedades	Ventajas	Desventajas
Ruby on Rails	Framework de desarrollo web escrito en Ruby MVC (Model-View-Controller) como patrón de diseño Convenio sobre configuración (Convention over Configuration) Enfoque en la simplicidad y la productividad del desarrollador	Gran eficiencia en la creación de aplicaciones web y móviles Muy buena escalabilidad Gran cantidad de módulos y bibliotecas de terceros Mejora la seguridad y la integridad de la aplicación	No es adecuado para aplicaciones de bajo nivel o de rendimiento Menor control en algunos aspectos técnicos debido al enfoque en la simplicidad
Django	Framework de desarrollo web escrito en Python MVC (Model-View-Controller) como patrón de diseño Enfoque en la simplicidad y la productividad del desarrollador Baterías incluidas: una amplia gama de funcionalidades integradas en el framework	Gran eficiencia en la creación de aplicaciones web Excelente soporte para la creación de aplicaciones de gran tamaño Gran cantidad de módulos y bibliotecas de terceros Mejora la seguridad y la integridad de la aplicación	Menor flexibilidad en algunos aspectos técnicos debido al enfoque en la simplicidad No es adecuado para aplicaciones de bajo nivel o de rendimiento
AngularJS	Framework de desarrollo	Eficiencia en la	Menor flexibilidad en

Framework	Propiedades	Ventajas	Desventajas
	<p>web escrito en JavaScript</p> <p>MVVM (Model-View-ViewModel) como patrón de diseño</p> <p>Enfoque en la creación de aplicaciones web de una sola página (Single Page Applications)</p> <p>Uso extensivo de HTML y CSS para la creación de vistas</p>	<p>creación de aplicaciones web de una sola página</p> <p>Capacidad de personalización de las vistas</p> <p>Soporte para la creación de aplicaciones de gran tamaño</p> <p>Módulos y bibliotecas de terceros</p> <p>Mejora la seguridad y la integridad de la aplicación</p>	<p>algunos aspectos técnicos debido al enfoque en la simplicidad</p> <p>No es adecuado para aplicaciones de bajo nivel o de rendimiento</p> <p>Necesidad de aprender la sintaxis y las convenciones específicas de AngularJS</p>
Laravel	<p>Framework de desarrollo web escrito en PHP</p> <p>MVC (Model-View-Controller) como patrón de diseño</p> <p>Enfoque en la simplicidad y la productividad del desarrollador</p> <p>Convenio sobre configuración (Convention over Configuration)</p>	<p>Eficiencia en la creación de aplicaciones web</p> <p>Soporte para la creación de aplicaciones de gran tamaño</p> <p>Gran cantidad de módulos y bibliotecas de terceros</p> <p>Mejora la seguridad y la integridad de la aplicación</p>	<p>No es adecuado para aplicaciones de bajo nivel o de rendimiento</p> <p>Menor flexibilidad en algunos aspectos técnicos debido al enfoque en la simplicidad</p>
Spring Boot	<p>Framework de desarrollo de aplicaciones Java</p> <p>Enfoque en la simplicidad y la productividad del desarrollador</p> <p>Automatiza la configuración de aplicaciones</p> <p>Permite una fácil integración con otras herramientas y servicios</p>	<p>Gran eficiencia en la creación de aplicaciones Java</p> <p>Excelente soporte para la creación de aplicaciones de gran tamaño</p> <p>Gran cantidad de módulos y bibliotecas de terceros</p> <p>Mejora la seguridad y la integridad de la aplicación</p>	<p>Requiere conocimiento previo de Java</p> <p>Menor flexibilidad en algunos aspectos técnicos debido al enfoque en la simplicidad</p> <p>Puede requerir una curva de aprendizaje pronunciada para programadores novatos en Java</p>

Fuente: Cuadro comparativo con los frameworks backend más utilizados en el mercado

Resultado del Análisis del Cuadro Comparativo para Tecnología de Framework

Al evaluar los diferentes frameworks presentados en el cuadro, podemos concluir que Spring Boot es una excelente opción para un proyecto de sistema de gestión de información académica. Algunas de las ventajas de Spring Boot incluyen su enfoque en la simplicidad y la productividad del desarrollador, así como su capacidad para automatizar la configuración de aplicaciones y permitir una fácil integración con otras herramientas y servicios.

Además, Spring Boot es altamente eficiente en la creación de aplicaciones Java y cuenta con una gran cantidad de módulos y bibliotecas de terceros que pueden mejorar la seguridad y la integridad de la aplicación. También es capaz de soportar la creación de aplicaciones de gran tamaño, lo cual es importante para un sistema de gestión de información académica.

Tabla 4.

Cuadro Comparativo para Tecnología de Servidores Web

Servidor Web	Propiedades	Ventajas	Desventajas
Apache	Software libre y gratuito Compatible con múltiples sistemas operativos Altamente personalizable Escalabilidad	Fácil configuración y administración Gran comunidad de usuarios y desarrolladores Estabilidad y seguridad comprobadas	Requiere configuración avanzada para un rendimiento óptimo Mayor consumo de recursos en comparación con otros servidores web Limitaciones en la concurrencia y en el

Servidor Web	Propiedades	Ventajas	Desventajas
	horizontal	Compatible con PHP, Perl, Python y otros lenguajes web populares	manejo de peticiones
Nginx	Software libre y gratuito Diseñado para manejar grandes volúmenes de tráfico web Escalabilidad horizontal Bajo consumo de recursos	Mayor rendimiento en comparación con Apache Manejo de peticiones concurrentes más eficiente Compatible con múltiples lenguajes web populares Fácil configuración y administración	Menos personalizable en comparación con Apache Configuración más avanzada para un rendimiento óptimo No compatible con algunos módulos de Apache
Microsoft IIS	Incluido en el paquete de software de Microsoft Compatible con el ecosistema de Microsoft Fácil integración con otras tecnologías de Microsoft Gestión centralizada de servidores web	Integración fácil con otros productos de Microsoft Fácil administración y configuración Soporte para ASP.NET y otras tecnologías de Microsoft Buena escalabilidad horizontal	Limitaciones en la compatibilidad con otras tecnologías no Microsoft Menos personalizable en comparación con Apache y Nginx Mayor consumo de recursos en comparación con otros servidores web
Tomcat	Servidor web y contenedor de servlets Java Compatible con múltiples sistemas operativos Soporte para Java EE Escalabilidad horizontal	Alto rendimiento para aplicaciones Java Fácil integración con otras tecnologías Java Soporte para múltiples lenguajes web populares Alta escalabilidad horizontal	Menos personalizable en comparación con Apache y Nginx Requiere conocimiento previo de Java No es tan rápido en el manejo de peticiones concurrentes como Nginx

Fuente: Cuadro comparativo de los servidores web más populares del mercado

Resultado del Análisis del Cuadro Comparativo para Tecnología de Servidores Web

Después de analizar las propiedades, ventajas y desventajas de los servidores web presentados en el cuadro, se puede concluir que Tomcat es la mejor opción para un proyecto de sistema de gestión de información académica.

En primer lugar, Tomcat es compatible con múltiples sistemas operativos, lo que lo hace muy versátil y fácil de implementar en diferentes entornos de servidor. Además, como servidor web y contenedor de servlets Java, Tomcat ofrece un alto rendimiento para aplicaciones Java, lo que es una ventaja significativa para un sistema de gestión de información académica que probablemente incluirá una gran cantidad de componentes Java.

Otra ventaja importante de Tomcat es su escalabilidad horizontal, lo que significa que puede crecer y adaptarse a medida que aumenta la cantidad de usuarios y la cantidad de información que maneja el sistema. Además, Tomcat es compatible con múltiples lenguajes web populares, lo que aumenta su flexibilidad y capacidad de integración con otras tecnologías.

Tabla 5.

Cuadro Comparativo para Tecnología Frontend

Frontend	Propiedades	Ventajas	Desventajas
HTML	Lenguaje de marcado utilizado para la creación de páginas web	Compatible con todos los navegadores web Accesibilidad para dispositivos de asistencia	No es un lenguaje de programación Solo se utiliza para la estructura y el contenido de una página web
CSS	Lenguaje de hojas de estilo utilizado para la presentación visual de una página web	Permite la separación de la estructura y la presentación visual de una página web	Algunas funcionalidades pueden ser difíciles de entender para los principiantes

Frontend	Propiedades	Ventajas	Desventajas
		Facilita la gestión y el mantenimiento de un sitio web Compatible con todos los navegadores web	El diseño responsivo puede requerir mucho tiempo y esfuerzo
JavaScript	Lenguaje de programación utilizado para la creación de interacciones dinámicas y efectos visuales en una página web	Se usa del lado del cliente como en el del servidor Compatible con todos los navegadores web Librerías y frameworks disponibles	Compatibilidad con diferentes navegadores puede requerir pruebas y ajustes adicionales
React	Biblioteca de JavaScript utilizada para la creación de interfaces de usuario	Fácil de usar y aprender Modularidad que facilita la reutilización de componentes Gran cantidad de documentación y recursos disponibles	Puede ser difícil configurar la primera vez Componentes difíciles de personalizar
Angular	Framework de JavaScript utilizado para la creación de aplicaciones web	Gran cantidad de herramientas y características integradas Comunidad de desarrolladores Escalabilidad Modularidad	Curva de aprendizaje pronunciada Cantidad considerable de código puede generar una aplicación más lenta
Vue	Framework de JavaScript utilizado para la creación de interfaces de usuario	Bajo acoplamiento Alta cohesión Pequeño tamaño y alto rendimiento	Algunas funcionalidades avanzadas pueden requerir conocimientos de JavaScript Comunidad de desarrollo menor que Angular y React

Fuente: Cuadro comparativo de los frameworks frontend utilizados en el mundo empresarial

Resultado del Análisis del Cuadro Comparativo para Tecnología Frontend

Después de analizar las propiedades, ventajas y desventajas de cada una de las tecnologías presentadas, se puede concluir que la tecnología Frontend Angular es la mejor opción para un proyecto de sistema de gestión de información académica.

Angular proporciona una estructura clara y modular para el desarrollo de aplicaciones web, lo que facilita la organización y mantenimiento del código. Utiliza TypeScript como lenguaje de programación, lo que agrega tipado estático y características avanzadas al desarrollo, mejorando la calidad y robustez del sistema.

Además, Angular ofrece una amplia gama de herramientas y bibliotecas, lo que permite el desarrollo eficiente de componentes reutilizables y la implementación de características avanzadas, como enrutamiento, manejo de formularios y comunicación con servicios web.

La separación clara de responsabilidades en Angular, a través del uso de componentes, directivas y servicios, permite un desarrollo colaborativo y escalable. Además, Angular cuenta con una comunidad activa y una amplia documentación, lo que facilita el aprendizaje y la resolución de problemas.

En resumen, Angular es una tecnología frontend poderosa y versátil que proporciona una estructura sólida, características avanzadas y una amplia comunidad de apoyo, lo que la convierte en la mejor opción para un proyecto de sistema de gestión de información académica.

Tabla 6.

Cuadro Comparativo para Tecnología de Sistemas Operativos

Sistema Operativo	Propiedades	Ventajas	Desventajas
Windows	Sistema operativo propietario de Microsoft	Ampliamente utilizado Gran cantidad de software disponible Interfaz gráfica de usuario intuitiva	Costo elevado Problemas de seguridad y vulnerabilidades Necesidad de hardware de alta especificación para un rendimiento óptimo
macOS	Sistema operativo propietario de Apple	Interfaz de usuario intuitiva Alto rendimiento en hardware de Apple Seguridad y estabilidad	Limitado a hardware de Apple Costo elevado Menos software disponible que Windows
Linux	Sistema operativo de código abierto	Personalizable Gratuito Seguridad y estabilidad Amplia variedad de distribuciones	Curva de aprendizaje pronunciada para los principiantes Algunos programas y hardware pueden no ser compatibles Falta de soporte para ciertos juegos y software de diseño
Unix	Sistema operativo de código abierto	Estabilidad Seguridad Flexibilidad Escalabilidad	Costo elevado Complejidad de configuración y mantenimiento Menor cantidad de software disponible que Windows o Linux
Android	Sistema operativo de código abierto para dispositivos móviles	Ampliamente utilizado Gran cantidad de aplicaciones disponibles Personalizable	Vulnerabilidades de seguridad Fragmentación en la versión de software y hardware Fabricantes no proporcionan actualizaciones regulares
iOS	Sistema operativo propietario para dispositivos móviles de Apple	Interfaz de usuario intuitiva Alto rendimiento y seguridad Gran cantidad de aplicaciones disponibles	Limitado a hardware de Apple Costo elevado Menos personalizable que Android

Tecnología de Seguridad	Propiedades	Ventajas	Desventajas
Firewall	Protege la red contra ataques externos	Control de acceso Detección y prevención de intrusiones	Puede ser vulnerable a ataques de spoofing No protege contra amenazas internas Puede afectar el rendimiento de la red
Antivirus	Protege contra malware y virus	Detección y eliminación de malware Actualizaciones frecuentes de la base de datos	No puede detectar malware nuevo o desconocido Pueden reducir el rendimiento del sistema
VPN	Permite conexiones seguras y privadas a través de una red pública	Encriptación de datos Seguridad y privacidad Acceso remoto	Puede disminuir la velocidad de la conexión Puede haber conflictos con otros programas de seguridad Puede ser difícil de configurar
Autenticación de dos factores	Proporciona una capa adicional de seguridad al inicio de sesión	Mayor seguridad Prevención de acceso no autorizado	Incómodo para los usuarios Requiere hardware adicional No garantiza una protección completa
Encriptación de datos	Protege la información confidencial mediante la codificación de datos	Mayor seguridad y privacidad Cumplimiento de regulaciones y leyes	Puede disminuir la velocidad de la conexión Puede ser difícil de configurar y mantener Vulnerabilidades en la implementación

Fuente: Datos de los sistemas de seguridad que pueden ser utilizados en el desarrollo de software

Resultado del Análisis del Cuadro Comparativo para Tecnología de Seguridad

En un proyecto de sistema de gestión de información académica, es importante tener en cuenta la seguridad de la información para proteger los datos de los estudiantes y profesores. Cada tecnología de seguridad tiene sus propias ventajas y desventajas.

El firewall es una tecnología de seguridad que protege la red contra ataques externos, pero puede ser vulnerable a ataques de spoofing y no protege contra amenazas internas.

El antivirus protege contra malware y virus, pero no puede detectar malware nuevo o desconocido y puede reducir el rendimiento del sistema.

La VPN permite conexiones seguras y privadas a través de una red pública, pero puede disminuir la velocidad de la conexión y puede ser difícil de configurar.

La autenticación de dos factores proporciona una capa adicional de seguridad al inicio de sesión, pero puede ser incómodo para los usuarios y requiere hardware adicional.

La encriptación de datos protege la información confidencial mediante la codificación de datos, pero puede disminuir la velocidad de la conexión y puede ser difícil de configurar y mantener.

En conclusión, es importante utilizar una combinación de tecnologías de seguridad para proteger adecuadamente la información académica. Se puede utilizar un firewall junto con un antivirus para proteger contra amenazas externas e internas. La autenticación de dos factores y la encriptación de datos pueden proporcionar una capa adicional de seguridad. Es importante evaluar cuidadosamente las ventajas y desventajas de cada tecnología de seguridad y seleccionar las que sean más apropiadas para las necesidades específicas del proyecto.

