Ingeniería en Tecnologías de la Información.

Tesis previa a la obtención del título de:

Ingeniero en Tecnologías de la Información.

Autor:

Santiago Fernando Japon Patiño.

Tutor:

Ing. Darío Javier Valarezo León.

Tema:

Desarrollo de un agente inteligente para la gestión de espacios físicos y visualización dinámica de datos a través de una plataforma web para la Universidad Internacional del Ecuador, campus Loja.

Loja – Ecuador

Septiembre de 2026

Aprobación del Tutor del Trabajo de Integración Curricular

Yo Darío Javier Valarezo León, en calidad de tutor del Trabajo de Integración Curricular denominado: Desarrollo de un agente inteligente para la gestión de espacios físicos y visualización dinámica de datos a través de una plataforma web para la Universidad Internacional del Ecuador, campus Loja. Realizado por Santiago Fernando Japon Patiño ha sido orientado y revisado en todas sus partes y considero que cumple los requisitos establecidos por la institución. En virtud de ello doy mi aprobación a fin de continuar con el proceso académico correspondiente.

Validar únicamente en FirmaEC. Firmado electrónicamente por:
DARIO JAVIER VALAREZO LEON

Darío Javier Valarezo León
1104737679
Tutor del Trabajo de Integración Curricular

Declaración de Autoría y Cesión de Derechos

Yo, Santiago Fernando Japon Patiño declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido presentado anteriormente para ningún grado o calificación profesional y que se ha consultado la bibliografía detallada.

Cedo mis derechos de propiedad intelectual a la Universidad Internacional del Ecuador, para que sea publicado y divulgado en internet, según lo establecido en la Ley de Propiedad Intelectual, Reglamento y Leyes.



Autor: Santiago Fernando Japon Patiño
C.I.: 1104868508

Dedicatoria.

A mis padres, grandes pilares em mi vida. Su esfuerzo constante, sus sacrifícios calados y su convicción incondicional han sido el motor que me impulsó a seguir adelante, incluso en los momentos más difíciles. Mas que su apoyo, han demostrado su ejemplo diario; son sin duda, el modelo a seguir que guía cada uno de mis pasos. Donde cada meta que alcance es este camino de la vida, siempre llevara su esencia, porque no hubiera logrado nada de esto, sin su presencia.

A mi directora de carrea, por su vocación, tiempo invaluable y paciencia. Agradezco su guía en este proceso de mi formación académica, su escucha atenta y conocimientos no solo moldearon mi criterio académico, sino que también me enseñaron a el valor de la empatía y la perseverancia. Gracias por creer en mí y por dedicarse a esta labor.

A mis docentes, por compartir generosamente su sabiduría, y a mis amigos, por su compañía y aliento que hicieron de este trayecto académico sea una gran experiencia.

Agradecimientos.

A la Universidad Internacional del Ecuador, campus Loja, y a la Facultad de ingenierías Digitales y Tecnologías Emergentes, por la información recibida y por las facilidades brindadas para el desarrollo de este trabajo de titulación.

A mi tutor, por su guía académica, su tiempo y sus observaciones oportunas, que orientaros cada etapa de esta investigación; y a los docentes de la Carrera de Ingeniera en Tecnologías de la Información, por los conocimientos compartidos a lo largo de mi formación. Al personal administrativo de la institución, por su colaboración durante el levantamiento de información y las pruebas del sistema, y a mis compañeros de carrera, por el apoyo y el trabajo conjunto en estos años.

Y a mi familia por su apoyo constante hasta la culminación de esta etapa.

Índice de Contenido.

Aprobación del Tutor del Trabajo de Integración Curricular ii

Declaración de Autoría y Cesión de Derechos iii

Acuerdo de Confidencialidad iv

Dedicatoria v

Agradecimientos. vi

Índice de Contenido. vii

Índice de tablas. xiii

Incide de ilustraciones. xiv

Resumen. 9

Abstract. 10

Introducción. 11

Justificación. 13

Objetivos 15

Objetivo General 15

Objetivos Específicos. 15

1. CAPITULO 1: Marco Teórico. 16

1.1. Marco contextual de la problemática. 16

1.2. Descripción y análisis del problema. 18

1.3. Esquema del problema. 20

1.4. Antecedentes y estado del arte. 22

1.4.1. Antecedentes internacionales: sistemas de gran escala. 24

1.4.2. Antecedentes latinoamericanos. 25

1.4.3. ¿Por qué el uso de heurística constructiva? 26

1.4.4. ¿A qué se refiere con optimizar el espacio? 29

1.4.5. Tendencias emergentes: agentes inteligentes y automatización. ... 30

1.5. Agentes Inteligentes. 31

1.5.1. Definición y arquitectura. 31

1.5.2. Tipos de agentes. 32

1.5.3. Aplicación en entornos universitarios. 34

1.6. Gestión de espacios físicos en educación superior 34

1.6.1. Problemática y desafíos. 34

1.6.2. Procesos manuales y automatizados. 35

1.6.3. Indicadores de ocupación y eficiencia. 36

1.7. Programación de horarios académicos. 37

1.7.1. Definición de variables. 37

1.7.2. Restricciones duras y blandas. 38

1.7.3. Complejidad del problema. 39

1.8. Algoritmo de optimización. 39

1.8.1. Métodos que buscan la optimalidad formal. 39

1.8.2. Heurísticas constructivas. 40

1.8.3. Las metaheurísticas. 41

1.8.4. Sistemas multiagente y aprendizaje por refuerzo. 42

1.8.5. Justificacion del enfoque adoptado. 43

1.9. Agentes conversacionales. 43

1.9.1. Definición y función. 43

1.9.2. Procesamiento de leguaje natural. 44

1.9.3. Canales de comunicación. 44

1.10. Automatización de flujos. 45

1.10.1. Low- code y no code. 45

1.10.2. Rol de n8n. 45

1.10.3. APIs y eventos asíncronos. 46

1.11. Procesamiento de documentos. 46

1.11.1. Extracción de datos. 46

1.11.2. Reglas y expresiones regulares. 48

1.11.3. Uso limitado de modelos de lenguaje. 48

1.12. Visualización y analítica espacio-temporal. 49

1.12.1. Mapas de calor y su ayuda a detectar picos y vacíos. 49

1.12.2. Calidad de los datos. 49

1.12.3. Paneles e indicadores. 50

1.12.4. Gemelos digitales. 50

1.13. Arquitectura de software. 51

1.13.1. Microservicios y contenerizacion. 51

1.13.2. API Gateway y control de acceso. 51

1.13.3. JWT y mensajería asíncrona. 52

1.14. Antecedentes y solcuiones existentes. 52

1.14.1. Sistemas manuales. 52

1.14.2. Sistemas institucionales de gran escala. 53

1.14.3. Antecedentes locales. 53

1.14.4. Comparación de soluciones. 54

1.15. Justificación por componente según problema operativo. 56

1.16. Brecha de investigación y aporte. 56

1.17. Métodos aplicados a la investigación. 57

1.17.1. Pertinencia del enfoque mixto: datos de integración. 57

1.17.2. Investigación aplicada. 58

1.17.3. Diseño aplicado. 58

1.17.4. Investigacion y acción. 59

1.17.5. Fases de desarrollo. 61

1.17.6. Técnicas e instrumentos. 62

1.17.7. Población y muestra. 63

1.17.8. Analisis de datos..... 63

1.17.9. Criterios de evaluación..... 64

1.17.10. Plan de pruebas..... 65

1.17.11. Consideraciones éticas y protección de datos. 66

Capitulo II: Análisis y Diseño. 69

2. Requisitos específicos del sistema..... 69

2.1. Requisitos no funcionales..... 11

2.2. Arquitectura del sistema y el diseño de la base de datos..... 13

2.2.1. Arquitectura general del sistema. 9

2.2.2. Arquitectura general del sistema. 9

2.3. Diseño y modelo de base de datos..... 11

2.3.1. Gestion de usuario y roles. 12

2.3.2. Catalogo académico. 12

2.3.3. Planificación de clases y asignación de aulas. 13

2.3.4. Reservas y eventos puntuales..... 15

2.3.5. Incidencias y notificaciones. 15

2.3.6. Relaciones clave..... 15

2.4. Conclusiones técnicas..... 16

3. CAPITULO III: Desarrollo e Implementación..... 17

3.1. Desarrollo, herramientas de construcción e integración..... 17

3.2. Arquitectura general e integración de componentes..... 17

3.3. Herramientas de construcción de la interfaz de usuario. 21

3.4. Motor de inteligencia artificial y algoritmos de asignación. 21

3.5. Reglas de exclusividad y exclusiones fijas:..... 23

3.6. IA generativa: apoyo opcional, invocada desde n8n. 23

3.7. Integración del canal conversacional..... 24

3.8. Infraestructura, despliegue y mantenimiento..... 24

3.9. Plan de pruebas y protocolos de verificación final..... 25

3.10. Objetivos y alcance de las pruebas. 26

3.11. Entorno y estrategia. 27

3.12. Pruebas funcionales. 9

3.13. Pruebas no funcionales. 9

3.14. Pruebas de aceptación de usuario. 11

3.15. Criterios de aceptación y protocolo de verificación final..... 12

CONCLUSIONES..... 14

RECOMENDACIONES. 15

BIBLIOGRAFIA..... 16

Índice de tablas.