Tabla 8.*Cuadro Comparativo para Arquitecturas de Software*

Arquitectura	Propiedades	Ventajas	Desventajas
Basada en Microservicios	Descompone la aplicación en servicios pequeños y autónomos que se ejecutan en procesos separados y se comunican mediante protocolos de red ligeros	Mejora la escalabilidad y la resiliencia de la aplicación Permite la implementación de una lógica de negocio distribuida Facilita la implementación de pruebas unitarias y la gestión de versiones	Puede aumentar la complejidad de la aplicación y la gestión de los servicios Requiere una mayor cantidad de tiempo y recursos para la implementación y el mantenimiento Puede tener problemas de coordinación entre los servicios si no se diseñan adecuadamente
Orientada a servicios	Los servicios se diseñan para ser independientes de la plataforma El lenguaje y la implementación, se comunican a través de protocolos estándar	Fácil de mantener a medida que el software crece Permite utilizar diferentes tecnologías y lenguajes de programación	Mayor complejidad que la arquitectura monolítica Dificultades en la coordinación entre los diferentes servicios
Basada en eventos	El software se basa en eventos y las acciones se activan como respuesta a ellos	Fácil de escalar Permite manejar grandes cantidades de datos en tiempo real	Mayor complejidad que la arquitectura monolítica Dificultades en la gestión de eventos y la coordinación de acciones
Basada en contenedores	Los contenedores son unidades de software autónomas y portátiles que se ejecutan en cualquier entorno que tenga el mismo sistema operativo subyacente	Fácil de desplegar Permite la portabilidad del software Permite utilizar diferentes tecnologías y lenguajes de programación	Mayor complejidad que la arquitectura monolítica Dificultades en la gestión de contenedores y la coordinación entre ellos

Arquitectura	Propiedades	Ventajas	Desventajas
Arquitectura basada en capas	Organiza la aplicación en capas horizontales y separadas, donde cada capa representa un conjunto de funcionalidades relacionadas	Mejora el modularidad y la escalabilidad de la aplicación Permite el reemplazo de una capa sin afectar a las demás Facilita la implementación de pruebas unitarias	Puede introducir complejidad en la aplicación si se añaden demasiadas capas Puede causar cuellos de botella en el rendimiento si se sobrecargan las capas intermedias Puede requerir una mayor cantidad de código para implementar una funcionalidad debido a la separación en capas
Arquitectura cliente-servidor	Es una arquitectura de software que divide la aplicación en dos partes: el cliente y el servidor El cliente solicita servicios o recursos al servidor, y éste responde a las solicitudes	Facilita el desarrollo y la gestión de aplicaciones escalables Mejora la eficiencia y la seguridad al centralizar el procesamiento y los recursos Permite la implementación de una lógica de negocio compartida	Requiere una mayor cantidad de tiempo y recursos para la implementación y el mantenimiento Puede aumentar la complejidad de la aplicación y la dependencia del servidor Puede tener problemas de rendimiento si se sobrecarga el servidor o la red

Fuente: Las arquitecturas de software más relevantes en el desarrollo de software

Resultado del Análisis del Cuadro Comparativo para Arquitecturas de Software

En un proyecto de sistema de gestión de información académica, la arquitectura basada en capas parece ser la mejor opción. Esta arquitectura mejora la modularidad y escalabilidad de la aplicación, permitiendo el reemplazo de una capa sin afectar a las demás. Además, facilita la

implementación de pruebas unitarias, lo cual es importante en un sistema académico para garantizar la calidad de los datos y la seguridad de la información.

La arquitectura basada en microservicios también puede mejorar la escalabilidad y la resiliencia de la aplicación, pero requiere una mayor cantidad de tiempo y recursos para la implementación y el mantenimiento, lo cual puede ser una limitación en un proyecto académico con recursos limitados.

La arquitectura orientada a servicios y la basada en eventos tienen ventajas en términos de escalabilidad y capacidad para manejar grandes cantidades de datos en tiempo real, pero también introducen mayor complejidad y dificultades en la coordinación entre los diferentes servicios o eventos.

La arquitectura basada en contenedores es útil para la portabilidad del software y la posibilidad de utilizar diferentes tecnologías y lenguajes de programación, pero también puede introducir mayor complejidad en la gestión de contenedores y la coordinación entre ellos.

La arquitectura cliente-servidor puede mejorar la eficiencia y la seguridad al centralizar el procesamiento y los recursos, pero requiere una mayor cantidad de tiempo y recursos para la implementación y el mantenimiento, lo cual puede ser una limitación en un proyecto académico con recursos limitados.

4.2.3. Selección de las tecnologías y arquitecturas que se adaptan al contexto del Genesis SIA versión 4.0.

Teniendo en cuenta el análisis realizado en los cuadros comparativos, se puede concluir que:

En cuanto a la elección de la base de datos relacional PostgreSQL, es importante destacar que esta es una de las bases de datos más populares y utilizadas en el mundo de la programación debido a su flexibilidad, estabilidad y capacidad de manejar grandes cantidades de datos. Además, su modelo de datos basado en relaciones permite establecer relaciones entre diferentes tablas, lo que resulta muy útil en un sistema de gestión de información académica, donde existen múltiples relaciones entre las diferentes entidades, como estudiantes, profesores, cursos, calificaciones, entre otros.

Por otro lado, el lenguaje de programación Java es muy popular en la industria del software debido a su portabilidad, es decir, la capacidad de ser ejecutado en diferentes plataformas sin necesidad de modificaciones. Además, Java cuenta con una gran cantidad de bibliotecas y herramientas disponibles que permiten desarrollar aplicaciones de manera eficiente y segura. También es un lenguaje muy robusto, lo que significa que es resistente a errores y puede manejar grandes volúmenes de datos sin problemas.

El framework Spring Boot es una herramienta de programación de aplicaciones web que permite a los desarrolladores crear aplicaciones de manera rápida y sencilla, sin necesidad de realizar una configuración manual compleja. Spring Boot se enfoca en la simplicidad y la

productividad del desarrollador, lo que significa que los desarrolladores pueden enfocarse en la lógica de negocio en lugar de la configuración de la aplicación. Además, Spring Boot permite integrar fácilmente otras herramientas y servicios, lo que lo hace ideal para proyectos complejos como un sistema de gestión de información académica.

El servidor web Tomcat es uno de los servidores web más populares y utilizados en el mundo de la programación. Es un servidor web de código abierto que permite a los desarrolladores ejecutar aplicaciones web de manera eficiente y segura. Tomcat es compatible con muchos sistemas operativos, incluyendo Windows, lo que lo hace ideal para proyectos de cualquier tamaño.

En cuanto al frontend se utilizó el framework Angular. Angular es un framework de

de programación. Proporciona una estructura sólida y escalable para construir aplicaciones web. Angular combina HTML, CSS y JavaScript para crear una experiencia de usuario interactiva y dinámica. Con Angular, se puede separar la lógica de la aplicación en componentes reutilizables y gestionar fácilmente la comunicación entre ellos. Además, ofrece potentes herramientas para el enrutamiento, la manipulación del DOM y la gestión del estado de la aplicación. Angular es ampliamente utilizado en el desarrollo de aplicaciones empresariales y aplicaciones web de gran escala debido a su capacidad para crear interfaces de usuario complejas y eficientes.

desarrollo web que utiliza TypeScript como lenguaje de programación. Proporciona una estructura sólida y escalable para construir aplicaciones web. Angular combina HTML, CSS y JavaScript para crear una experiencia de usuario interactiva y dinámica. Con Angular, se puede separar la lógica de la aplicación en componentes reutilizables y gestionar fácilmente la comunicación entre ellos. Además, ofrece potentes herramientas para el enrutamiento, la manipulación del DOM y la gestión del estado de la aplicación. Angular es ampliamente utilizado en el desarrollo de aplicaciones empresariales y aplicaciones web de gran escala debido a su capacidad para crear interfaces de usuario complejas y eficientes.

arquitectura hexagonal en un sistema de información académica ofrece beneficios como una clara separación de responsabilidades, modularidad,

Por último, la implementación de la arquitectura hexagonal en un sistema de información académica ofrece beneficios como una clara separación de responsabilidades, modularidad,

flexibilidad, facilidad para realizar pruebas unitarias y reutilización de componentes. Esta arquitectura permite un mantenimiento y evolución más sencillos, así como una mayor adaptabilidad a medida que cambian los requisitos. Además, facilita la integración con otros sistemas y la adopción de nuevas tecnologías, proporcionando una base sólida y flexible para el sistema de información académica.

En conclusión, la elección de PostgreSQL como base de datos relacional, Java como lenguaje de programación, Spring Boot como framework, Tomcat como servidor y una arquitectura basada en capas, proporciona un sistema de gestión de información académica escalable, eficiente, seguro y confiable.

4.3. Tercer objetivo

Diseñar los componentes arquitectónicos a partir del análisis y caracterización previamente realizada proponiendo así la estructura del sistema.

Diseñar componentes de arquitectura de software del Genesis SIA versión 4.0.

Crear el diseño de interfaces UI del sistema de información académica Genesis SIA versión 4.0

El tercer objetivo de diseñar los componentes arquitectónicos y proponer la estructura del sistema se logró a través de un enfoque metodológico que involucró el análisis y caracterización

previa del entorno y los requerimientos del sistema Genesis SIA. Para ello, se llevó a cabo una exhaustiva revisión de las necesidades de las instituciones educativas y los usuarios finales.

En primer lugar, se realizó un análisis detallado de los procesos académicos y administrativos existentes en las instituciones educativas, identificando las diferentes áreas y funciones que debían ser consideradas en el diseño del sistema. Esto permitió comprender las interacciones y flujos de información necesarios para un funcionamiento eficiente.

A continuación, se realizó una caracterización de los requisitos y expectativas de los usuarios, a través de encuestas, entrevistas y sesiones de retroalimentación. Se recopilaron y analizaron los comentarios y sugerencias de los usuarios para comprender sus necesidades específicas y sus preferencias en términos de funcionalidades y experiencia de usuario.

Con base en esta información, se procedió a diseñar los componentes arquitectónicos del sistema Genesis SIA. Se definió una estructura lógica y modular que permitiera una fácil escalabilidad y adaptabilidad a futuras actualizaciones y requisitos adicionales. Se consideraron las mejores prácticas en términos de arquitectura de software y se establecieron estándares de diseño para garantizar la coherencia y la eficiencia en el desarrollo del sistema.

4.3.1. Diseño componentes de arquitectura de software del Genesis SIA versión 4.0.

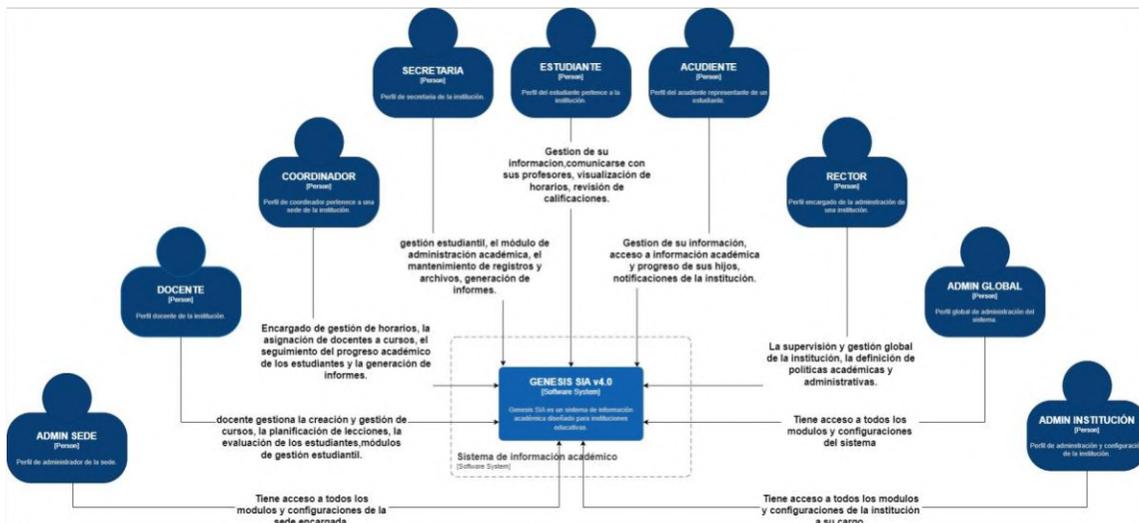
En el diagrama de contexto, se representa el sistema en el centro como un único bloque rectangular, rodeado por sus actores externos o sistemas con los que interactúa. Los actores

pueden ser usuarios, otros sistemas o servicios externos. El objetivo principal de este diagrama es mostrar las relaciones de alto nivel entre el sistema y su entorno, sin entrar en detalles internos de implementación.

El diagrama de contexto ayuda a visualizar los límites del sistema y a comprender su interacción con los actores externos. Proporciona una visión clara de las interfaces y los puntos de entrada y salida del sistema, lo que facilita la comunicación con los stakeholders y el entendimiento general del sistema.

Figura 19.

Diagrama de contexto del Genesis SIA v4.0



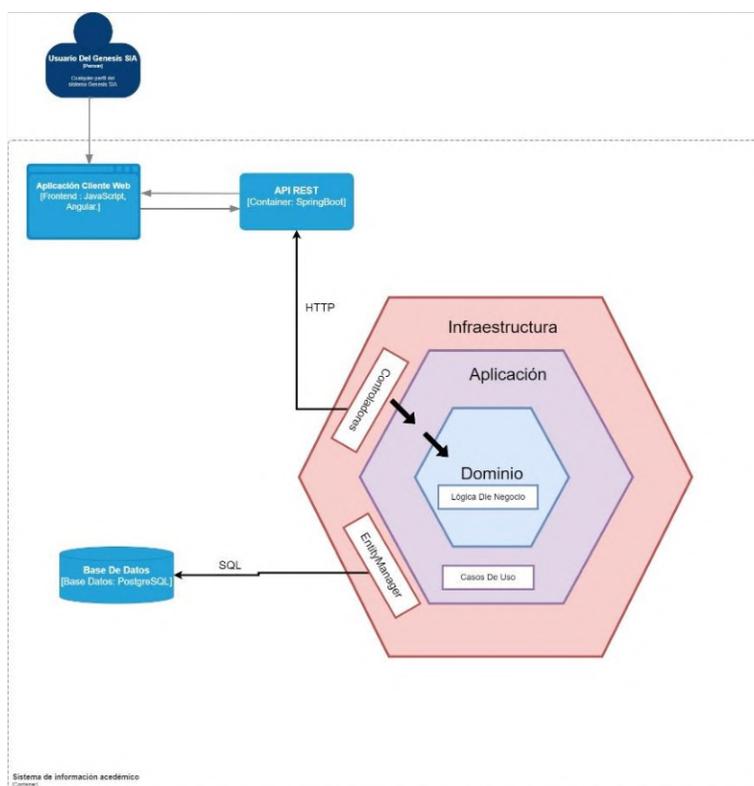
Fuente: Este diagrama nos permite tener una vista general de sistema de información académico y los perfiles que interactúan con el sistema.

El propósito principal del diagrama de contenedores es proporcionar una visión general de la infraestructura y los entornos de ejecución del sistema, separando claramente los componentes internos de la forma en que se despliegan y se relacionan entre sí. Esto facilita la comprensión de las dependencias y la planificación del despliegue del sistema.

El diagrama de contenedores en el modelo C4 es una herramienta efectiva para visualizar y comunicar la infraestructura y los entornos de ejecución de un sistema de software, mostrando los contenedores y sus relaciones, lo que ayuda en el diseño, la planificación y el análisis de la arquitectura.

Figura 20.

Diagrama de contenedores de Genesis SIA v4.



Fuente: Este diagrama nos permite tener una vista general de sistema de información

démico.

Teniendo en cuenta lo anterior y por medio de modelo c4 para diagramar arquitecturas se presenta la propuesta de diseño arquitectura para el sistema de información académico Genesis SIA v4.

La arquitectura del sistema está diseñada de manera robusta y escalable, con una combinación de tecnologías backend y frontend que trabajan en conjunto para ofrecer un funcionamiento eficiente. Cada componente está cuidadosamente diseñado para proporcionar funcionalidades específicas y garantizar la integridad y seguridad de los datos. En conjunto, la arquitectura y los componentes del sistema Genesis SIA v4 brindan una solución completa y confiable para la gestión de información académica. La arquitectura elegida para el desarrollo del software es la arquitectura hexagonal, también conocida como arquitectura de puertos y adaptadores. Esta arquitectura se basa en el principio de separar el núcleo de la lógica de negocio del sistema de sus interfaces externas y tecnologías específicas. En la arquitectura hexagonal, el núcleo de la aplicación se encuentra en el centro, rodeado por puertos y adaptadores que permiten la comunicación con el mundo exterior.

Los puertos definen las interfaces a través de las cuales el sistema interactúa con actores externos, como usuarios, sistemas externos o servicios web. Los adaptadores son responsables de implementar dichas interfaces y permitir la conexión entre el sistema y los diferentes componentes externos.

Esta arquitectura proporciona flexibilidad, modularidad y facilita la incorporación de cambios, ya que los componentes externos pueden ser fácilmente reemplazados o modificados sin afectar el núcleo de la aplicación. La arquitectura hexagonal es especialmente adecuada para sistemas complejos y evolutivos, donde la separación de preocupaciones y la interoperabilidad son fundamentales.

Dentro del diagrama de contenedores podemos observar la implementación de la arquitectura hexagonal la cual cuenta con las siguientes capas:

Capa de dominio: Es el núcleo de la aplicación y contiene la lógica empresarial y las reglas del dominio. No depende de ninguna capa externa y define las entidades, los objetos de valor y los casos de uso principales. Esta capa no contiene referencias directas a ninguna otra capa.

Capa de aplicaciones: Esta capa contiene los casos de uso de la aplicación y orquesta las interacciones entre la capa de dominio y las interfaces externas. Se encarga de recibir las solicitudes de entrada y traducirlas en comandos y consultas para la capa de dominio. También maneja las respuestas y las adapta para enviarlas de vuelta a las interfaces externas.

Capa de interfaces: Aquí se encuentran las interfaces externas, como la interfaz de usuario, las APIs, los servicios web u otras aplicaciones externas. Esta capa se comunica con la capa de aplicaciones para enviar las solicitudes y recibir las respuestas. Las interfaces se

implementan a través de adaptadores, que convierten los datos de las interfaces externas en un formato comprensible para la capa de aplicaciones y viceversa.

Capa de infraestructura: Esta capa contiene los detalles de implementación y las tecnologías específicas utilizadas en la aplicación. Incluye la implementación de bases de datos, servicios externos, frameworks, etc. La capa de infraestructura proporciona las implementaciones concretas de los puertos definidos en la capa de dominio y actúa como adaptadores para conectar con servicios externos.

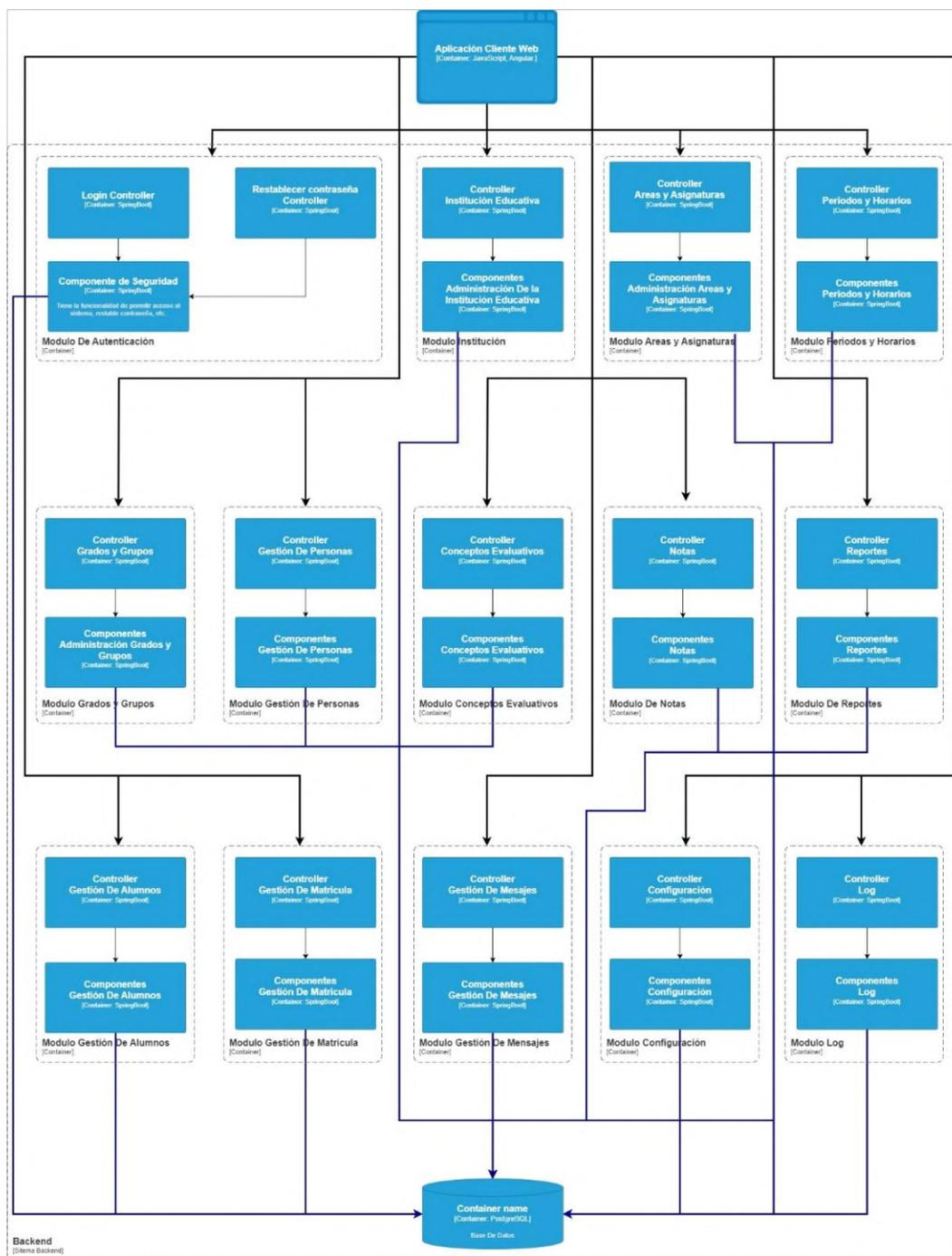
Estas serán las capas utilizadas en el nuevo sistema en el cual cada módulo del sistema de información académico tendrá esta distribución de capas con el propósito de tener bajo acoplamiento y alta cohesión entre los módulos del sistema que permitan tener un sistema abierto a la extensión y de fácil mantenimiento al tener las responsabilidades bien definidas dentro del software.

El objetivo del diagrama de componentes es proporcionar una vista de alto nivel de la estructura interna del sistema y cómo los componentes interactúan entre sí. También puede mostrar interfaces y contratos entre componentes, permitiendo comprender las responsabilidades y las interacciones clave en el sistema.

Además, el diagrama de componentes puede ayudar a identificar puntos críticos de la arquitectura, como componentes centrales, componentes reutilizables y componentes que podrían requerir una mayor modularización.

Figura 21.

Diagrama de componentes



Fuente: Este diagrama nos muestra los diferentes módulos que conforman el sistema.

El diagrama de componentes muestra todos los módulos que conforman el sistema de información académica. Estos módulos incluyen el módulo de gestión de estudiantes, que se encarga de manejar el registro de los estudiantes, sus datos personales, el control de asistencia y el seguimiento de su progreso académico. Otro módulo importante es el módulo de gestión de profesores, que permite administrar la información de los docentes, asignar horarios, controlar la carga académica y gestionar sus evaluaciones. Además, el sistema cuenta con un módulo de gestión de cursos, que facilita la creación y administración de los planes de estudio, la asignación de créditos y la programación de clases. Otros módulos destacados son el módulo de gestión de calificaciones, el cual permite el registro y cálculo de las notas de los estudiantes, y el módulo de gestión de horarios, que facilita la programación y asignación de horarios de clases. Estos son solo algunos de los módulos presentes en el sistema de información académica, cada uno desempeñando un papel fundamental en la gestión eficiente y precisa de la información académica.

4.3.2. Diseños de interfaces UI del sistema de información académica Genesis SIA versión

4.0

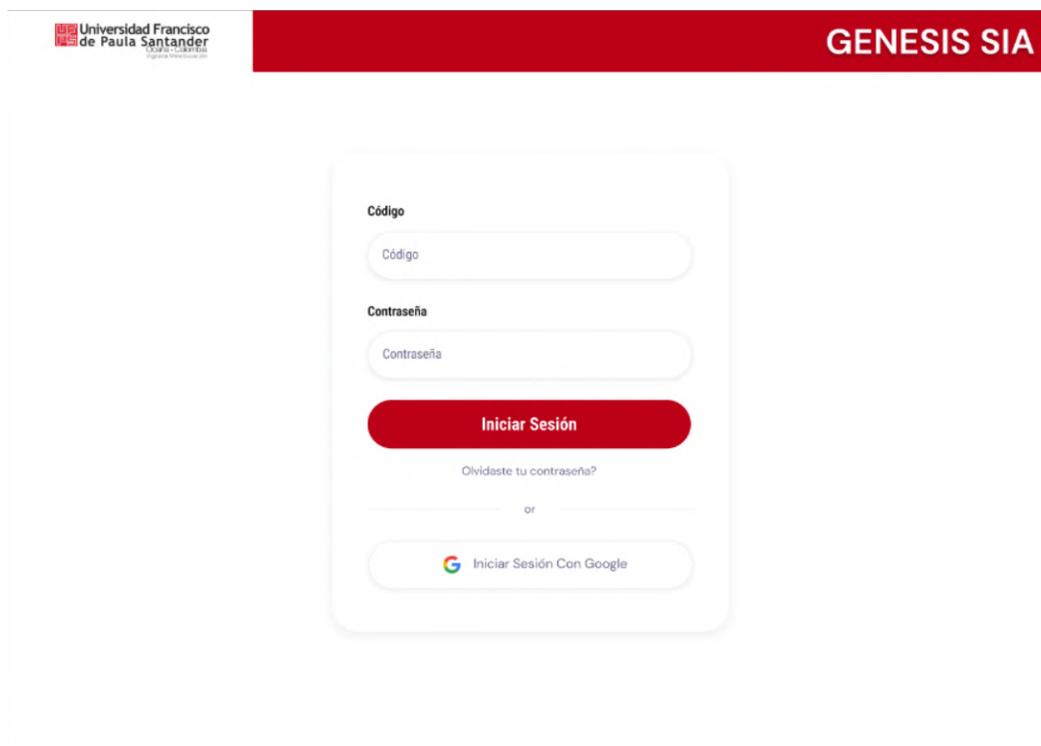
Con el propósito de modernizar y mejorar la experiencia de los usuarios del sistema Genesis SIA v4, se han diseñado las interfaces de usuario (UI) utilizando Figma. Esta herramienta de diseño colaborativa ha permitido crear interfaces innovadoras y atractivas, con una disposición intuitiva de los elementos, colores vibrantes y una tipografía legible. El uso de Figma ha posibilitado una mayor agilidad en el proceso de diseño, así como una colaboración fluida entre los miembros del equipo. Como resultado, el nuevo sistema Genesis SIA v4 ofrece

una experiencia de usuario actualizada y mejorada, que se adapta a las necesidades y expectativas de los usuarios modernos.

A continuación, se presenta los diseños del prototipo del Genesis SIA v4:

Figura 22.

Interface de login

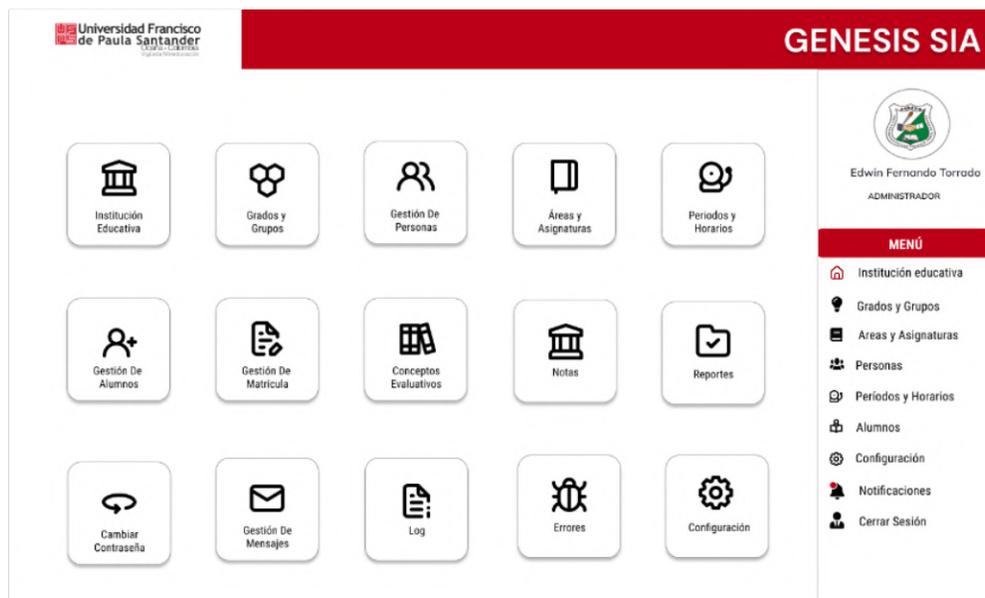


The image shows a login interface for the Genesis SIA v4 system. At the top left, there is a logo for Universidad Francisco de Paula Santander. At the top right, the text "GENESIS SIA" is displayed in white on a red background. The main content area is a white rounded rectangle containing the following elements: a "Código" label above a text input field; a "Contraseña" label above a text input field; a red "Iniciar Sesión" button; a link "Olvidaste tu contraseña?"; the word "or" centered below a horizontal line; and a button with the Google logo and the text "Iniciar Sesión Con Google".

Fuente: Interface de login del sistema Genesis SIA v4.0

Figura 23.

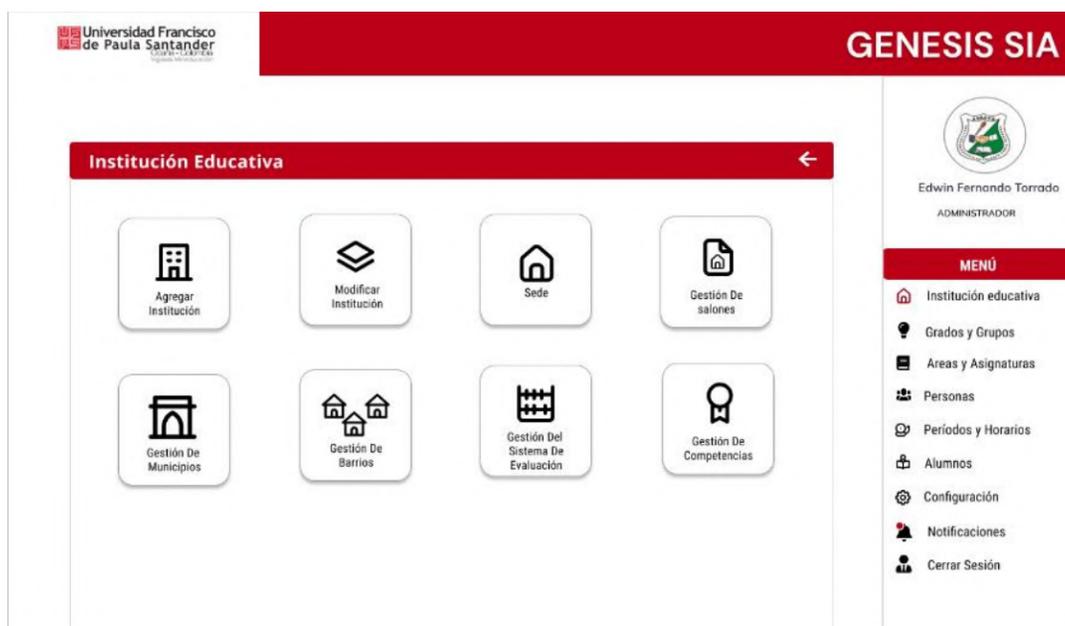
Interface Módulos



Fuente: Interface donde se encuentran los diferentes módulos del Genesis SIA v4.0

Figura 24.