Tabla 1 55

Tabla 2 60

Tabla 3 69

Tabla 4 11

Tabla 5 19

Tabla 6 28

Tabla 7 9

Tabla 8 9

Tabla 9 12

Incide de ilustraciones.

Figura 1..... 20

Figura 2..... 9

Figura 3..... 11

Resumen.

Hasta ahora en cada semestre, la gestión de los espacios físicos en la Universidad Internacional del Ecuador, campus Loja, se realizaba con hojas de cálculo diferentes por escuela. Carecía de validaciones y de una visión clara de cómo se usaban las aulas; de mismo modo como coordinar horarios y aulas. Normalmente, la dicha tarea terminaba por fallar en hojas de cálculo esparcidas y el criterio de una a dos personas que terminaban por mantener casi de memoria la planificación de toda una institución en la Universidad Internacional del Ecuador, campus Loja.

Rommie surge en respuesta a esta situación, buscando una salida que no dependa del trabajo manual o de múltiples plataformas. Este proyecto no solamente permite la automatización del proceso administrativo, sino que transforma también el modo de llevarlo a cabo. Combina la heurística de asignación por fases, un canal conversacional de WhatsApp para que docentes y administrativos consulten y reserven espacios sin necesidad de aprender un nuevo sistema y pronto a implementar, además de contar con paneles visuales con heatmaps para mostrar de manera visual el uso real de la infraestructura.

En tal sentido el proyecto ofrece una propuesta de modelo que puede ser replicable para un segmento de la educación superior que las soluciones tecnológicas actuales suelen omitir, demostrando que este tipo de propuestas, puede tener un impacto positivo ante la problemática que se está abordando.

Palabras Clave: asignación de aulas, agente inteligente, automatización universitaria, optimización de espacios académicos.

Abstract.

Until now, each semester, the management of physical spaces at the International University of Ecuador, Loja campus, was carried out using separate spreadsheets for each school. There was a lack of validation and a clear overview of how classrooms were being used; the same applied to the coordination of timetables and classrooms. Typically, this task would end up falling apart due to scattered spreadsheets and the judgement of one or two individuals who ended up relying almost entirely on memory to manage the planning for an entire institution at the International University of Ecuador, Loja campus.

Rommie was developed in response to this situation, seeking a solution that does not rely on manual labour or multiple platforms. This project not only enables the automation of the administrative process but also transforms the way it is carried out. It combines phased allocation heuristics with a WhatsApp chat channel, allowing lecturers and administrative staff to check availability and book spaces without needing to learn a new system a feature set to be implemented soon as well as featuring visual dashboards with heatmaps to graphically display the actual usage of the facilities.

In this regard, the project offers a model that can be replicated for a segment of higher education that current technological solutions tend to overlook, demonstrating that this type of initiative can have a positive impact on the issues being addressed.

Keywords: classroom allocation, intelligent agent, university automation, optimisation of academic spaces.

Introducción.

La planificación académica universitaria lleva a conseguir organizar actividades docentes en periodos de tiempo y en espacios físicos específicos. Este encajar responde a unas restricciones de dos tipos: las de carácter obligatorio, entre las cuales figuran, por ejemplo, la ausencia de solapes, la obtención del aula disponible y la capacidad de aforo seguro; y las blandas, entre las que se encuentran la desviación, la preferencia institucional y el aprovechar recursos. La literatura clásica sitúa estos problemas en el contexto de la planificación de los horarios educativos y nos previene que, cuanto más aumentan las carreras, los paralelos, los docentes, los tipos de aulas, o las condiciones específicas, mayor es la dificultad. Históricamente, la tarea se ha resuelto de forma manual, mediante hojas de cálculo o registros sueltos, con el costo, en horas de trabajo, que eso implica y con una gran propensión a errores humanos. Dos docentes ocupando la misma aula a la vez, o un aula grande ocupada por un grupo poco numeroso son ejemplos de esta peligrosidad (Beyrouthy et al., 2009).

No obstante, en el campus Loja de la Universidad Internacional del Ecuador (UIDE), la información de planificación situaba y recogía en hojas de cálculo que el personal docente elaboraba desde cada unidad académica. Ese procedimiento, si bien permite el registro de datos, no garantiza la validación sistemática, la reconexión de conflictos, la reconexión de los cambios ni la visualización de la ocupación a nivel institucional. De aquí nace el escaso nivel de visualización que se tiene del grado de uso real, generando cuellos de botella en lo operativo y una subestimación sistemática de los recursos institucionales. Este es el motivo por el que esta investigación propone un sistema web que será capaz de transformar la planificación subiendo información estructurada, aplicar reglas de asignación, resolver excepciones y habilitar conductos de

consulta y reservas. Su propósito es incrementar el ratio de uso real de aulas (RUR) mediante una asignación dinámica, disminuir los conflictos en programación y habilitar tableros institucionales de ocupación (*Empty Classrooms, Full Campuses: Rethinking Space Utilization in Higher Education*, n.d.).

Este trabajo hace referencia a un agente inteligente cuando menciona el componente software encargado de recoger información del entorno institucional, interpretarla, aplicar reglas y criterios de decisión, generar una asignación y comunicar la misma. Esta capacidad no depende solo de contar con un modelo generativo: la asignación principal se produce a través de reglas y una función heurística de puntuación, y la inteligencia artificial generativa se utiliza solo como ayuda opcional para identificar estructuras variables de la información contenida en los archivos. Precisamente este aspecto es importante ya que evita dar la impresión de que el sistema es capaz de aprender por sí mismo, que es algo que se queda muy lejos de las posibilidades reales del prototipo.

La tesis está estructurada en seis capítulos. En el primero de ellos se formula el problema, la pregunta de investigación y los objetivos. En el segundo, se desarrolla el estado del arte más los fundamentos conceptuales. En el tercero se define la metodología de investigación y su desarrollo, en el cuarto se plantea la descripción de los requisitos, la arquitectura, el modelo de datos, los algoritmos, los módulos y la seguridad. En el quinto se plantea la estrategia de pruebas, los resultados de la etapa de desarrollo y su discusión. Finalmente, en el sexto se exponen las conclusiones, las propuestas de mejora y líneas de futuro planteadas.

Justificación.

La gestión de los espacios físicos de la Universidad Internacional del Ecuador en Loja se gestiona hasta la fecha a través de hojas de cálculo independientes por escuela y sin validaciones automáticas ni una visión compartida acerca del utilizo efectivo de las aulas, de manera que, en la práctica, dicha planificación acaba dependiendo del criterio de una o dos personas que gestionan la programación institucional. Tal desconexión no es un caso aislado, pues estudios revisados sobre timetabling manual informan tasas de utilización de aulas bajo el 40%. Se puede ver, de este modo, cuánto se pierde en eficiencia cuando el sistema de asignación de espacios no cuenta con una plataforma centralizada.

De este núcleo problemático una gestión tradicional de los espacios físicos en la UIDE, campus Loja se derivan causas concretas: la diversidad de formatos de planificación existente entre escuelas; la asignación de espacios en virtud solamente de la revisión manual; la ausencia de un enfoque institucional de la ocupación en tiempo real; y la falta de estandardización de los canales para hacer reservas y para notificaciones. Y estas causas, a su vez, producen consecuencias directas sobre la comunidad universitaria: reubicaciones o cancelaciones de clases de última hora, eficiencia negativa de la capacidad instalada, retraso en la publicación de los horarios de clases, sobrecarga administrativa que se traduce, al final, en insatisfacción del personal y de estudiantes y docentes.

En este sentido, Rommie se justifica como una respuesta inevitable que no está ni sujeta al trabajo manual ni a múltiples plataformas dispersas, sino que el proyecto combina tres elementos que dan respuesta, cada uno de ellos, a una limitación que se ha identificado: una heurística constructiva por fases para automatizar la asignación de

espacios de forma rápida e interpretable; un canal de mensajería en WhatsApp que permita a los docentes y al personal administrativo consultar y reservar espacios sin necesidad de tener que aprender un sistema nuevo, y paneles con mapas de ocupación que transforman la ocupación real de la infraestructura en información visual accionable para la toma de decisiones.

La importancia de esta combinación es que, aunque existen sistemas de programación académica y prototipos de asignación de aulas documentados en la literatura, ninguno integra estos componentes (gestión de datos institucionales, verificación de restricciones, asignación heurística, consulta conversacional y visualización de ocupación) en un solo sistema diseñado específicamente para el contexto del campus de Loja. En esa línea, Rommie no busca dar respuesta a todas las variantes del problema de programación académica a nivel institucional, sino aportar un modelo auditable y replicable para un segmento de la educación superior al que suelen dejar de lado las soluciones tecnológicas actuales.

Objetivos

Objetivo General.

Desarrollar un agente inteligente para la gestión de espacios físicos y visualización dinámica de datos para la Universidad Internacional del Ecuador campus Loja.

Objetivos Específicos.

- Investigar el estado del arte sobre la implementación de agentes inteligentes conversacionales integrados a WhatsApp.
- Diseñar la arquitectura del sistema que integra el agente inteligente, gestión y almacenamiento de la información, basada en las necesidades institucionales.
- Desarrollar e implementar el agente inteligente para WhatsApp que permita realizar reservas de espacios físicos.
- Implementar la plataforma web de gestión de datos dinámica que permita interpretar el uso de aulas mediante mapas de calor y cuadros de mando.
- Realizar el plan de pruebas y validación del agente inteligente y la plataforma web.

1. CAPITULO 1: Marco Teórico.

1.1. Marco contextual de la problemática.

La distribución de los espacios físicos recorre toda la práctica académica. Una decisión mal tomada puede dar lugar a reservas duplicadas y desplazamientos inútiles, o bien, aulas que generan poca capacidad o un uso inadecuado de los laboratorios, e incluso los propios cambios de última hora que suelen acompañar a estos desaciertos. Estos no son errores anecdóticos: la literatura deja constancia de la falta de aulas disponibles que obliga en cada año académico a derivar cursos de una manera individualizada, trastocando un horario final y generando costos operativos recurrentes; o como la subutilización crónica de los espacios, aulas vacías con escasez percibida, una plaza que pone en evidencia la falta de conexión entre la capacidad instalada y su gestión efectiva; o incluso la información acerca de la ocupación de los espacios, que es clave en la planificación de horarios, mantenimiento, proyecciones de ampliación, distribución de recursos (Duncan L. Hill, 2008).

La característica distintiva de los problemas de programación universitaria consiste en el uso conjunto de restricciones de tipo duro y restricción de tipo blando. Las duras son las que son obligatorias para que se considere válida una solución: por ejemplo, la misma aula no puede acoger dos clases a la vez. En la taxonomía clásica del University Course Timetabling Problem (UCTTP), las restricciones duras primarias que hay que considerar son: (i) una ocupación única de aula por slot temporal, (ii) ausencia de conflictos debidos a la coincidencia de alumnos y docentes, y (iii) correspondiente entre las características del aula y los requisitos del evento. Las restricciones de tipo blando tratan de mejorar la calidad del horario: idoneidad del aula, carga estudiantil lo más

compacta posible, estabilidad de aula para un mismo curso, preferencias de tiempo o espacio (Riis, 2026).

En la UIDE, en el campus Loja, ese procedimiento, tal como reflejan los documentos de trabajo, se llegaba a realizar a través de archivos Excel y una conexión sostenida entre los profesores y el personal de administración. Cada una de las carreras podía tener estructuras distintas, de manera que la consolidación requería revisar encabezamientos y horarios, sumando el número de estudiantes; a eso se le deberá sumar la validación de los tipos de espacio y las posibilidades de cruce. Este modelo descentralizado reproduce el modelo encontrado en la literatura: procesos de timetabling fragmentarios por unidad académica que generan inconsistencias en encabezamientos, formatos y criterios de validación, restando cohesión a la consolidación institucional. A la falta de una fuente central, tanto la consulta en tiempo real como la conservación de un histórico de decisiones verificadas quedan comprometidas, un problema que los estudios de optimización de espacios consideran barrera crítica para incrementar las tasas de utilización RUR (Duncan L. Hill, 2008).

Sustituir una hoja de cálculo por una página web no da respuesta al problema de fondo. El verdadero problema es convertir la información variada del exterior en datos consistentes y utilizar criterios explícitos de asignación a la vez que gestionar las excepciones y trasmitir los resultados a tiempo. La evidencia empírica pone de manifiesto cómo los métodos heurísticos y los métodos de programación de restricciones superan al método manual en términos de utilización del espacio y de satisfacción de preferencias (Ahmad et al., 2018). Para ello, se necesita un artefacto técnico que integre ingeniería de requisitos y diseño de software y reglas de toma de decisiones, visualización entre otros, y que tenga en cuenta la seguridad del sistema y la evaluación de la mismo con los

usuarios finales. Un sistema de apoyo a la decisión (DSS por las siglas en inglés para decisión support system) para timetabling ha de considerar cuáles son las restricciones duras y cualas las duras y debe haber un mínimo de violaciones con respecto a las blandas, tal y como modelan los planteamientos de constraint programming (Ahmad et al., 2018).`

1.2. Descripción y análisis del problema.

El centro del problema consiste en una gestión fraccionada, pero con margen de mejora, de los espacios físicos académicos de la UIDE, campus Loja. En primer lugar, esto se debe a que depende de procesos manuales y carece de validaciones automáticas. Esto no es algo aislado, ya que los estudios comparados muestran que los timetables generados a mano hacen que los espacios previstos para las clases se utilicen menos del 40%, lo cual refleja ineficiencias en la gestión espacial y temporal. Igualmente, la programación manual se vuelve lenta y requiere tiempos de procesamiento prolongados, y demora tanto a los estudiantes y a los administradores. Los conflictos, los cambios improvistos y los retrasos de última hora se hacen más frecuentes a medida que la población estudiantil crece y también lo hace la oferta de curso. Se añade la ausencia de un repositorio que fuera único, donde se pueda unir toda la información de clases y de docentes con las de carreras, horarios y aulas, que provocaba, por lo tanto, que el personal invirtiese tiempo en verificar la disponibilidad y corregir incongruencias detectables antes del envío de sus planificaciones (Theophily Nsulangi et al., 2024).

Las causas identificadas son las siguientes:

- Diversidad de formatos de planificación;
- Asignación basada en revisiones manuales;
- Inexistencia de una visión institucional de ocupación;

- Canales no estandarizados para reservas y notificaciones;
- Limitada trazabilidad de los cambios.

Estas causas explotan un fenómeno descrito como "silos de información": la información sobre scheduling circula en diferentes manos y se alimenta débilmente de forma manual a sistemas independientes; de aquí se producen desalineación e imperfecciones que constituyen amenazas para la eficiencia administrativa. En instituciones con scheduling distribuido, los departamentos poseen sus sistemas separados (Excel, Word, CELCAT), que implican incoherencias en las definiciones de términos académicos, en las cifras de horas de enseñanza y en la calidad. La falta de un inventario de espacios al día y de medidas estándar permite poco gestionar o asignar espacios (*Why Excel for Class Scheduling & Information Silos Are Bad for Student Success*, n.d.).