Módulo Institución educativa



Fuente: Interface del módulo institución educativa con sus funcionalidades

Figura 25.

Interface agregar institución educativa

The screenshot shows the 'Agregar Institución Educativa' form in the GENESIS SIA system. The form is titled 'Institución Educativa' and has a red header with the text 'AGREGAR INSTITUCIÓN EDUCATIVA'. The form contains the following fields:

- CODIGO DANE *
- NOMBRE DE LA INSTITUCIÓN *
- DEPARTAMENTO *
- CIUDAD *
- DIRECCIÓN *
- TELÉFONO *
- EMAIL *
- REGISTRO SECRETARÍA DE EDUCACIÓN *
- DECRETO DE CREACIÓN *
- NATURALEZA *
- NIT *
- RESOLUCIÓN DE APROBACIÓN *
- ESPECIALIDAD *
- GENERO *
- CALENDARIO *
- METODOLOGÍA *
- TESORERO *
- RECTOR *
- SECRETARÍA *

A red 'GUARDAR' button is located at the bottom of the form. The right sidebar shows the user 'Edwin Fernando Torrado ADMINISTRADOR' and a 'MENÚ' with options: Institución educativa, Grados y Grupos, Areas y Asignaturas, Personas, Períodos y Horarios, Alumnos, Configuración, Notificaciones, and Cerrar Sesión.

Fuente: Formulario para agregar una institución educativa

Figura 26.

Interface de gestión de sede

The screenshot shows the 'Gestión de Sede' interface in the GENESIS SIA system. The interface has a red header with the text 'GESTIÓN DE SEDE'. Below the header is a search bar with the text 'Sede' and a 'NUEVA SEDE' button. The main content is a table with the following columns:

CODIGO DANE	NOMBRE DE SEDE	DIRECCIÓN	TELÉFONO	COORDINADOR	ESPECIALIDAD
123456789	ALGODONAL	VIA ACOLBURE, OCCARA	3107795441	JUAN PABLO BAYONA	ACADEMICO

The right sidebar shows the user 'Edwin Fernando Torrado ADMINISTRADOR' and a 'MENÚ' with options: Institución educativa, Grados y Grupos, Areas y Asignaturas, Personas, Períodos y Horarios, Alumnos, Configuración, Notificaciones, and Cerrar Sesión.

Fuente: Interface que permite administrar las sedes de la institución educativa

Figura 27.

Interface agregar sede

Universidad Francisco de Paula Santander
GENESIS SIA

Gestión de Sede

AGREGAR SEDE

CÓDIGO DANE * NOMBRE DE LA SEDE *
TELÉFONO * DIRECCIÓN *
ESPECIALIDAD * COORDINADOR *

GUARDAR

Edwin Fernando Torrado
ADMINISTRADOR

MENÚ

- Institución educativa
- Grados y Grupos
- Areas y Asignaturas
- Personas
- Periodos y Horarios
- Alumnos
- Configuración
- Notificaciones
- Cerrar Sesión

Fuente: Interface que permitirá agregar una nueva sede en la institución

Figura 28.

Interface gestión de salones

Universidad Francisco de Paula Santander
GENESIS SIA

Gestión de Salones

AGREGAR SALÓN

NOMBRE DEL SALÓN * CUPO MÁXIMO *

GUARDAR

Sede

Filtros BUSCAR SALONES

	NOMBRE DE SALÓN	CUPO MÁXIMO	
<input type="checkbox"/>	SALÓN N°1	50	⋮

Edwin Fernando Torrado
ADMINISTRADOR

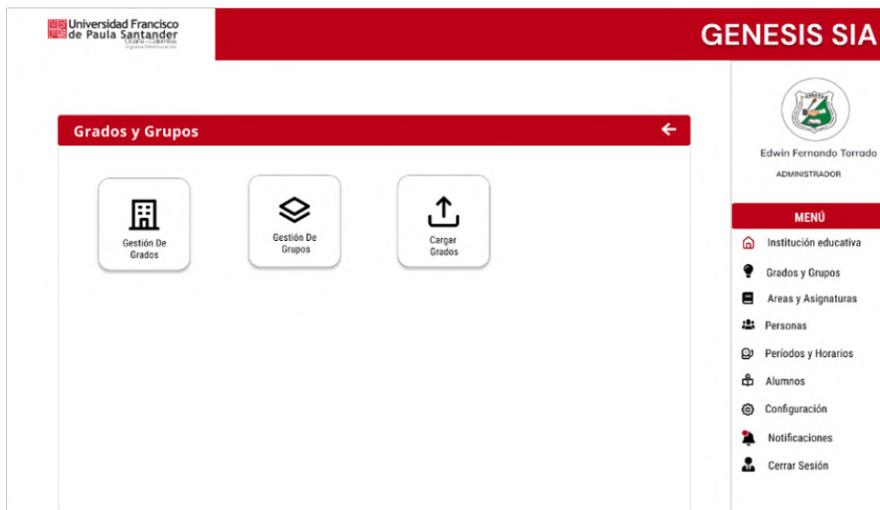
MENÚ

- Institución educativa
- Grados y Grupos
- Areas y Asignaturas
- Personas
- Periodos y Horarios
- Alumnos
- Configuración
- Notificaciones
- Cerrar Sesión

Fuente: Interface donde administraran los salones y sus características

Figura 29.

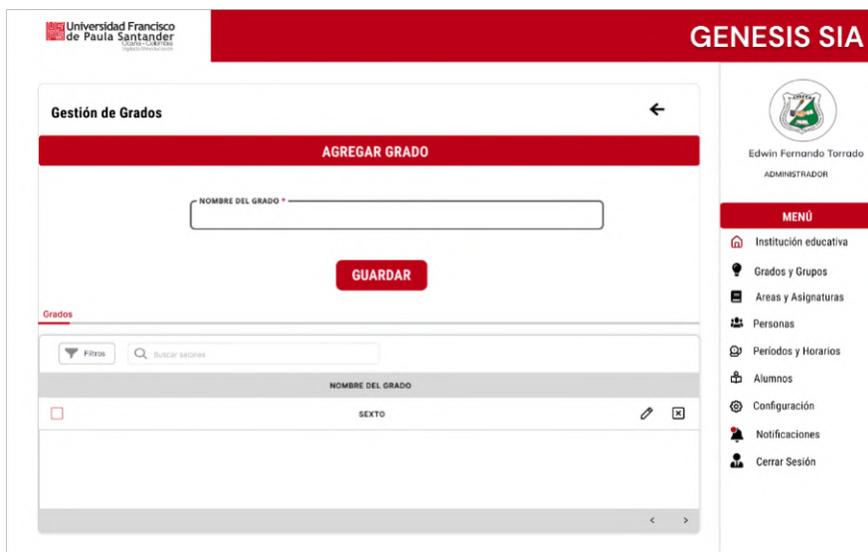
Interface de modulo grados y grupos



Fuente: Interface donde se encuentran las funcionalidades de módulo grados y grupos

Figura 30.

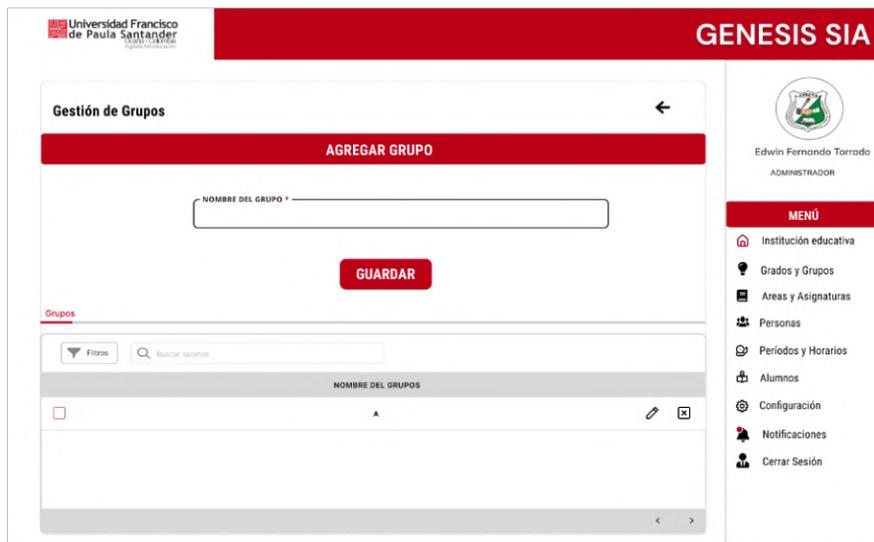
Interface gestión de grados



Fuente: Interface donde se administran los grados de la institución

Figura 31.

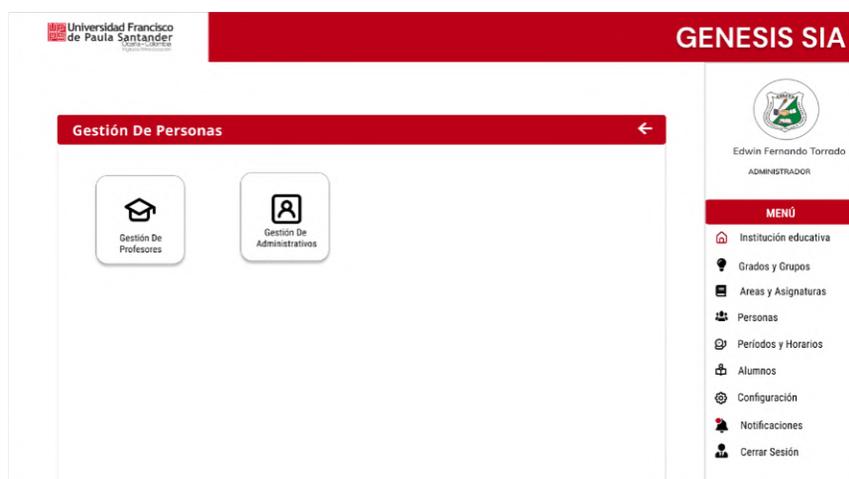
Interface gestión de grupos



Fuente: Interface donde se administran los grupos de la institución

Figura 32.

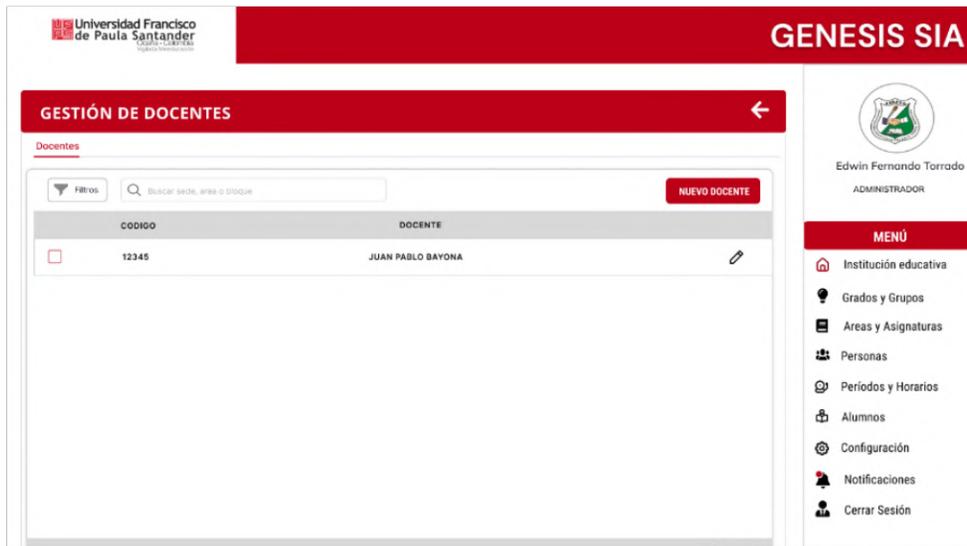
Interface gestión de personas



Fuente: Interface donde se encuentran las funcionalidades de módulo gestión de persona

Figura 33.

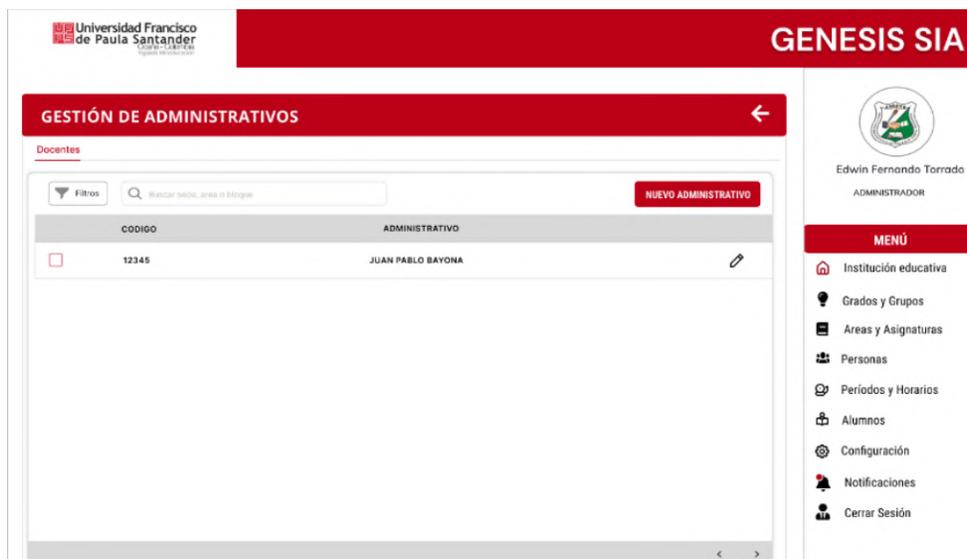
Interface de gestión de docente



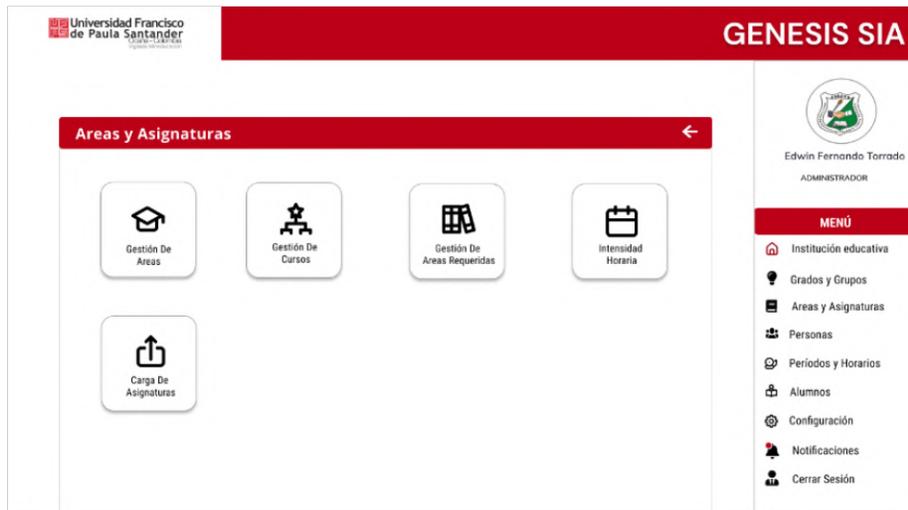
Fuente: Interface donde se encuentran las funcionalidades de módulo gestión de docentes

Figura 34.