De esas causas se producen conflictos de horas y asignaciones normales en relación al tamaño del grupo. Con todo, surgen retrasos y las repetidas demandas. La literatura emplea, pues, tales síntomas como expresiones habituales de los riesgos de la gestión manual: y es, por ejemplo, que surgen conflictos de scheduling que requieren múltiples revisiones, lo que dificulta a los administradores tener horarios correctos y ordenados; la subutilización del espacio —las aulas vacías, si la escasez es percibida— expresa la brecha entre la capacidad instalada y su aplicación que los estudios de facilities management achacan a la falta de medición y a la carencia de políticas efectivas; con todo, también se genera la dificultad manifiesta de tener una justificación de por qué ante una clase se le adjudica determinado espacio (Vincent Recreo et al., n.d.).

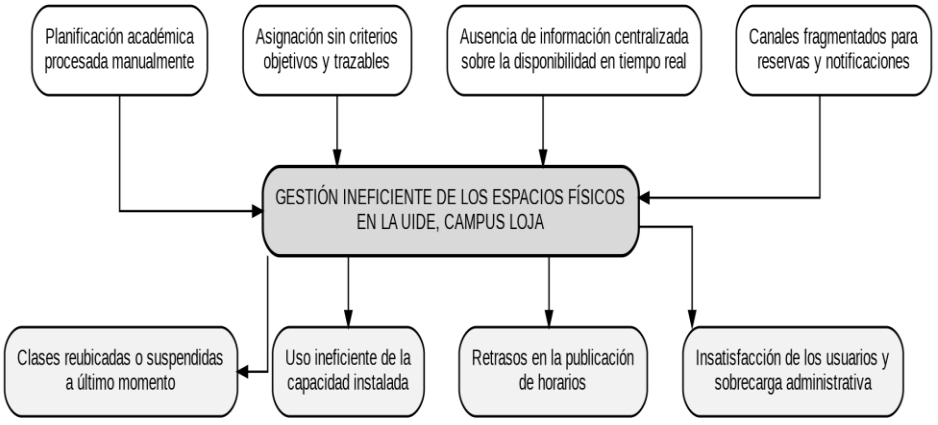
También está presente una dimensión de protección de datos. Habiendo administrado la plataforma, información relativa a estudiantes, docentes y también a

autoridades, debe limitarse las consultas al propósito académico; este principio es coherente, por ejemplo, con marcos normativos como el FERPA (Family Educational Rights and Privacy Act), que establece que la información identificable de los registros educativos no puede ser revelada sin el consentimiento del estudiante, salvo en las excepciones debidamente contempladas; por ejemplo, cuando hay un riesgo de identificación indirecta, la disposición indica que no se han de eliminar solo nombres y números de identificación, sino también las variables personales y otra información que permita considerar rápidamente la identidad del estudiante. Para compartir la información con proveedores o sistemas externos también hacen falta acuerdos escritos que protejan la privacidad del estudiante y limiten el uso de la misma a aquellos fines educativos no autorizados. En particular, la función nombrada "ubicación del docente" ha de ser entendida solo como la consulta del aula correspondiente a una actividad programada, no como geolocalización, no como seguimiento continuo ni como exposición de información ajena al horario académico (*FERPA | Protecting Student Privacy*, n.d.).

1.3. Esquema del problema.

Figura 1

Árbol del Problema.



Nota. Figura 1 muestra como el problema se enlaza con efectos académicos administrativos.

Ahora, para empezar a recorrer esta figura, enfoquemos nuestra mirada en el centro neurálgico de la figura, donde puede apreciarse el problema principal, es decir, el problema del que se trata la figura, justo en el eje que explica toda la figura en el siguiente enunciado: "Gestión ineficiente de los espacios físicos en la UIDE, campus Loja". Es, en consecuencia, el eje que articula toda esta figura. En la parte superior de la ilustración 1 se agrupan las causas raíz que originan esta problemática. Observamos cuatro desencadenantes claros:

1. Como primer aspecto, una "Planificación académica procesada manualmente" que nos indica que se pueden usar procesos ancestrales y dados a errores.
2. Al lado, concurren el hecho de que también existe una "Asignación sin criterios objetivos y trazables", esto es, sin reglas explicitas ni históricos que justifiquen qué decisiones se tomaron.
3. Mucho más abajo en este bloque, la figura reseña la "Ausencia de información para la disponibilidad en tiempo real". Esto es una falta de información sobre qué aulas están libres u ocupadas, "ahora".
4. Por último, terminando este bloque de causas, aparejado con el resto de los datos, encontramos "Canales fragmentados para reservas y notificaciones", es decir, dispersada de la comunicación y gente administrativa.

Ahora en la parte inferior de la ilustración 1. Allí se despliegan las consecuencias directas que se derivan del problema central. Notemos cómo estas afectan tanto a la comunidad educativa como a la propia institución:

1. La primera conclusión que emerge, y que se percibe claramente, es que las "Clases son reubicadas o suspendidas a último momento", generando de esta forma la incertidumbre en los docentes y estudiantes

2. Después de ello, la imagen de la ilustración expresa el concepto "Uso ineficiente de la capacidad instalada", lo que implica que los recursos físicos (aulas, laboratorios, auditorios) no están aprovechando su máxima capacidad.
3. Además, hay evidencias de que "Retrasos en la publicación de horarios", los que generan un efecto dominó que desbarata la planificación de toda la comunidad universitaria.
4. Finalmente, como conclusión de este círculo de acción, la figura del diagrama muestra "Insatisfacción de los usuarios y sobrecarga administrativa", donde el personal queda desbordado y los estudiantes y profesores acumulan quejas.

1.4. Antecedentes y estado del arte.

En esta sección se expone los fundamentos teóricos, conceptuales y tecnológicos sobre los que descansa el proyecto, basado en la automatización de la gestión de espacios físicos, la optimización de la asignación de horarios y la visualización analítica de los datos en un contexto institucional. En el contexto de la educación terciaria actual, la optimización de los recursos y la administración digitalizada se ha convertido en un pilar para incrementar la eficacia operativa y la calidad de servicios lo que constituye un soporte lógico de la estructura que sirva para la gestión de espacios físicos, la optimización de la asignación de horarios y la visualización analítica de los datos en un contexto institucional. En la educación superior contemporánea, se ha vuelto un pilar para elevar la eficiencia operativa y la calidad de los servicios. En el caso concreto de la Universidad Internacional del Ecuador (UIDE), campus Loja, articular agentes inteligentes, orquestadores de flujos y modelos matemáticos de optimización abre la vía

a resolver problemáticas complejas de planificación, reducir los tiempos de operación y disminuir los errores humanos que son propios de la gestión manual.

La programación educativa incluye la asignación de asignaturas, docentes, estudiantes, periodos y espacios. Ningún modelo único es útil para todas las instituciones, de modo que la implementación debe corresponder a las especificidades de cada situación concreta. Para aplicar eficazmente esta metodología de optimización en la UIDE-Loja es preferible estructurar el trabajo en diferentes etapas secuenciales que permitan la adaptación del modelo a la realidad institucional.

Es adecuado iniciar a caracterizar la operación vigente, para luego conocer a los actores participantes y evaluar variables relevantes. Las variables relevantes son el número de grupos, la capacidad real de las aulas y las necesidades de ciertas escuelas. A continuación, corresponde incorporar la visión de los estudiantes por medio de encuestas que permitan obtener, por un lado, sus preferencias de horarios: el bloque de la 7:00 horas a 11:00 horas. y, por el otro, criterios de flexibilidad, que serían aquellas que operarían como restricciones blandas del modelo.

Con la ayuda de la información anterior se realizará el modelado matemático de la optimización mediante programación entera teniendo en cuenta los requerimientos de la universidad más que basarse en un modelo general. Una vez formulado será probado aplicándolo a las facultades donde hay más dificultad para así demostrar su eficiencia. Luego viene la fase final de realizar talleres para la administración donde ellos se ocuparán de otras variables también y así determinar la viabilidad de la operación antes de la implementación de todo el proceso. Otra característica adicional es la de la digitalización del proceso la cual ayuda a cargar automáticamente la base de datos del modelo.