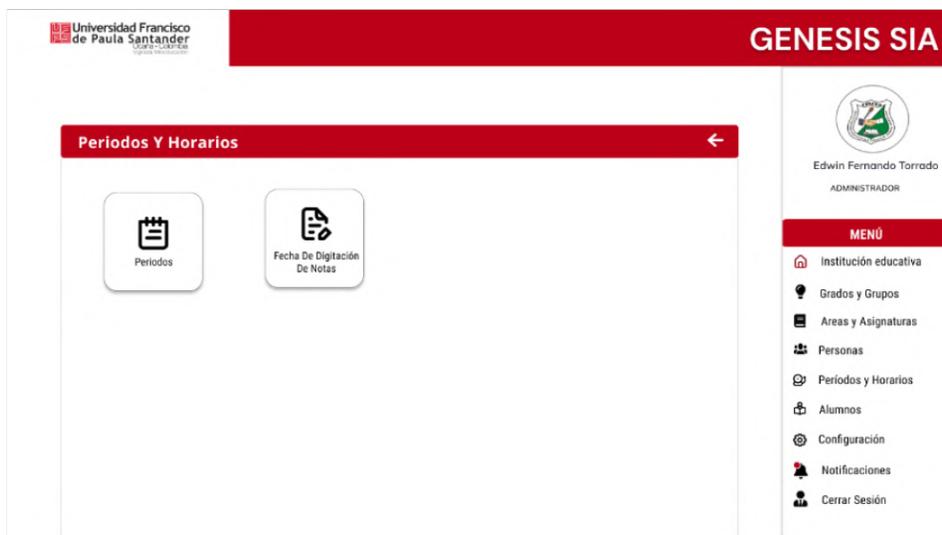
Interface gestión de administrativos



Fuente: Interface donde se encuentran las funcionalidades de módulo gestión de administrativos

Figura 35.*Interface de área y asignaturas*

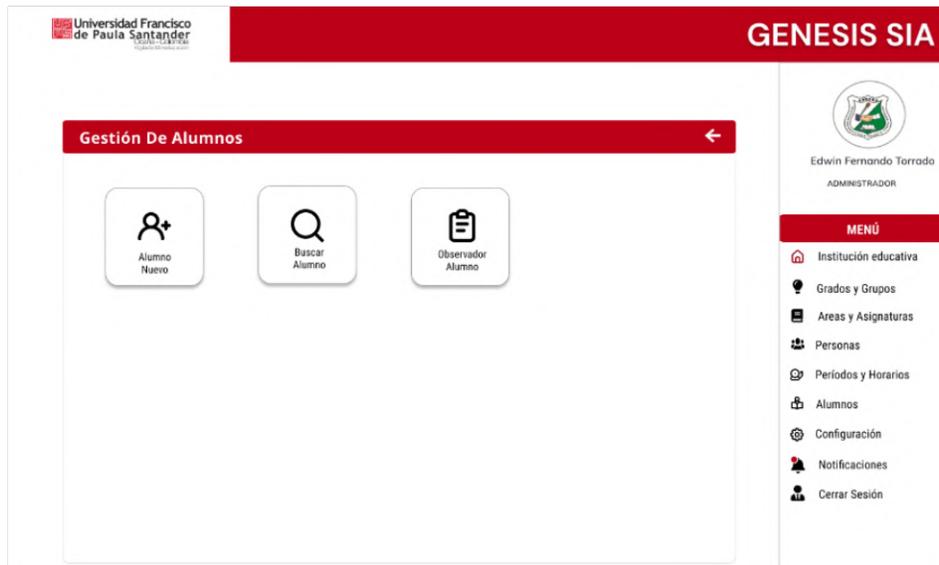
Fuente: Interface donde se encuentran las funcionalidades de módulo gestión de áreas y asignaturas

Figura 36.*Interface de periodos y horarios*

Fuente: Interface de las funcionalidades de módulo gestión de periodos y horarios

Figura 37.

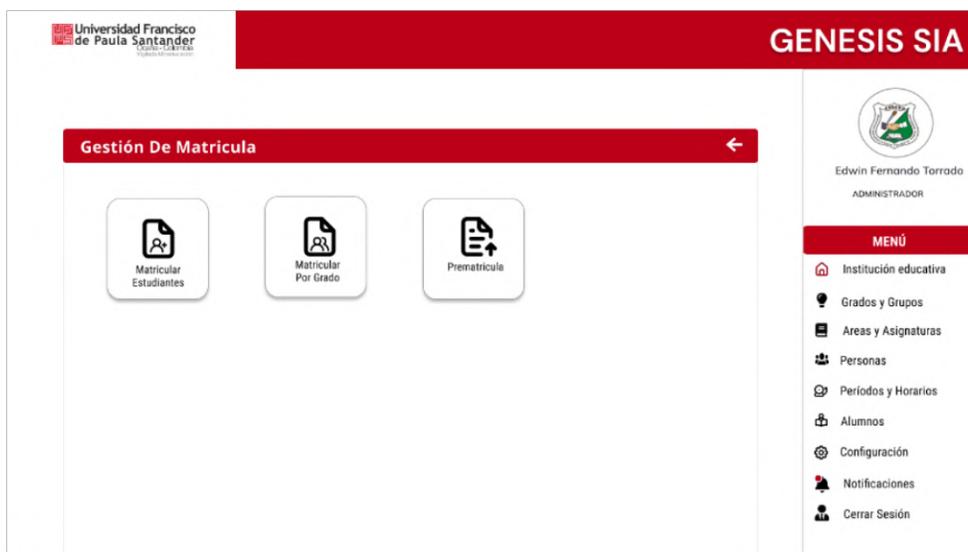
Interface de gestión de alumnos



Fuente: Interface de las funcionalidades de módulo gestión de alumnos

Figura 38.

Interface de gestión de matricula



Fuente: Interface de las funcionalidades de módulo gestión de matricula

Figura 39.

Interface de módulo concepto evaluativo

Universidad Francisco de Paula Santander
GENESIS SIA

Edwin Fernando Torrado
ADMINISTRADOR

Concepto Evaluativo

Gestión De Conceptos Evaluativos

SEDE GRADO ASIGNATURA

SEDE

GRADO ASIGNATURA

DESCRIPCIÓN DEL LOGRO

GUARDAR

MENÚ

- Institución educativa
- Grados y Grupos
- Áreas y Asignaturas
- Personas
- Periodos y Horarios
- Alumnos
- Configuración
- Notificaciones
- Cerrar Sesión

Fuente: Interface de las funcionalidades de módulo gestión de los conceptos evaluativos

Figura 40.

Interface de módulo de notas

Universidad Francisco de Paula Santander
GENESIS SIA

Edwin Fernando Torrado
ADMINISTRADOR

Notas

Asignaturas Observaciones Recuperación De Período Notas Por Estudiante

Generar Promedios Nivelación Final

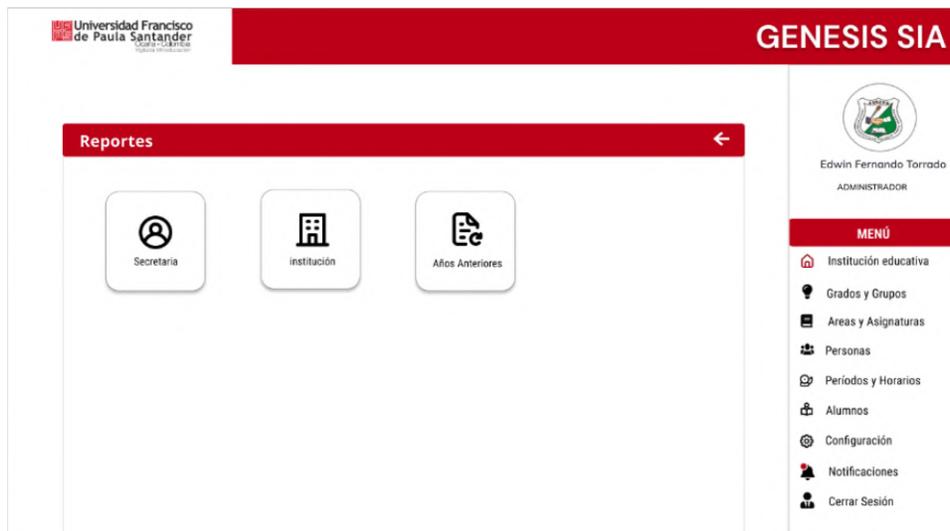
MENÚ

- Institución educativa
- Grados y Grupos
- Áreas y Asignaturas
- Personas
- Periodos y Horarios
- Alumnos
- Configuración
- Notificaciones
- Cerrar Sesión

Fuente: Interface de las funcionalidades de módulo gestión de notas de la institución

Figura 41.

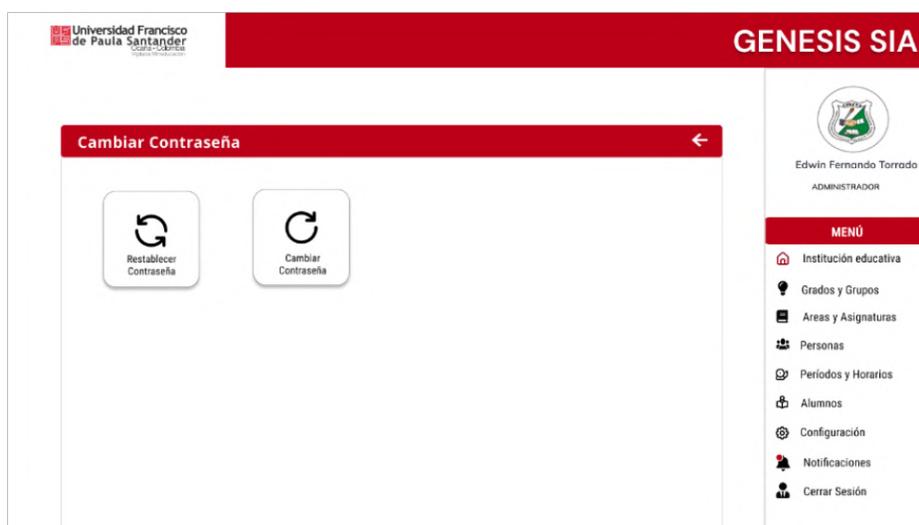
Interface de módulo de reportes



Fuente: Interface de las funcionalidades de módulo gestión de reportes de la institución

Figura 42.

Interface de cambiar contraseña



Fuente: Interface de las funcionalidades de módulo de cambio de contraseña

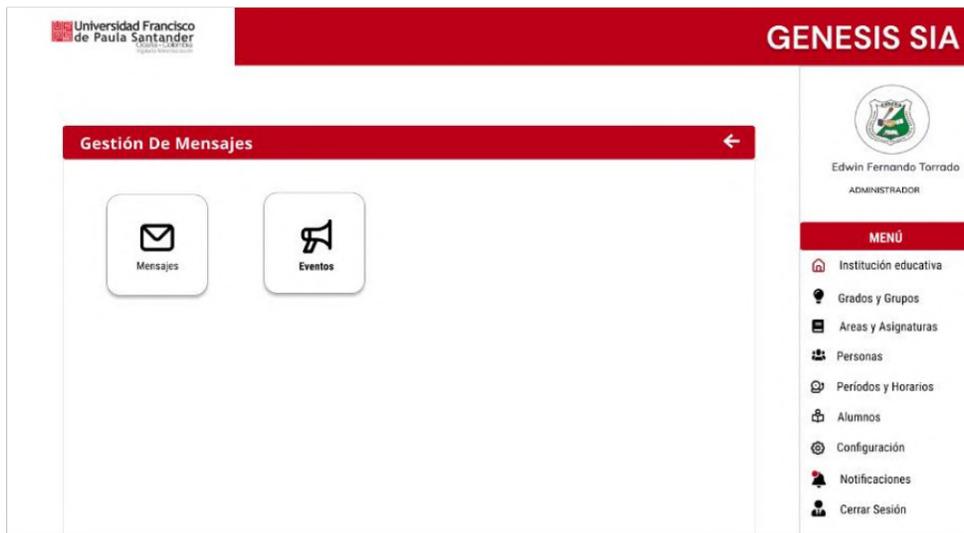
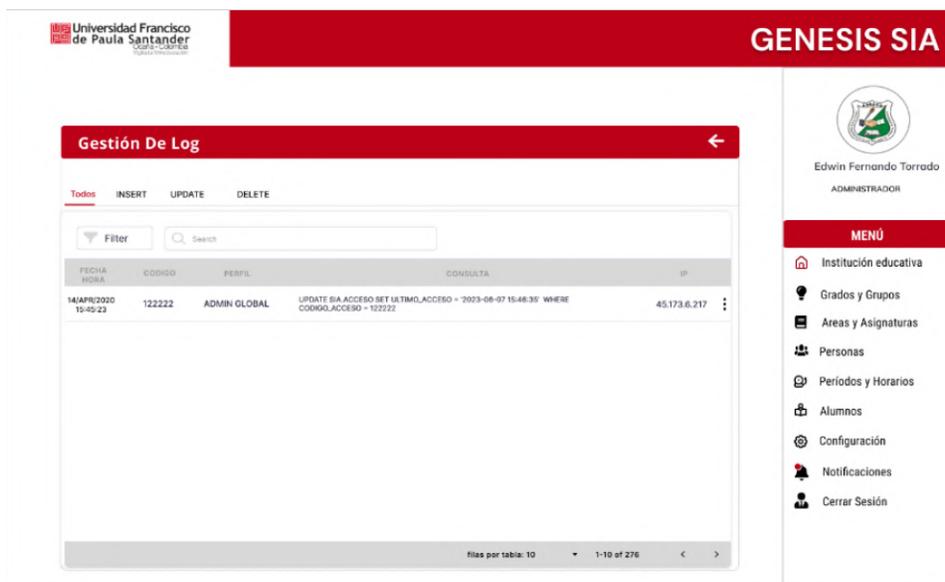
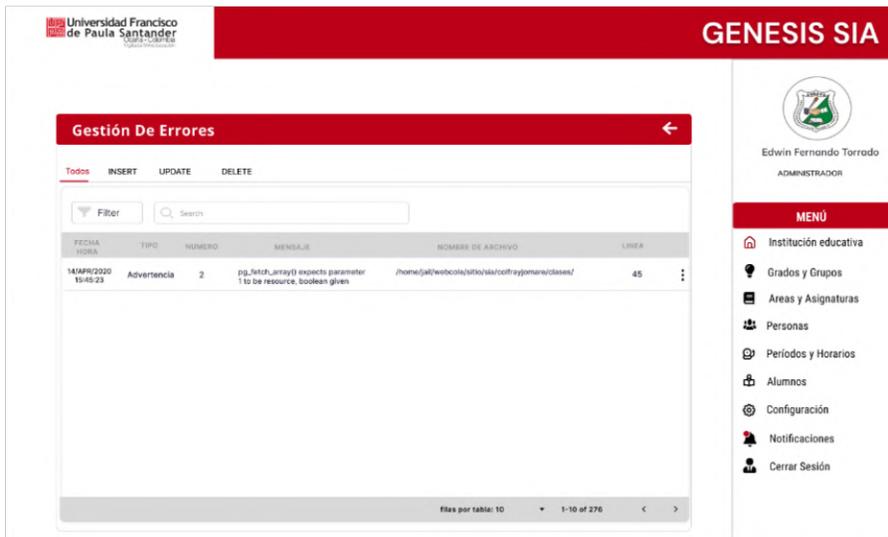
Figura 43.*Interface de gestión de mensajes***Fuente:** Interface de las funcionalidades de módulo de gestión de mensajes**Figura 44.***Interface de log***Fuente:** Interface de las funcionalidades de módulo de gestión de log

Figura 45.

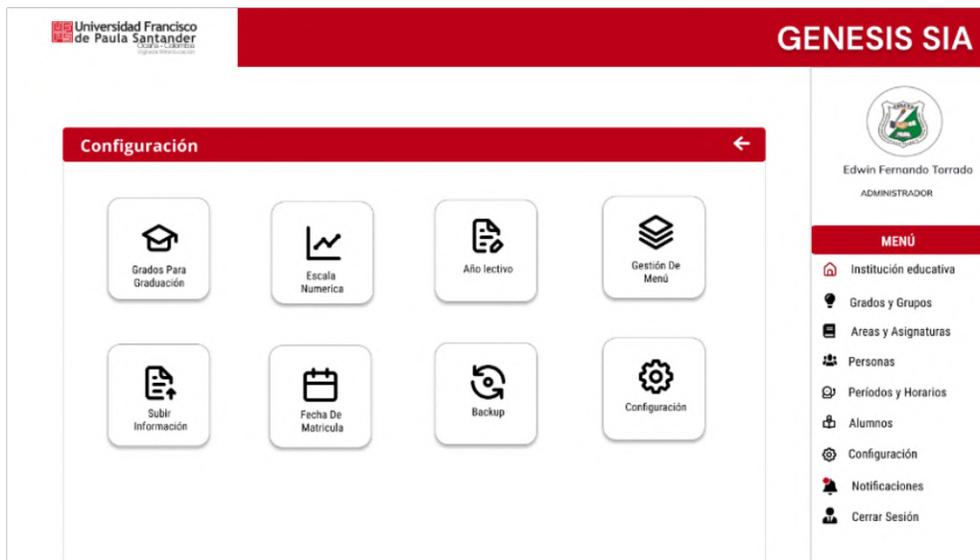
Interface de gestión errores



Fuente: Interface de las funcionalidades de módulo de gestión de errores

Figura 46.

Interface de módulo de configuración



Fuente: Interface de las funcionalidades de módulo de configuración de sistema

Figura 47.

Modulo cambiar contraseña

The screenshot shows the 'Gestión de contraseña' (Password Management) interface. At the top left is the logo of Universidad Francisco de Paula Santander. The main header is 'GENESIS SIA'. Below the header, the page title is 'Gestión de contraseña' with a back arrow. A red bar contains the text 'CAMBIAR CONTRASEÑA'. The form contains three input fields: 'CONTRASEÑA ACTUAL *', 'CONTRASEÑA NUEVA *', and 'REESCRIBA CONTRASEÑA NUEVA *'. A red 'GUARDAR' button is centered below the fields. On the right side, there is a user profile for Edwin Fernando Torrado, ADMINISTRADOR, and a 'MENÚ' section with the following items: Institución educativa, Grados y Grupos, Areas y Asignaturas, Personas, Períodos y Horarios, Alumnos, Configuración, Notificaciones, and Cerrar Sesión.

Fuente: Interface de las funcionalidades de módulo de cambio de contraseña

Figura 48.

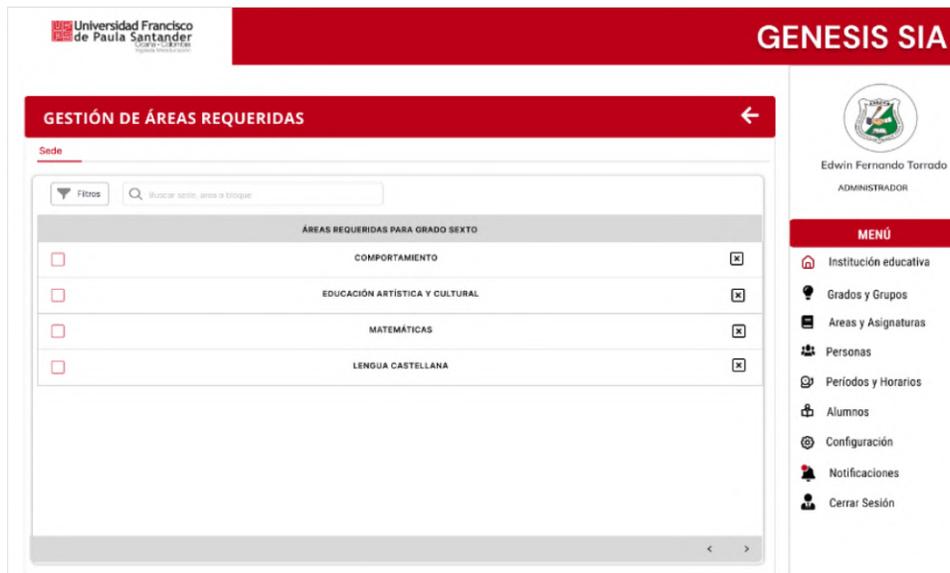
Carga académica por curso

The screenshot shows the 'Carga Académica Por Curso' (Academic Load by Course) interface. At the top left is the logo of Universidad Francisco de Paula Santander. The main header is 'GENESIS SIA'. Below the header, the page title is 'Carga Académica Por Curso' with a back arrow. A red bar contains the text 'SELECCIONE LA SEDE'. The form contains three dropdown menus: 'SEDE *', 'GRADO *', and 'GRUPOS *'. A red 'BUSCAR' button is centered below the fields. On the right side, there is a user profile for Edwin Fernando Torrado, ADMINISTRADOR, and a 'MENÚ' section with the following items: Institución educativa, Grados y Grupos, Areas y Asignaturas, Personas, Períodos y Horarios, Alumnos, Configuración, Notificaciones, and Cerrar Sesión.

Fuente: Interface de asignación de carga académica por curso

Figura 49.

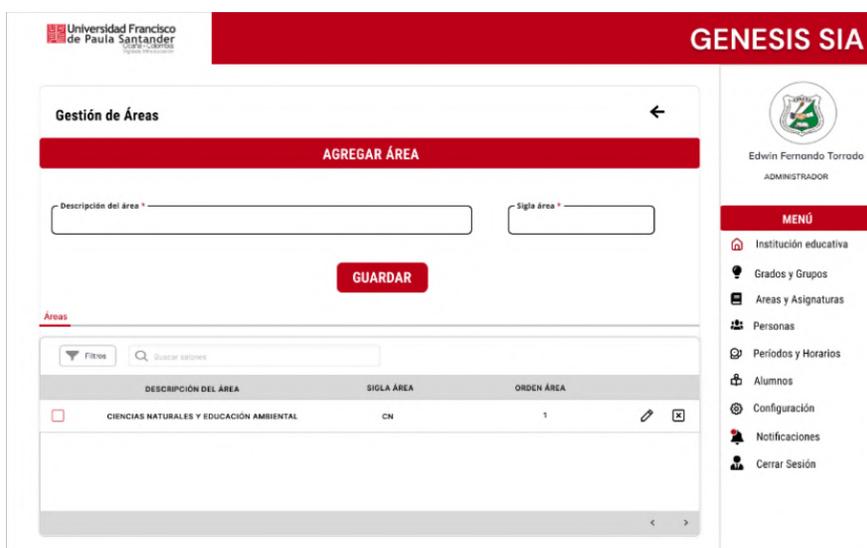
Módulo de búsqueda áreas requeridas



Fuente: Interface de búsqueda de las áreas requeridas

Figura 50.

Módulo de gestión de área



Fuente: Interface de gestión de áreas dictadas por la institución

Figura 51.

Módulo gestión de áreas requeridas

Fuente: Interface de áreas requeridas para la selección de las áreas para los cursos de la institución

Figura 52.

Módulo gestión de carga académica por curso

GRADOS	GRUPOS	AREA	ASIGNATURA	DOCENTE
SEXTO	1	CIENCIAS NATURALES Y EDUCACIÓN AMBIENTAL	NATURALES	DOCENTE DEMO

Fuente: Interface para la gestión de la carga académica de la institución educativa

Figura 53.

Módulo gestión de curso

Universidad Francisco de Paula Santander

GENESIS SIA

GESTIÓN DE CURSOS

Sede

Filtros NUEVO CURSO

GRADO	GRUPO	SALÓN	JORNADAS	TITULAR	ESTADO
SEXTO	1	SALÓN 1	MAÑANA	JUAN PABLO BAYONA	1

Edwin Fernando Torrado
ADMINISTRADOR

MENÚ

- Institución educativa
- Grados y Grupos
- Áreas y Asignaturas
- Personas
- Períodos y Horarios
- Alumnos
- Configuración
- Notificaciones
- Cerrar Sesión

Fuente: Interface para la gestión de los cursos de cada sede

Figura 54.

Módulo agregar curso por sede

Universidad Francisco de Paula Santander

GENESIS SIA

Gestión de Curso

AGREGAR CURSO

GRADO * GRUPOS * SALÓN *

JORNADA * TITULAR *

GUARDAR

Edwin Fernando Torrado
ADMINISTRADOR

MENÚ

- Institución educativa
- Grados y Grupos
- Áreas y Asignaturas
- Personas
- Períodos y Horarios
- Alumnos
- Configuración
- Notificaciones
- Cerrar Sesión

Fuente: Interface para agregar cursos a la institución

Figura 55.

Módulo para agregar intensidad horaria

Universidad Francisco de Paula Santander

GENESIS SIA

Edwin Fernando Torrado
ADMINISTRADOR

Intensidad Horaria Por Asignatura

AGREGAR ASIGNATURA

Asignaturas * Números De Horas * Peso % *

AGREGAR

Áreas

ÁREA	ASIGNATURA	NUMERO DE HORAS	PESO %
<input type="checkbox"/>	CIENCIAS NATURALES Y EDUCACIÓN AMBIENTAL	NATURALES	5 100

MENÚ

- Institución educativa
- Grados y Grupos
- Áreas y Asignaturas
- Personas
- Períodos y Horarios
- Alumnos
- Configuración
- Notificaciones
- Cerrar Sesión

Fuente: Interface para agregar la intensidad horaria de cada asignatura

Figura 56.

Módulo buscar intensidad horaria

Universidad Francisco de Paula Santander

GENESIS SIA

Edwin Fernando Torrado
ADMINISTRADOR

Intensidad horaria

SELECCIONE LA SEDE

SEDE * GRADO *

BUSCAR

MENÚ

- Institución educativa
- Grados y Grupos
- Áreas y Asignaturas
- Personas
- Períodos y Horarios
- Alumnos
- Configuración
- Notificaciones
- Cerrar Sesión

Fuente: Interface para búsqueda de intensidad horaria por sede y curso

Figura 57.

Módulo de gestión de periodos

Universidad Francisco de Paula Santander

GENESIS SIA

Gestión De Periodos

PERIODOS

Descripción Período *

Peso *

Fecha Inicia *

Fecha Final *

ULTIMO PERIODO

AGREGAR

Asignaturas

Filtros

PERIODOS	FECHA INICIO	FECHA FINAL	PESO	ULTIMO
PRIMERO	2023-01-22	2023-04-06	20	NO

Edwin Fernando Torrado
ADMINISTRADOR

MENÚ

- Institución educativa
- Grados y Grupos
- Áreas y Asignaturas
- Personas
- Periodos y Horarios
- Alumnos
- Configuración
- Notificaciones
- Cerrar Sesión

Fuente: Interface para agregar, editar o eliminar periodos en las sedes de la institución

Figura 58.

Módulo para agregar nuevo alumno

Universidad Francisco de Paula Santander

GENESIS SIA

Gestión de alumnos

AGREGAR ESTUDIANTE

TIPO DE DOCUMENTO *

DOCUMENTO *

LUGAR DE EXPEDICIÓN DEPARTAMENTO *

LUGAR DE EXPEDICIÓN MUNICIPIO *

LUGAR DE NACIMIENTO *

FECHA DE NACIMIENTO *

NOMBRES *

APellidos *

NACIONALIDAD *

RESIDENCIA *

ESTADO CIVIL *

SEXO *

DIRECCION *

TELEFONO *

EMAIL *

ESTRATO *

SISEN *

REGIMEN *

EPS / AFS *

BARRIO DE RESIDENCIA *

GUARDAR

Edwin Fernando Torrado
ADMINISTRADOR

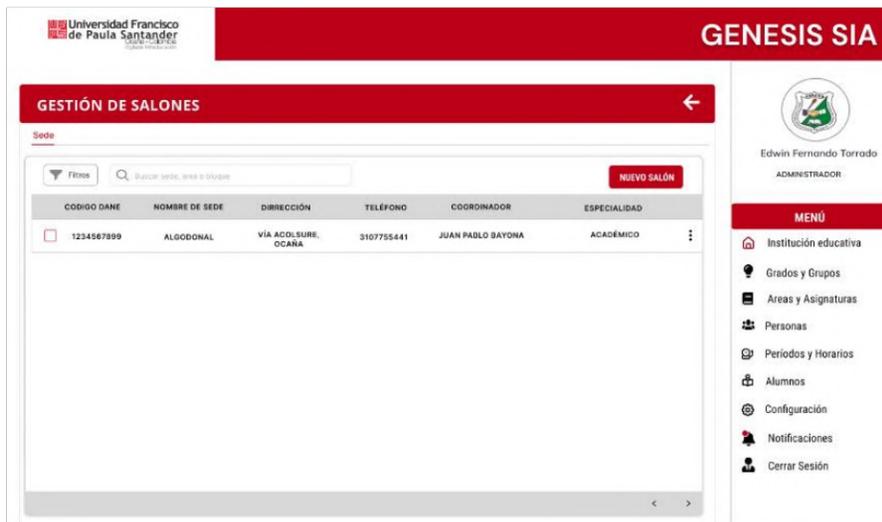
MENÚ

- Institución educativa
- Grados y Grupos
- Áreas y Asignaturas
- Personas
- Periodos y Horarios
- Alumnos
- Configuración
- Notificaciones
- Cerrar Sesión

Fuente: Interface para agregar un nuevo alumno a la institución

Figura 59.

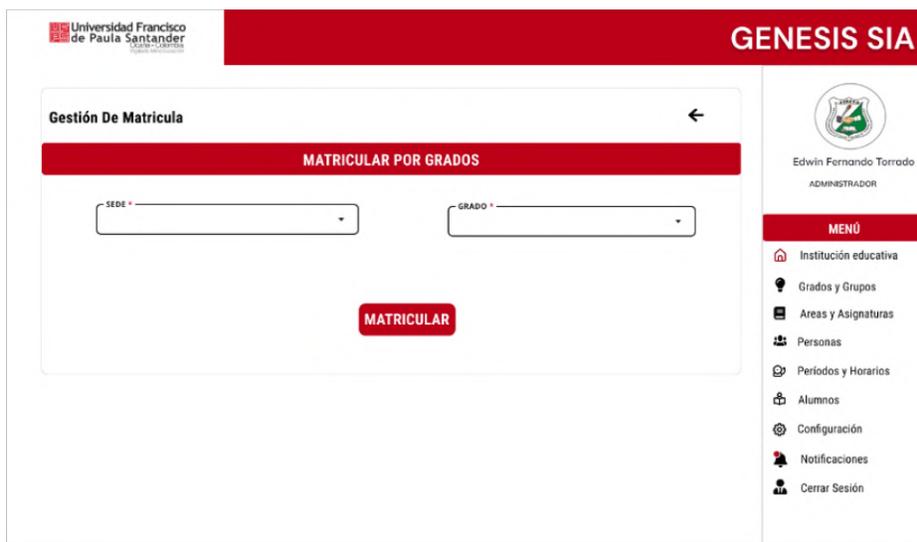
Módulo de gestión de salones



Fuente: Interface para agregar, editar o eliminar salones de cada sede

Figura 60.

Módulo de matrícula por grados



Fuente: Interface para matricular alumnos por grados

Figura 61.