1.4.1. Antecedentes internacionales: sistemas de gran escala.

Los antecedentes directos de esta propuesto admiten dos agregaciones: por una parte, los sistemas de programación académica que han sido constituidos y operados por instituciones de gran escala y, por la otra, los desarrollos de carácter local que han sido producidos por universidades latinoamericanas para intentar resolver la misma cuestión, aunque a menudo de forma acotada en recursos. La diferencia no es baladí: del tipo de diferencia depende de qué modo se transformen aquellos elementos que de cada experiencia son efectivamente transferibles al caso de la UIDE-Loja.

UniTime toma la delantera en la primera categoría, pues se trata del referente más afianzado: un sistema de código abierto para la programación de horarios universitarios tal y como es documentado por Rudová, Müller y Murray (Rudová et al., 2010). Nace de la necesidad de solucionar el problema de programación de cursos de Purdue University, donde se utiliza desde el año 2005 de forma ininterrumpida en todos los problemas de programación que la institución afronta en cada periodo académico. La escala de ejecución refleja el techo de madurez alcanzado por este enfoque: durante el periodo de matrícula de primavera de 2024 el campus principal se ocupó de más de 8.400 clases programadas que corresponden a aproximadamente 3.000 cursos, para más de 50.000 estudiantes.

Se extraen dos lecciones de signo opuesto de lo anterior. La primera lección es que un sistema de programación académica open-source puede mantenerse en producción alrededor de dos décadas y por tanto da apoyo a la viabilidad del modelo de licencia utilizado aquí. La segunda lección es que la funcionalidad de UniTime, que entre otros aspectos considera seccionalización de estudiantes, preinscripción y asesoría académica, corresponde a un tamaño institucional muy diferente al de un campus regional de

veinticinco aulas (Rudová et al., 2010). Utilizar UniTime supondría, por tanto, un costo de implementación muy alto en comparación con el problema que la UIDE-Loja debe resolver.

Sistemas como Celicat, Timetabling.com y Accruent ofrecen características comparables, pero al ser licencias de propiedad tienen costos adicionales y no pueden ser utilizados por organizaciones con presupuestos reducidos (Oude Vrielink et al., 2019). Sin embargo, la tendencia es a un modelo híbrido cuando un motor de optimización con código de código abierto funciona en la plataforma web con una interfaz de usuario personalizada (A. Kumar, 2026).

1.4.2. Antecedentes latinoamericanos.

Dentro de la segunda categoría, el antecedente más cercano al presente trabajo es el que culminó Coronel Cárdenas, desarrollado en la Universidad Politécnica Salesiana, sede Quito, campus Sur. Su diagnóstico se correspondía casi punto por punto con el de la UIDE-Loja: la asignación de horarios y aulas se hacía de forma manual, lo que generaba errores, había que invertir tiempo extra en corregir las novedades y desembocaba tanto en una ausencia de optimización de las aulas como en traslados innecesarios para los estudiantes (Paspuel López & Naranjo Luzuriaga, 2017). Como respuesta se obtuvo puesto un prototipo de un algoritmo genético implementado sobre la base de datos que permitía que la última generación creada constituyese la asignación más eficiente de horarios y aulas.

Torres-Ovalle et al. estudiaron en Universidad de La Sabana un problema real de programación de cursos y asignación de aulas, modelado con programación entera lineal y resuelto mediante un esquema jerárquico de dos fases, con resultados de mejor

utilización de aulas y potencial reducción de recursos y costos (Torres-Ovalle et al., 2014). La propuesta fue validada con datos reales del primer semestre de 2012 y mostró una mejor utilización de las aulas, manteniendo el cumplimiento de restricciones duras y blandas del problema (Torres-Ovalle et al., 2014).

1.4.3. ¿Por qué el uso de heurística constructiva?

Desde el punto de vista metodológico resulta importante marcar el contraste entre ese antecedente y la propuesta que aquí se plantea, debido a que ambos estudios abordan el mismo problema dentro de contextos institucionales semejantes adoptando familias algorítmicas diferentes: Coronel Cárdenas hace uso de una metaheurística de tipo poblacional y Rommie adopta una heurística constructiva por fases; lejos de ser arbitraria.

Los algoritmos genéticos (GA) requieren definir una función de aptitud y configurar operadores de selección, cruce y mutación para evolucionar una población de soluciones candidatas (Alhijawi & Awajan, 2024; Y. Chen, n.d.). La elección de las tasas de mutación y cruce es crítica para el rendimiento del algoritmo, y no existe un valor universal óptimo, lo que impone una carga de configuración inicial (M. Kumar et al., 2010).

La naturaleza estocástica del algoritmo introduce variabilidad entre ejecuciones, ya que la convergencia depende de los números aleatorios generados durante la simulación (Ryder, n.d.). El rendimiento y la convergencia son altamente sensibles a los parámetros de control elegidos, dificultando de esta manera la estandarización de los datos; los parámetros óptimos suelen ser específicos a cada caso, sugiriendo que el diseño experimental es el mejor enfoque para encontrar la combinación adecuada de operadores (Pongcharoen et al., 2001).

Los parámetros óptimos suelen ser específicos a cada caso, sugiriendo que el diseño experimental es el mejor enfoque para encontrar la combinación adecuada de operadores como:

- La sensibilidad a los parámetros y la convergencia prematura son desafíos persistentes.
- La falta de estandarización en los puntos de corte llega a dificultar las comparaciones entre estudios.
- Las configuraciones estáticas predefinidas rinden peor que los enfoques adaptativos dinámicos.

Los algoritmos abordados pueden encontrar soluciones de alta calidad, sin embargo, no garantizan converger al optimo global, sino que se detiene en un mínimo alcance local (Ryder, n.d.). Asegurar la convergencia global con una confianza determinada exige una ejecución suficientemente extensa, donde la cota superior de iteraciones es inversamente proporcional al número de individuos en la población.

Las heurisitcas constructivas se usan precisamente para construir una solución incial mediante reglas de ordenación o selección, y en timetabling educativo suelen basarse en heurísticas de coloreo o prioridades explícitas (Pillay & Özcan, 2017a). En curriculum-based timetabling, un algoritmo voraz puede asignar cursos a salas disponibles con reglas como Largest-First, Best-Fit y “highest unavailable first”, lo que hace visible por qué un curso fue colocado antes que otro (Makalesi Müfredat Tabanlı Ders Çizelgeleme Problemi için Yeni Bir Açgözlü Algoritma Batuhan COŞAR et al., 2023). En un híbrido más reciente, la asignación de aulas se resolvió con una combinación de algoritmo voraz y programación dinámica, separando la decisión temporal de la espacial para mejorar la distribución de recursos (Gama et al., 2025).

El nuevo algoritmo greedy para CB-CTT reportó tiempos de ejecución de pocos milisegundos en sus experimentos (Makalesi Müfredat Tabanlı Ders Çizelgeleme Problemi için Yeni Bir Açgözlü Algoritma Batuhan COŞAR et al., 2023). Mulvey ya mostraba que un modelo de red para aula/tiempo aproxima soluciones rápidamente en problemas realistas donde muchos modelos exactos no escalan (Mulvey, 1982).

En conjunto, la evidencia sugiere que no existe un método universalmente superior para todos los problemas de programación académica, sino que la elección depende de la estructura de las restricciones y del objetivo institucional (Pillay & Özcan, 2017a), Aunque los algoritmos genéticos y sus variantes híbridas destacan por su capacidad de explorar espacios complejos y mejorar restricciones blandas en instancias grandes o multimodales, esa ventaja suele venir acompañada de mayor sensibilidad a parámetros, tiempos de cómputo más altos y menor transparencia operativa (Qiu et al., 2026). En cambio, las heurísticas constructivas han mostrado buen desempeño cuando la prioridad es obtener soluciones factibles, rápidas y trazables, incluso superando tanto a software existente como a algoritmos genéticos en algunos casos aplicados (Kahar & Kendall, 2010).

Por ello, para el caso de UIDE-Loja, donde las restricciones blandas son relativamente acotadas y la necesidad principal es contar con una herramienta ágil, comprensible y útil para la gestión cotidiana, la opción más defendible es una heurística constructiva, dejando las metaheurísticas como alternativa futura si el problema incorpora objetivos de optimización más complejos, como compactación, balance fino de preferencias o criterios operativos adicionales(Hambali et al., 2020).

1.4.4. ¿A qué se refiere con optimizar el espacio?