Módulo de gestión de eventos

The screenshot shows the 'Gestión de mensajes' interface for 'NUEVO EVENTO' in the GENESIS SIA system. The interface is part of the Universidad Francisco de Paula Santander system. It features a red header with the university logo and the text 'GENESIS SIA'. Below the header, the user is identified as 'Edwin Fernando Torrado ADMINISTRADOR'. A sidebar menu on the right lists various system functions: 'Institución educativa', 'Grados y Grupos', 'Areas y Asignaturas', 'Personas', 'Periodos y Horarios', 'Alumnos', 'Configuración', 'Notificaciones', and 'Cerrar Sesión'. The main content area is titled 'Gestión de mensajes' and contains a form for creating a new event. The form includes a 'DESTINATARIO*' dropdown menu, a 'SEDE*' dropdown menu, a large text area for the 'MENSAJE', and a red 'ENVIAR' button at the bottom.

Fuente: Interface para la gestión de eventos de la institución educativa

Figura 62.

Módulo de gestión de mensajes

The screenshot shows the 'Gestión de mensajes' interface for 'NUEVO MENSAJE' in the GENESIS SIA system. The interface is part of the Universidad Francisco de Paula Santander system. It features a red header with the university logo and the text 'GENESIS SIA'. Below the header, the user is identified as 'Edwin Fernando Torrado ADMINISTRADOR'. A sidebar menu on the right lists various system functions: 'Institución educativa', 'Grados y Grupos', 'Areas y Asignaturas', 'Personas', 'Periodos y Horarios', 'Alumnos', 'Configuración', 'Notificaciones', and 'Cerrar Sesión'. The main content area is titled 'Gestión de mensajes' and contains a form for creating a new message. The form includes a 'DESTINATARIO*' dropdown menu, a large text area for the 'MENSAJE', and a red 'ENVIAR' button at the bottom.

Fuente: Interface para la gestión de los mensajes de la institución educativa

Figura 63.*Módulo de prematricula*

The screenshot displays a web interface for pre-registration. At the top left is the logo for Universidad Francisco de Paula Santander. A red header bar on the right contains the text 'GENESIS SIA'. The main content area is titled 'Gestión De Matricula' and features a red sub-header 'PREMATRICULAR POR GRADOS'. Below this, there are two dropdown menus labeled 'SEDE' and 'GRADO'. A red button labeled 'MATRICULAR' is positioned below the dropdowns. On the right side, a sidebar shows the user's name 'Edwin Fernando Torrado' and role 'ADMINISTRADOR', followed by a 'MENÚ' section with a list of navigation options: 'Institución educativa', 'Grados y Grupos', 'Areas y Asignaturas', 'Personas', 'Períodos y Horarios', 'Alumnos', 'Configuración', 'Notificaciones', and 'Cerrar Sesión'.

Fuente: Interface para la prematricula de los alumnos por grados

Conclusiones

A través de reuniones con el cliente y encuestas realizadas a los usuarios del sistema de información académico Genesis SIA v3, se identificaron y comprendieron las necesidades y requisitos específicos de los usuarios. Mediante un análisis exhaustivo de la información recolectada, se definió la arquitectura y los componentes necesarios para desarrollar el nuevo sistema de información académico Genesis SIA v4.0.

En base a estas definiciones, se determinaron las tecnologías que mejor se ajustaban a los requerimientos del sistema. Se decidió utilizar el framework Spring Boot para el desarrollo del backend, debido a su capacidad para desarrollar aplicaciones Java de forma eficiente y escalable. Para el frontend, se optó por el uso de Angular, una tecnología de desarrollo web ampliamente reconocida por su capacidad de crear interfaces de usuario interactivas y modernas. En cuanto a la base de datos, se seleccionó PostgreSQL, una opción confiable y robusta que proporciona un alto rendimiento y seguridad en el almacenamiento de los datos académicos.

En el nuevo sistema de información académico Genesis SIA v4, se tomó la decisión de utilizar la arquitectura hexagonal para su desarrollo. Esta arquitectura permite separar claramente la lógica de negocio del sistema de sus interfaces externas, lo que facilita su mantenimiento, evolución y adaptación a futuros cambios.

Además, se llevó a cabo un rediseño completo de las interfaces de usuario con el objetivo de lograr un sistema más moderno y atractivo visualmente. Se trabajó en mejorar la experiencia del usuario, optimizando la usabilidad, la accesibilidad y la interacción con el sistema.

Recomendaciones

Para la etapa de desarrollo del Genesis SIA v4 se recomienda tener en cuenta la adopción de las prácticas de desarrollo de software planteado en esta investigación que permitan crear un software robusto y escalable.

Muy necesario al momento de desarrollar el software se adopten las tecnologías planteadas como Spring boot que son estándares en el desarrollo de software empresarial y están preparadas para crear aplicaciones de alta complejidad con énfasis en la seguridad de las aplicaciones lo que permitirá tener desarrollos dentro de la oficina de interconectividad de mayor confiabilidad y mantenibilidad.

Tener presente en el desarrollo de nueva versión del Genesis SIA investigar e implementar versionado del software, integración continua y despliegue continuo, entornos de testeo y aplicación de metodologías ágiles en equipo de desarrollo que permitan crear software resistente a fallos.

Referencias

- Ajzen, I. (2019). *The Theory of Planned Behavior The Theory of Planned Behavior*. 5978(August). [https://doi.org/10.1016/0749-5978\(91\)90020-T](https://doi.org/10.1016/0749-5978(91)90020-T)
- Alcaldía de Ocaña, N. (2023). *Municipio de Ocaña - Ocaña Turística*. 11–14.
- Ahmed, W. M., Hossain, M. A., & Alam, M. S. (2020). Design and Implementation of Microservices Architecture in Educational Information System. 2020 7th International Conference on Informatics, Electronics & Vision (ICIEV), pp. 581-585.
- Aliaga Meléndez, C., & Dávila Rojas, O. (2021). Blackboard Platform: A tool for the Teaching Learning process. HAMUT, Revista Cuatrimestral de divulgación científica Universidad a las Peruanas, 42-58.
- Apache Software Foundation, T. (2021). Apache Tomcat. Obtenido de <https://tomcat.apache.org/>
- Arias, J., & Murillo, D. (2016). Diseño e implementación de un sistema de información académico basado en la plataforma Microsoft. . Revista Tecnología y Sociedad, 12(26), 54-66.
- Arroyave, M., & López, O. (2013). Desarrollo de un sistema de información académica para la Universidad Tecnológica de Pereira.
- Balalaie, A., Heydarnoori, A., & Jamshidi, P. (2016). Microservices architecture enables devOps: Migration to a cloud-native architecture. IEEE Software,, 33(3), 42-52.
- Bakken, B. (2015). Object-Oriented Programming in PHP. . *In Pro PHP Refactoring Apress.*, (pp. 35-59).
- Burch, J., & Grudnitski, G. (1989). *Information systems: Theory and practice*. John Wiley & Sons.

- Congreso de Chile, L. d. (2002). *Ley 19.799, de 19 de octubre de 2002, de Fomento del Software Libre (Chile)*. Obtenido de, de Fomento del Software Libre (Chile)
- Congreso de la República de Colombia, C. (2023). *Ley Estatutaria 1581 de 2012. 52306*.
- Congreso de la República del Perú, P. (2008). *Ley de Firmas y Certificados Digitales 27269*.
- Consejo Superior Universitario UFPSO, C. (2019). *Acuerdo 51 de 2019. 51*, 1–55.
- Consejo Superior Universitario UFPSO, N. (2019). *Identidad Institucional UFPSO*. 12–26.
[https://ufpso.edu.co/ftp/pdf/documentos/Identidad Institucional.pdf](https://ufpso.edu.co/ftp/pdf/documentos/Identidad%20Institucional.pdf)
- Cornejo, A. (2015). *Normas ISO 12207*. 11–14.
- Chappell, D. A. (2004). *Introduction to service-oriented architecture*. O'Reilly Media, Inc.
- Chavarria-Neira, B., & Gudiño, E. (2017). *Ingeniería de Sistemas. Implementación de un Servidor Web y un Diseño de una Página utilizando Herramientas de Software Libre para el Dispensario "Sagrada Familia" de la Ciudad de Guayaquil*. Universidad Politécnica Salesiana Sede Guayaquil, Guayaquil, Ecuador.
- Cheswick, W. R., & Bellovin, S. M. (2003). *Firewalls and internet security: Repelling the wily hacker*. Addison-Wesley Professiona.
- CISCO, S. (2021). *Encriptación de datos*. Obtenido de
<https://www.cisco.com/c/en/us/products/security/what-is-encryption.html>
- Conolly, T., & Carolyn, B. (2015). *Database Systems: A Practical Approach to Design, Implementation and Management*. Londres: Global Edition.
- Coordinación General de Planeación y Simplificación de la Gestión Institucional, U. (2023). *Sistema Integral de Información Académica*. Obtenido de
<https://www.planeacion.unam.mx/subdireccion-de-sistemas-de-informacion-estadistica/sistema-integral-de-informacion-academica/>

Coronel, C., & Morris, S. (2017). Database Systems: Design, Implementation and Management.

Boston: Cengage Learning.

Cruz, A., Sanchez, A., & Guerrero, J. (2013). SOA-based academic management system. IEEE -

In 2013 8th Iberian Conference on Information Systems and Technologies (CISTI) ., (pp. 1-6).

Departamento Administrativo de la Función Pública, pp. (2000). *Ley 603 de 2000*. 1–2.

División-de-sistemas, U. (2023). *Sistema de Información Institucional*. 1–37.

<https://divisis.ufpso.edu.co/contenido/15/sistemas-de-informacion.html>

De la Cruz Barboza, L. (2018). e Licenciado en Educación Especialidad: Matemática e

Informática. Concepto y definición de Sistemas Operativos, historia de los Sistemas

Operativos, características de los Sistemas Operativos, clasificación de los sistemas

operativos, Aplicaciones. Universidad Nacional de Educación Enrique Guzmán y Valle,

Lima, Perú.

Deitel, P., & Harvey, D. (2018). Java: How To Program, Early Objects. Pearson.

Downey, A. (2012). Think Python: How to Think Like a Computer Scientist. Green Tea Press.

O'REILLY.

Elizondo Rodríguez, A. (2013). *Sistema de Información Académica*.

European Commission, E. (2021). *Erasmus*. 1–3.

Eastern Daylight Time, B. (19 de 05 de 2020). Businesswire. Obtenido de La Universidad de

California, Berkeley se asocia con NTT para acelerar la aplicación de tecnologías

inteligentes: <https://www.businesswire.com/news/home/20200519005955/es/>

Enriquez, J. G. (2016). USABILIDAD EN APLICACIONES MÓVILES. Argentina: UNPA,

Universidad Nacional de la Patagonia Austral.

Erl, T. (2005). *Service-oriented architecture: concepts, technology, and design*. . Prentice Hall.

Fernández, R., & García, A. (2012). Diseño e implementación de un sistema de gestión académica basado en servicios web y Oracle. En *Memorias del 9° Congreso Internacional de Informática en Salud*.

Flores Huacho, N. J. (2021). *Licenciado en Educación: Especialidad: Informática. HTML5*.

Universidad Nacional de Educación Enrique Guzmán y Valle, Lima, Perú.

Fowler, M. (2002). *Patterns of Enterprise Application Architecture*. . Addison-Wesley Professional.

Fowler, M. (2002). *Patterns of Enterprise Application Architecture*. Addison-Wesley. Obtenido de <https://www.php-fig.org/psr/psr-12/>

Giménez, J. A. (2019). *Licenciatura en Análisis de Sistemas Informáticos. Buenas prácticas en el diseño de bases de datos*. Universidad Tecnológica Intercontinental, Asunción, Paraguay.

Obtenido de file:///C:/Users/WPOSS/Downloads/9.pdf

Gómez, C. F. (2019). *Ingeniero de Sistemas - Magister: Diseño y Gestión de Proyectos*

Producción de los Recursos Educativos

Tecnológicos. La importancia de la Usabilidad en la pro

ad UNIMINUTO, Bogotá.

Digitales Abiertos. Secretaría de Educación y Universida

González, J. M., & Giraldo, M. A. (2016). *ACAD-Web: Sistema de gestión académica en línea*.

Revista de Investigación Académica, 1(1), 23-28.

Guíñez, J. R., & Pérez, J. G. (2014). *SOA-based academic management system in the*

University of Guadalajara. IBBB in 2014 IBBB Global Engineering Education Conference

(EDUCON), (pp. 774-778).

García, J. (2018). *Génesis SHK 3.0*.

- Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). Design patterns: Elements of reusable
 Design patterns: Elements of reusable object-oriented software. . *Addison-Wesley
 Professional*.
- Hernandez Sampiere; Pilar Baptista; Fernández Collado. (2014). *Metodología de la Investigación*
 (Vol. 6).
- Han, J., Lee, Y., & Yoo, S. (2017). An Event-Driven Approach for Academic Information
 Management System. . Proceedings of the 2017 International Conference on Information
 and Communication Technology Convergence, 123-124. doi:10.1109/ICTC.2017.8190795.
- Hernández, U. (11 de 03 de 2019). CTO de Código Facilito. Obtenido de Angular, React o Vue
 ¿cuál elegir?: <https://codigofacilito.com/articulos/angular-react-vue>
- Ho, T. (2017). Security Best Practices for Banner Student Information System. . Educause
 Security Professionals Conference.
- Holovaty, A., & Kaplan-Moss, J. (2009). The Definitive Guide to Django. Spring Link.
- Humala Tasso, Ollanta; Jiménez Mayor, Juan; Pedraza Sierra, W. (2013). *Ley 29733 de
 Protección de Datos Personales*. 491320–491334.
- IEEE Standard, I. (2009). *IEEE Standard for Software Design Descriptions (IEEE Std 1016-
 2009)*. Obtenido de <https://ieeexplore.ieee.org/document/5166874>
- Israel Núñez, P. (2018). *La gestión de la Información, el conocimiento, la inteligencia y el
 aprendizaje organizacional desde una perspectiva socio-psicológica*. 1–48.
- IBM. (2004). Introduction to service-oriented architecture. . Obtenido de
<https://www.ibm.com/developerworks/library/ws-soa-design1/>
- IBM. (2021). Event-driven architecture (EDA). Obtenido de
<https://www.ibm.com/cloud/learn/event-driven-architecture>

- ISO/IEC. (2013). ISO/IEC 27001:2013 Information technology – Security techniques – Information security management systems – Requirements. ISO.
- Jaimes, Juan; García, J. (2016). *Sistemas de Información académica : herramienta estratégica para ser competitivos*. 10–13.
- Jisc. (2017). *Student Records Systems: From Past to present*. Obtenido de <https://www.jisc.ac.uk/blog/student-records-systems-from-past-to-present-21>
- Khalib, R., Ghayyur, S., & Igbal, M. (2019). A Systematic Literature Review of Design Pattern Practices. . *Journal of Software: Evolution and Process*, 31(10).
- Kamal, N., Shah, M. A., & Hussain, M. (2019). Microservices Architecture for Higher Education Information Systems: A Case Study. 2019 International Conference on Frontiers of Information Technology (FIT), 107-112.
- Kizza, J. M. (2016). *Computer network security*. . Springer.
- Larco Guzmán, Geovanny; Navarrete Zurita, D. (2020). *Sistema de gestión seguimiento académico y sílabo*.
- Lebensold, J. (2018). *React Native Cookbook: Bindings the Web to Native Plataforms*. Estados Unidos: O'Reilly.
- Li, Y., Chen, J., Zhang, Y., & Liu, Y. (2016). Design and Implementation of an Academic Information Management System Based on the B/S Architecture. . *Journal of Educational Information and Technology*, 1, 29-34.
- Liao, C. (2008). Development and use of management information systems in higher education. *Journal of Higher Education Policy and Management*, 11-21.
- Ministerio de Educación Nacional, C. (2019). *SISTEMA NACIONAL DE INFORMACIÓN DE LA EDUCACIÓN SUPERIOR - SNIES*. 57(57).