Una vez revisados los antecedentes, queda muy bien acotada la brecha que aborda el trabajo. Existen los sistemas maduros, aunque dimensionados para instituciones de gran dimensión; existen los desarrollos locales que trabajan la asignación, aunque en este caso son simplemente un prototipo que deja fuera tanto el canal de interacción del usuario final como la visualización institucional de la ocupación. Pero, en definitiva, ninguno de los antecedentes revisados propone una solución en código abierto que haya integrado la ingesta automática de horarios, la distribución automática de aulas, la consulta conversacional de docentes y estudiantes y el panel de ocupación para la toma de decisiones administrativas. En esta integración está la especificidad del aporte de este trabajo (*IEC 62443: Ultimate OT Security Guide | Rockwell Automation | UK*, 2025).

Pero puede surgir la pregunta: ¿Qué realmente implica cuando se dice que el espacio debe ser optimizado cuantitativamente? Diferentes piezas de literatura especializada ofrecen varios tipos de índices. El más popular se llama la Tasa de Utilización del Aula (CUR por sus siglas cortas). Se define como la relación entre las horas efectivamente utilizadas y las horas disponibles durante una semana normal. Por lo general la mayoría de las universidades se esfuerzan por alcanzar un CUR del 65 al 75% para el aula de uso general, lo que significa aproximadamente 30-34 horas de uso por semana en un periodo de 45 horas (de lunes a viernes de 7:00 am a 10:00 pm). (lunes a viernes, 7:00 a.m. a 10:00 p.m.) (*IEC 62443: Ultimate OT Security Guide | Rockwell Automation | UK*, 2025).

El otro que se debe considerar aquí es el porcentaje de ocupación de los asientos que calcula el número promedio de estudiantes presentes en relación a la capacidad del salón. Si un salón tiene un alto RUR pero un bajo porcentaje de ocupación de los asientos

(como por ejemplo un 80% de asistencia pero solo un 40% de asientos ocupados) esto sugiere que el salón está sobrecapacitado y necesita trasladarse a un salón más pequeño (*IEC 62443: Ultimate OT Security Guide | Rockwell Automation | UK*, 2025).

Tableros de uso del espacio como los que se usan en las universidades americanas y europeas incorporan esos factores junto con mapas de calor que ilustran los altos y bajos del uso del espacio por hora y por edificio. Ello les permite identificar ciertas tendencias: ¿Es cierto que hay aulas sin usar los viernes por la tarde? ¿Todas las clases de ingeniería están programadas entre las 8:00 am y las 10:00 am, por lo tanto subutilizando las otras horas del día? Las respuestas a estas preguntas determinaran si la programación necesita ser ajustada (Yaw Dwamena Quansah et al., n.d.).

1.4.5. Tendencias emergentes: agentes inteligentes y automatización.

Si bien la optimización clásica es un enfoque,la literatura reciente se centra en el uso de agentes inteligentes y la automatización basada en la Inteligencia Artificial en la gestión del espacio.Con la ayuda de la inteligencia artificial los sistemas de reservas pueden predecir la demanda futura de las habitaciones basándose en la información histórica y hacer reservas de manera proactiva en vez de reactiva.Por ejemplo,en el caso de una habitación frecuentemente reservada que se cancela el último minuto el sistema indicará "disponible bajo demanda" (*SPACE UTILIZATION & SPACE NEEDS REPORT*, n.d.).

El Accruent es una aplicación de gestión de espacios disponible comercialmente la cual utiliza agentes de Inteligencia Artificial para la gestión de reservas de habitaciones,el análisis de las tendencias de uso con el fin de hacer cambios en la asignación de habitaciones,la resolución de posibles conflictos antes de que estos surjan.

Esta función está más allá del alcance del presente prototipo pero proporciona una tendencia-no solamente la ejecución de reglas sino el aprendizaje de la experiencia (Hasmadi et al., 2026).

En el contexto latinoamericano, sin embargo, la adopción de estas tecnologías es incipiente. La mayoría de las universidades regionales dependen aún de procesos manuales o sistemas heredados (legacy systems) que no se comunican entre sí. Esto genera lo que la literatura denomina "data sprawl": información fragmentada en hojas de cálculo, correos electrónicos y bases de datos desconectadas, lo que dificulta la auditoría, el cumplimiento normativo y la toma de decisiones basada en evidencia (Consulting, n.d.).

La propuesta de este proyecto se encuentra entre dos extremos.No se trata ni de sistemas industriales complejos ni de enfoques académicos simples que no llegan a la etapa de producción.Se trata de un enfoque de código abierto y reproducible que combina la ingestión de datos, la optimización, la visualización y las consultas conversacionales en un solo sistema.La tarea no es solamente abordar el problema actual que tenemos sobre la mesa-el caso UIDE-Loja-sino proporcionar un marco de trabajo que puedan utilizar otras organizaciones de la región (Crocker, n.d.).

1.5. Agentes Inteligentes.

1.5.1. Definición y arquitectura.

En el contexto de esta investigación, un agente inteligente se entiende como un componente de software capaz de recibir información de su entorno, procesarla y ejecutar acciones orientadas al cumplimiento de objetivos específicos. Esta definición se relaciona con el modelo clásico de agentes, en el que las percepciones obtenidas del entorno se

transforman en acciones mediante una función o mecanismo de decisión (Russell et al., 2020). En consecuencia, un agente no debe confundirse necesariamente con un sistema generativo; también puede operar mediante reglas, estados internos, funciones de utilidad o procedimientos de búsqueda.

La arquitectura de un agente inteligente consta al menos de los elementos de entorno,percepción,toma de decisiones y acción. Por ejemplo en el caso de Rommie el entorno está formado tanto por el conocimiento académico como espacial de la UIDE,campus Loja. El sistema obtiene datos sobre las asignaturas,los profesores,el horario,las clases luego utiliza ciertas reglas de validación y asignación para llegar a una solución. Esta limitación es muy importante ya que ayuda a identificar claramente el verdadero alcance del sistema y no exagerar sus capacidades atribuyéndole algunas características de aprendizaje autónomo.

En este caso el agente actúa como un puente entre la información institucional,el motor de asignación y los usuarios.Puede recibir una petición,consultar la información,verificar las condiciones y dar una respuesta.Pero cada paso que da el agente tiene que estar conectado a una regla o un procedimiento previamente establecido.Entonces la independencia del agente no es su libertad para tomar cualquier decisión sino para llevar a cabo un flujo de acciones de forma independiente.

1.5.2. Tipos de agentes.

La literatura especializada clasifica a los agentes según el tipo de información que utilizan y la complejidad del proceso de toma de decisiones. Los agentes reactivos reaccionan solamente al estado actual.Esto es adecuado para acciones inmediatas pero es

limitante cuando se trata de incorporar la historia o cualquier otro estado no perceptible al sistema (Russell et al., 2020).

Los agentes basados en modelos mantienen un estado interno que representa aspectos relevantes del entorno, lo que les permite tomar decisiones cuando la información no está disponible en una sola observación (Schlegel et al., 2021). En entornos parcialmente observables, este estado funciona como una memoria que resume la historia de interacciones y posibilita predicciones y decisiones precisas (Cyun et al., 2025; Schlegel et al., 2021).

Los agentes con un objetivo específico se enfocan en la acción que puede llevar a una situación deseable.En vez de simplemente responder a alguna condición ellos evalúan si la acción ayuda a alcanzar el objetivo. Por ejemplo el objetivo puede ser asignar un salón que se ajuste a la capacidad necesaria sin superponerse con alguna otra clase.

Un paso más que dan los agentes basados en la utilidad es que permiten la comparación de diferentes soluciones. Mediante el uso de funciones de utilidad se le da un puntaje a una alternativa y se elige una solución más conveniente basándose en preferencias o limitaciones suaves. Esta técnica es útil en el caso de la asignación de espacios ya que puede haber muchos salones los cuales puedan servir al propósito pero no todos son igualmente adecuados en cuanto al tamaño,el equipamiento o la cercanía (Russell et al., 2020).

Finalmente, los agentes de aprendizaje modifican su comportamiento a partir de la experiencia y de la interacción con el entorno. Esta categoría no debe atribuirse automáticamente a cualquier sistema que utilice inteligencia artificial. En el caso de Rommie, la asignación principal se basa en reglas y en una heurística constructiva por fases. Por ello, el sistema puede describirse como un agente inteligente orientado a

objetivos y apoyado en reglas, pero no como un agente de aprendizaje autónomo, salvo que el desarrollo incorpore mecanismos específicos de entrenamiento y actualización.

1.5.3. Aplicación en entornos universitarios.

En la educación superior, los agentes inteligentes pueden apoyar actividades que van más allá de la tutoría virtual. Entre sus posibles aplicaciones se encuentran la atención de consultas, la automatización de trámites, la gestión de reservas y el apoyo a la planificación académica. Su utilidad depende de la calidad de los datos, de la claridad de las reglas y de la integración con los sistemas institucionales.

En el contexto de este proyecto el uso del agente inteligente se da para el cumplimiento de tres tareas similares: una la cual se ocupa de la interpretación de datos académicos y espaciales; dos la ejecución de las reglas de asignación y validación correspondientes; y tres la comunicación del resultado a través de canales web y conversacionales. Esta definición permite conectar la idea de un agente con su uso en la vida real.

1.6. Gestión de espacios físicos en educación superior

1.6.1. Problemática y desafíos.

La gestión del espacio físico es otra función transversal dentro de la estructura de la gestión universitaria. Los espacios físicos como aulas,laboratorios y espacios de colaboración deben ser asignados a funciones que se realicen simultáneamente y que puedan tener diferentes requerimientos.En el caso de un laboratorio por ejemplo no se puede considerar igual que un aula regular debido a sus requerimientos especiales.

Existen otros aspectos a considerar además del número de habitaciones vacías que se pueden utilizar.Estos incluyen la capacidad,el equipamiento,la ubicación,el tiempo y

el uso real del espacio.Como resultado puede haber una falta de espacio en algunos periodos aun cuando haya habitaciones disponibles en otros momentos.Los estudios de uso universitario separan la disponibilidad física del uso real de la habitación ya que esta última puede estar físicamente disponible pero en realidad subutilizada la mayor parte del tiempo durante el año (Abdipoor et al., 2023a).

Debido a la información no consolidada se dificulta comprender cómo se está utilizando cada espacio y bajo qué términos.Esto tiene un impacto en la programación de clases,la planificación del mantenimiento y la toma de decisiones de la infraestructura.La gestión del espacio debe tomar en consideración la utilización inmediata y también la generación de información que pueda ser utilizada en el análisis de los patrones de uso.

1.6.2. Procesos manuales y automatizados.

La gestión manual del sistema se basa típicamente en el uso de hojas de cálculo, comunicaciones por correo electrónico, conversaciones telefónicas y la revisión repetida de documentos.Estos medios pueden ser bastante efectivos para organizaciones pequeñas o procesos estáticos pero la eficiencia disminuye cuando aumenta el número de participantes,instructores,clases y limitaciones.Los datos pueden duplicarse fácilmente,quedarse obsoletos o tener varias versiones.

Además, el proceso manual dificulta identificar el origen de una decisión. Si una clase recibe un aula determinada, no siempre queda registrada la razón de la asignación, la persona que la aprobó o el momento en que se produjo el cambio. Esta limitación afecta la trazabilidad y vuelve más compleja la corrección de errores.

La automatización no es simplemente la transferencia de la hoja de cálculo a la página web.Se necesita la centralización de la información,la fijación de reglas,la

validación de las restricciones y el registro de las acciones realizadas.La automatización ayuda a minimizar las repeticiones de ciertas acciones y proporciona respuestas consistentes pero depende en gran medida de la calidad de la información de entrada.

1.6.3. Indicadores de ocupación y eficiencia.

El uso del espacio puede ser estudiado usando medidas temporales y espaciales.El uso temporal implica relacionar las horas de uso de un aula con las horas potenciales de uso mientras que el uso espacial puede implicar comparar el número de asientos ocupados con la capacidad del salón. La medida de la ocupación está definida teóricamente por la ecuación:

$$Utilizacion\ de\ asientos = \frac{numero\ de\ estudiantes\ asignados\ o\ presentes}{capacidad\ del\ aula}$$

Estas variables deben tomarse en consideración como un todo.La habitación puede registrar grandes cantidades de uso pero aún así tener grupos más pequeños. Del mismo modo un aula puede tener una gran capacidad pero estar vacía durante varias sesiones. Por lo tanto el análisis no debe centrarse solamente en la reserva sino que también debe tomar en cuenta el tiempo,la capacidad y la demanda.

La visualización de esta información permite identificar patrones. Esto ocurre, por ejemplo, con las matrices de calor que ayudan a mostrar las franjas que tienen más concentración de actividad y los periodos en que ciertos espacios permanecen vacíos (Islam et al., 2016). En Rommie, esta visualización se plantea como un apoyo para la consulta y la toma de decisiones, no como un sustituto del análisis administrativo.

1.7. Programación de horarios académicos.

1.7.1. Definición de variables.

La programación universitaria se define en la literatura como un problema de optimización combinatoria que asigna actividades o cursos a franjas de tiempo y recursos físicos y humanos (M. C. Chen et al., 2021). En esa asignación intervienen recursos como docentes, aulas, estudiantes, laboratorios o instalaciones docentes, y la solución debe construirse dentro de una oferta limitada de periodos y espacios (Algethami & Laesanklang, 2021).

Una de las variantes más conocidas del university course timetabling problem es el Curriculum-Based Course Timetabling; esta variante forma parte de las principales categorías del problema y recibió una definición formal en la segunda International Timetabling Competition de 2007 (Bettinelli et al., 2015). En este caso, la programación se estructura en torno a currículos o planes de estudio previamente definidos, dentro de un problema de asignación semanal de clases a periodos y aulas (Bonutti et al., 2010). Su objetivo central es evitar conflictos dentro de un mismo currículo, de modo que asignaturas pertenecientes al mismo grupo curricular no queden programadas simultáneamente (Bonutti et al., 2010).

La selección de un modelo debe coincidir con el entorno institucional.No basta con utilizar una fórmula sólo porque se haya aplicado en alguna otra universidad.Se necesita identificar la disponibilidad de los datos,las limitaciones implicadas y el grado de especificidad requerido por la institución.

1.7.2. Restricciones duras y blandas.

Restricciones estrictas son un requisito indispensable. El incumplimiento hace que una asignación sea sin sentido. Tales restricciones son una prohibición de dos actividades asignadas a un mismo profesor durante el mismo periodo, la imposibilidad de dar dos clases al mismo tiempo en un mismo aula y el requisito de que la capacidad del aula sea suficiente para el grupo. La adaptación del aula a la actividad también podría verse como una restricción estricta.

Las restricciones blandas representan preferencias o medidas de calidad del horario. Su incumplimiento no invalida necesariamente una solución, pero sí disminuye su calidad, utilidad o aceptabilidad. Entre las más frecuentes se encuentran las preferencias por ciertos horarios, la minimización de huecos y desplazamientos, la asignación de aulas con capacidad adecuada al tamaño del grupo y la distribución equilibrada de las clases durante la semana (Colajanni & Daniele, 2020). En conjunto, la mejor forma de citar ese párrafo es presentar las restricciones blandas como criterios de calidad y ejemplificarlas con preferencias horarias, compactación del horario, capacidad del aula y equilibrio semanal.

La distinción entre estos dos tipos debería considerarse al diseñar el algoritmo. Las restricciones duras deben ser validadas antes de la aceptación de una asignación. Las restricciones blandas pueden aplicarse para evaluar las opciones disponibles.Esto dará más claridad al proceso y nos ayudará a entender por qué se aceptó una solución en particular o por qué se penalizó una opción en concreto.

1.7.3. Complejidad del problema.

La dificultad del timetabling aumenta con el número de cursos, aulas, docentes, estudiantes, franjas horarias y restricciones que deben satisfacerse simultáneamente (Arias-Osorio & Mora-Esquivel, 2020). El problema consiste en asignar eventos a espacios y tiempos bajo recursos limitados y múltiples restricciones, por lo que se trata de un problema de optimización combinatoria. El espacio de búsqueda crece muy rápido porque cada evento puede combinarse con distintos salones y periodos, y se ha descrito que existen (r^e) formas de asignación cuando se consideran (e) eventos y (r) recursos, con crecimiento exponencial del tiempo computacional al aumentar el tamaño del problema (Bashab et al., 2022). Por ello, versiones primitivas del problema de timetable ya fueron demostradas como NP-completas por Even, Itai y Shamir, y la literatura posterior sobre university course timetabling lo sigue tratando como NP-hard (Abdipoor et al., 2023b).