- Mabrouk, M. A. (2019). Design and Development of an Academic Information System (AIS) for Secondary Schools in Saudi Arabia. *Journal of Educational Technology Development and Exchange*, 12(1), 1-14.
- Ministerio de trabajo, C. (2012). *Resolución 1409 de 2012.pdf*.
- Martin, R. (2018). Clean Architecture: A CRAFTSMAN'S GUIDE TO SOFTWARE STRUCTURE AND DESIGN. Estados Unidos: Pearson Education.
- Martínez, M. M., & Mendoza, H. M. (2016). Sistema de información académica y financiera (SIAF): una propuesta de integración en universidades mexicanas. . *Revista Electrónica de Investigación Educativa*, 18(1), 1-16.
- Maturana, L., & Méndez, E. (2018). Desarrollo de un sistema de información académica. *Revista Científica de Tecnología Educativa*,. SISAC, 4(1), 17-22.
- Mendoza, M. (16 de 07 de 2020). OpenWebinars. Obtenido de Qué es un lenguaje de programación: <https://openwebinars.net/blog/que-es-un-lenguaje-de-programacion/>
- Mendoza, M. J. (2020). Desarrollo de un sistema de información para la gestión académica en la Universidad de los Andes. *Revista de Investigación Académica*, 5(1), 34-42.
- Microsoft Corporation, I. (2021). Internet Information Services (IIS). . Obtenido de <https://www.iis.net/>
- Moodle, D. (26 de 12 de 2022). Acerca de Moodle. Obtenido de https://docs.moodle.org/all/es/Acerca_de_Moodle
- Mozilla Developer Network, C. (2021). CSS: Cascading Style Sheets. Obtenido de <https://developer.mozilla.org/en-US/docs/Web/CSS>
- Mozilla Developer Network, C. (2021). HTML: Hypertext Markup Language. Obtenido de <https://developer.mozilla.org/en-US/docs/Web/HTML>

Mozilla Developer Network, C. (2021). JavaScript. Obtenido de

<https://developer.mozilla.org/en-US/docs/Web/JavaScript>

MySQL. (2023). Documentación MySQL. Obtenido de <https://dev.mysql.com/doc/>

Newman, S. (2015). *Building Microservices: Designing Fine-Grained Systems*. . O'Reilly Media, Inc.

Tolosa Martínez, C. (2011). *MANUAL INTERNO DE POLÍTICAS Y PROCEDIMIENTOS DE DATOS PERSONALES*.

O'Brien, James; Marakas, G. (2012). *Management Information Systems*.

Oracle. (2022). *Java*. Obtenido de <https://www.oracle.com/java/>

Oliveira, T., & Ferreira, R. (2016). A NoSQL Approach to Academic Information Management. *Proceedings of the 20th International Database Engineering & Applications Symposium*, (pp. 11-16). ACM. <https://doi.org/10.1145/2912251.2912255>.

Oracle. (2023). Obtenido de <https://www.oracle.com/database/>

Ortega, D., Guevara, M., & Benavides, J. (2017). Elementary: Un Framework de Programación Web. *Electrónica de Estudios Telemáticos*, 144-171.

Palma Pérez, N. (2020). Efficiency of Apache 2 and Nginx web servers: a case study . *Serie Científica de la Universidad de las Ciencias Informáticas*, 97-112.

cteristics, and challenges.

Papazoglou, M. P. (2016). Service-oriented computing: concepts, characteristics, and challenges. *Information systems*, 33(3), 283-315.

e un sitio Web para un

Pérez Pérez, V. (s.f.). Técnico en Ingeniería Informática. Desarrollo de un sitio Web para un colegio. Universidad Politécnica de Valencia, Valencia, España.

y.postgresql.org/

PostgreSQL. (2023). ¿Nuevo en PostgreSQL? Obtenido de <https://www.postgresql.org/>

- Regulation European Union, a. (2016). Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive. *Official Journal of the European Union*, 119.
- Rogers, E. M. (2010). *Diffusion of innovations*. Simon and Schuster.
- Rails Core Team. (30 de 06 de 2016). Rails 5.0: Action Cable, modo API y mucho más. Obtenido de <https://rubyonrails.org/2016/6/30/Rails-5-0-final>
- Rendón, Y. A. (28 de 05 de 2019). Academia Pragma. Obtenido de Bases de datos relacionales vs. no relacionales: <https://www.pragma.com.co/academia/lecciones/bases-de-datos-relacionales-vs.-no-relacionales>
- Richards, M., & Ford, N. (2015). *Software Architecture Patterns*. O'Reilly Media, Inc.
- Robledano, A. (24 de 09 de 2019). OpenWebinars. Obtenido de Qué es MySQL: Características y ventajas: <https://openwebinars.net/blog/que-es-mysql/>
- Rueda, L., & Orozco, J. (2014). Diseño e implementación del sistema de información y seguimiento de la responsabilidad académica (SISRA) de la Universidad Nacional de Colombia. . *Revista Tecnura*, 18(41), 103-115.
- Sánchez, R. A., Hueros, A. D., & García, A. M. (2017). A review of the acceptance of academic information systems. *Telematics and Informatics*, 34(5), 818-835.
- Sahay, A., & Patwardhan, V. (2015). Design and Development of Academic Information System. . *International Journal of Computer Science and Mobile Computing*, 4(6), 331-339.
- Salazar, J., & Linares, R. (2016). Diseño e implementación de un sistema de gestión académica para la Universidad Autónoma de Bucaramanga.
- Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). *Operating system concepts*. . Wiley.

- Stajano, F. (2011). *Security for ubiquitous computing*. . Wiley.
- Stallings, W. (2017). *Cryptography and network security: principles and practice*. Pearson Education.
- Stauffer, M. (2019). *Laravel Up & Running: A Framework for Building Modern PHP Apps*. Estado Unidos: O'Reilly.
- Spring Boot, W. (2022). *Spring Boot*. Obtenido de <https://spring.io/projects/spring-boot>
- Tanenbaum, A. S., & Woodhull, A. S. (2015). *Operating systems: Design and implementation*. . Prentice Hall Press.
- The Apache Software Foundation., I. (2023). HTTP Server Project. Obtenido de <https://httpd.apache.org/>
- Universidad de Antioquia, C. (2012). *Sistema de Información Integral*. Obtenido de <https://www.udea.edu.co/wps/portal/udea/web/inicio/udea-noticias>
- Universidad de Pamplona, C. (2004). *Sistema de Información Académica*. Obtenido de https://www.unipamplona.edu.co/unipamplona/portallIG/home_1/recursos/sia/sia.jsp
- Universidad de pamplona, C. (2016). *Implementación de la nueva versión del Sistema de Información Académica SIA2*. Obtenido de https://www.unipamplona.edu.co/unipamplona/portallIG/home_1/recursos/sia/Noticias2016/Noticia1.jsp
- Universidad del valle, c. (2017). *Sistema de Información Institucional*. Obtenido de <https://www.univalle.edu.co/registro/sistema-de-informacion-estudiantil/>
- Universidad Nacional de Colombia, c. (2017). *Sistema de Información Académica*. Obtenido de <https://www.jisc.ac.uk/blog/student-records-systems-from-past-to-present-21>
- United States Copyright Office, cop. (2023). What is copyright? *Copyright in the Music*

Industry, 15–26. <https://doi.org/10.4337/9781839101274.00014>

Universidad Francisco de Paula Santander- Mision & Visión, N. . (2023). Misión y Visión de la Universidad Francisco de Paula Santander Ocaña. *Universidad Francisco de Paula Santander*, 109. <https://ufpso.edu.co/Mision-vision>

Universidad Francisco de Paula Santander-Ocaña, L. (2008). *Resolución No. 0233. 0233(0233)*, 1–3.

Universidad Francisco de Paula Santander - Objetivos Institucionales, N. . (2023). Objetivos Institucionales. *Universidad Francisco de Paula Santander*, 109. <https://ufpso.edu.co/Objetivos>

Universidad Francisco de Paula Santander, C. (2018). *Manual de Usuario para el Sistema de Información Académica*.

Universidad Francisco de Paula Santander Ocaña - Mapa de Procesos, N. (2023). *Mapa de Universidad Francisco de Paula Santander Ocaña - Organigrama*. (2023). *Organigrama UFPSO*. <https://ufpso.edu.co/Estructura>

Universidad Francisco de Paula Santander Ocaña - Reseña Histórica, N. . (2023). Reseña Histórica Universidad. *Tesis, 024*, 1–49. <https://ufpso.edu.co/Historia>

University of Tsukuba, J. (2023). *University of Tsukuba*. 1–6.

UC Berkeley, U. o. (2023). Berkeley. Obtenido de Sistemas de Información Estudiantil: <https://sis.berkeley.edu/>

Universidad Nacional Autónoma de México, U. (2021). Un modelo de planeación y los sistemas de información para la toma de decisiones en una organización compleja. México: Cuadernos de Planeación Universitaria.

University of Illinois, U. (2023). University of Illinois Chicago. Obtenido de

<https://www.uic.edu/>

UC Berkeley, U. o. (2023). Berkeley. Obtenido de Sistemas de Información Estudiantil:

<https://sis.berkeley.edu/>

Universidad Nacional Autónoma de México, U. (2021). Un modelo de planeación y los sistemas de información para la toma de decisiones en una organización compleja. México:

Cuadernos de Planeación Universitaria.

University of Illinois, U. (2023). University of Illinois Chicago. Obtenido de

<https://www.uic.edu/>

Vázquez-Poletti, J. L., Salas, P. J., & Vaquero, L. M. (2013). Cloud Computing for e-Learning:

A Case Study in the University of Granada. . IEEE Transactions on Cloud Computing, 173-181. <https://doi.org/10.1109/TCC.2013.13>.

Villagómez, A., & Fernando, A. (2020). Desarrollo del Sistema Web para la gestión Académica de la Unidad Educativa "Modesto A. PeñaHerrera". Utilizando herramientas VUE.JS y Spring Framework. Repositorio Universidad Técnica del Norte.

Viloria, R., Valles, J., & Morales, R. (2013). Sistemas de Información Académicos en Software Libre: Un Análisis Comparativo. . Revista de Investigación Académica, 57, 1-12.

VMware Tanzu, I. (2023). ¿Why Spring? Obtenido de <https://spring.io/why-spring>

Webb, P., Syer, D., & Long, J. (2019). Spring Boot Reference Documentation. Obtenido de <https://docs.spring.io/spring-boot/docs/current/reference/pdf/spring-boot-reference.pdf>

Welling, L., & Thomson, L. (2016). PHP and MySQL® Web Development. Addison-Wesley Professional.

Zend Technologies, l. (2016). *PHP Best Practice*. Obtenido de https://files.zend.com/help/Zend-Developer-Guide/best_practices.htm

Apéndices

Apéndice A: Formatos de entrevista a los desarrolladores

Determinar a profundidad el contexto del GENESIS SIA por parte de los ingenieros encargados del desarrollo y mantenimiento del software.

1: ¿Que tecnologías se utilizaron para el desarrollo del software Genesis SIA v3?

arquitectura esta implementado en Genesis SIA v3?

2 ¿Cual patrón de a

percibe al momento de dar soporte al software?

3 ¿Que dificultades

sis SIA v3 cumple con los principios SOLID?

4 ¿El sistema Gene

sis SIA v3 permite solucionar incidentes de forma administrativa?

5 ¿El sistema Gene

6 ¿Con cuales métodos cuentan los colegios para reportar incidentes o solicitudes sobre el Genesis SIA v3?

7 ¿El sistema Genesis SIA v3 implementa algún patrón de diseño en su implementación?

8 ¿El Sistema Genesis SIA v3 es adaptable a las necesidades particulares de cada Institución educativa?

Apéndice B: Formatos de entrevista a los rectores

Conocer el contexto de las instituciones para determinar opciones de mejora para el nuevo sistema GENESIS SIA v4

1 ¿Cuál es su nivel de satisfacción con el sistema Genesis SIA v3?

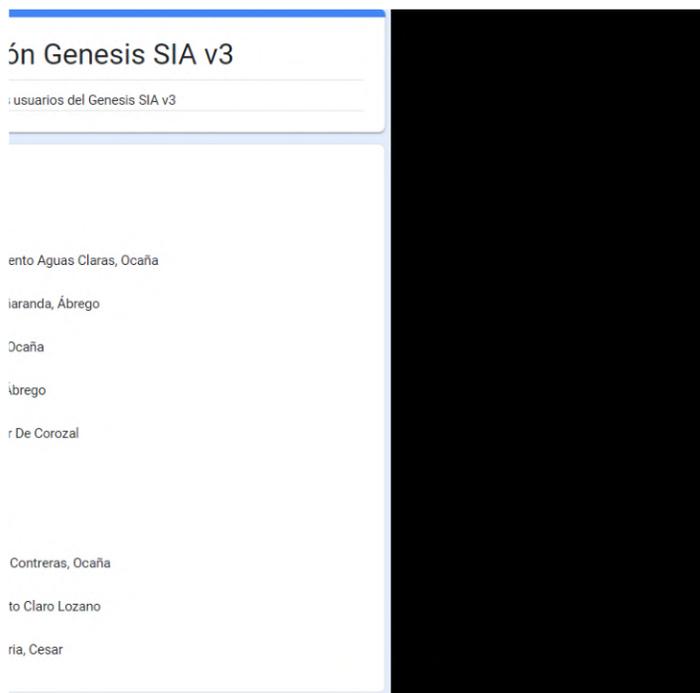
2 ¿Con Cuál sistema evaluativo cuenta la institución educativa?

3 ¿Considera que el sistema Genesis SIA v3 es intuitivo y fácil de utilizar?

4 ¿Considera adecuado el soporte brindado por el proyecto Genesis SIA v3?

5 ¿Qué requisitos nuevos considera que son necesarios implementar en el nuevo sistema Genesis SIA v4?

Apéndice C: Encuesta a los usuarios del sistema Genesis SIA v3.0



Elija su perfil:

- Rector
- Coordinador
- Secretaria
- Profesor
- Administrador de sede
- Estudiante

1. ¿Qué tan satisfecho/a está con el uso que le da al Genesis SIA v3?

- Muy satisfecho/a
- Satisfecho/a
- Neutro
- Insatisfecho/a
- Muy Insatisfecho/a

2. ¿Cómo calificaría la facilidad de uso del Genesis SIA en su jornada laboral?

- Fácil
- Normal
- Difícil

3. ¿Cómo calificaría el soporte que ofrece el Genesis SIA al momento de resolver solicitudes que usted ha realizado?

- Muy satisfecho/a
- Satisfecho/a
- Neutro
- Insatisfecho/a
- Muy Insatisfecho/a

4. ¿Qué tan satisfecho/a está con la confiabilidad del sistema Genesis SIA v3?

- Muy satisfecho/a
- Satisfecho/a
- Neutro

5. ¿Qué tan satisfecho/a está con la apariencia del sistema Genesis SIA v3?

- Muy satisfecho/a
- Satisfecho/a
- Neutro
- Insatisfecho/a
- Muy Insatisfecho/a

6. ¿Con qué frecuencia encuentra errores en el sistema Genesis SIA v3?

- Nunca
- Casi nunca
- Ocasionalmente
- Casi siempre
- Cada semana
- Cada mes

7. ¿Cree necesario la creación de un nuevo sistema información académica Genesis SIA ?

- Muy Necesario
- Necesario
- No Es Necesario

8. ¿Tiene alguna sugerencia o idea para mejorar la experiencia de uso del software?

Texto de respuesta larga

Apéndice D: Repositorio de Figma Genesis SIA v4.0