1.8. Algoritmo de optimización.

1.8.1. Métodos que buscan la optimalidad formal.

Los métodos exactos buscan una solución óptima a partir de una formulación matemática del problema de timetabling (Daskalaki et al., 2004a). En estos casos la programación lineal y la programación entera mixta permiten la formulación de decisiones binarias o enteras como la asignación de clases a periodos de tiempo y aulas mediante variables y restricciones que simbolicen reglas o preferencias operativas o institucionales (Schimmelpfeng & Helber, 2006). Estas formulaciones pueden resolverse con procedimientos exactos como Branch-and-Cut, documentado explícitamente para timetabling universitario, y con solvers de programación entera que implementan

esquemas de búsqueda exacta sobre modelos ILP o MIP (*Computational Study of a Cutting Plane Algorithm for University Course Timetabling – Optimization Online*, n.d.).

Su principal ventaja es que, bajo las condiciones del modelo, estos métodos pueden probar optimalidad, producir soluciones óptimas o entregar cotas de calidad sobre la solución encontrada (Avella & Vasil'ev, 2005). Sin embargo, su limitación central es el costo computacional, que aumenta con el tamaño del problema, el número de variables, eventos y restricciones, y que en varios casos obliga a descomponer el modelo, relajar restricciones o resolver subproblemas más pequeños. Por eso, su aplicación debe evaluarse según el tamaño real de la instancia y los recursos computacionales disponibles, ya que algunos casos pueden resolverse rápidamente hasta optimalidad, mientras que otros requieren enfoques híbridos o por etapas para mantener tiempos razonables (Daskalaki et al., 2004b).

1.8.2. Heurísticas constructivas.

Las heurísticas constructivas generan una solución de manera gradual: parten de una solución vacía y la extienden sucesivamente hasta completar el horario (Pillay & Özcan, 2017b). En timetabling, ese proceso suele implementarse como una asignación secuencial de eventos según un orden de prioridad calculado por reglas heurísticas; para cada evento, la construcción evalúa alternativas factibles y selecciona la que mejor satisface los criterios definidos en ese estado parcial de la solución (Razali et al., 2024a).

Una ventaja de este enfoque es que permite formular reglas de decisión explícitas y comprensibles, porque la prioridad puede definirse mediante criterios transparentes como conflicto, matrícula o dificultad de asignación. En particular, varios

trabajos priorizan primero los eventos más difíciles de ubicar, ya sea por tener más conflictos o por disponer de menos opciones factibles restantes (Razali et al., 2024a). Esa lógica respalda decisiones como asignar antes laboratorios o clases con pocas aulas compatibles y dejar para después los eventos con mayor flexibilidad.

La debilidad de este método es que no asegura la mejor solución global ya que la elección inicial limita las elecciones futuras y crea soluciones no óptimas o incluso parciales de factibilidad (Pillay & Özcan, 2017c). Esta fragilidad secuencial ocurre en los casos donde el proceso de construcción llega a su fin,es decir cuando la mayor parte del cronograma ya ha sido asignado y ya no hay lugares factibles para el resto de las tareas (Razali et al., 2024b). Por eso algunos autores consideran combinar la construcción con la optimización o utilizar varios reglas.

1.8.3. Las metaheurísticas.

Las metaheurísticas son metodologías de alto nivel ampliamente utilizadas para resolver problemas de optimización difíciles, y su diseño busca explorar espacios de búsqueda muy grandes con el fin de encontrar soluciones cercanas al óptimo en problemas complejos (Ezugwu et al., 2021). En este campo suele distinguirse entre métodos de trayectoria única, como simulated annealing y tabu search, y métodos basados en población, como los genetic algorithms. La búsqueda tabú se caracteriza por incorporar memoria en el proceso de búsqueda, mientras que los algoritmos genéticos operan sobre poblaciones de soluciones candidatas y aplican mecanismos de selección, cruce y mutación (Joseph et al., 2022). El recocido simulado pertenece a las metaheurísticas de búsqueda local y se usa precisamente como un mecanismo de exploración para evitar

comportamientos puramente codiciosos en espacios con múltiples mínimos locales (Ruiz-Torrubiano, 2024).

Estos métodos suelen producir soluciones de buena calidad en problemas combinatorios y de optimización donde la búsqueda exacta puede ser poco práctica. Sin embargo, su desempeño depende de decisiones de diseño adicionales, entre ellas la configuración de parámetros, los criterios de terminación, la forma de inicializar la búsqueda y la función de evaluación utilizada. En algoritmos genéticos, esas decisiones incluyen explícitamente el tamaño de la población, las probabilidades de cruce y mutación, y las condiciones de parada (Vega et al., 2022). La evidencia también muestra que estos parámetros pueden alterar de forma importante el rendimiento y hasta la interpretación comparativa entre algoritmos, por lo que su ajuste no debe tratarse como un detalle menor (Kaushik & Nadeem, 2025).

1.8.4. Sistemas multiagente y aprendizaje por refuerzo.

En los sistemas de múltiples agentes las decisiones son tomadas por agentes que representan componentes del problema incluyendo profesores, aulas o materia de estudio. Existen dos formas de coordinar entre agentes incluyendo la negociación y los protocolos.Este enfoque proporciona una forma de representación del problema de manera distribuida; sin embargo la arquitectura requerida para este enfoque podría ser más de lo que se necesita de manera concentrada en un solo campus.

Existen otros enfoques también incluyendo el aprendizaje por refuerzo.En el aprendizaje por refuerzo un agente aprende la selección de acciones basándose en la recompensa obtenida durante la interacción con el entorno.El uso del aprendizaje por refuerzo para la programación académica requiere la definición de

estado,acción,recompensa y condiciones de entrenamiento.Por lo tanto este enfoque no debe confundirse con la ejecución de un modelo de lenguaje y un sistema basado en reglas.

1.8.5. Justificacion del enfoque adoptado.

La elección de una heurística constructiva por fases se justifica cuando el problema exige construir asignaciones en tiempos razonables mediante reglas de prioridad, en lugar de buscar optimalidad exacta en cada instancia (Hussain et al., 2023). En problemas de scheduling, este tipo de enfoque suele organizarse en fases sucesivas de construcción o decisión La fase constructiva puede levantar una solución completa de manera gradual mediante asignaciones sucesivas.

Una estrategia por fases también permite separar el tratamiento de restricciones duras y preferencias blandas (Constantino et al., 2013). En timetabling, la prioridad práctica suele ser encontrar primero una solución factible con costo cero en restricciones duras y luego reducir el costo de restricciones blandas; dividir el procedimiento en etapas puede mejorar el cumplimiento de restricciones duras y el tiempo de procesamiento frente a tratar todos los criterios simultáneamente (Wijaya, 2020).

1.9. Agentes conversacionales.

1.9.1. Definición y función.

Los agentes conversacionales representan programas informáticos que permiten a un ser humano comunicarse usando un lenguaje natural a través de la computadora. Inicialmente los programas conversacionales se basaban en sistemas basados en reglas y respuestas predeterminadas. Las aplicaciones conversacionales modernas son capaces de

emplear enfoques de Procesamiento del Lenguaje Natural para reconocer la intención de la persona que realiza la solicitud así como cualquier información necesaria.

En el caso de Rommie el agente conversacional cumple la función de acceso. La función del agente es dirigir las solicitudes relacionadas con la programación, los salones o la reserva al flujo apropiado. No cumple la función del motor de asignación sino que sirve como interfaz para realizar las solicitudes.

1.9.2. Procesamiento de leguaje natural.

El Procesamiento del Lenguaje Natural permite interpretar el significado del mensaje y conectarlo con la intención. Por ejemplo la pregunta "¿En qué aula se lleva a cabo la clase del martes?" se puede vincular a la intención de saber la ubicación mientras que "Necesito reservar un laboratorio" significaría hacer una reserva.

Además para realizar la tarea el sistema necesita determinar entidades como el nombre del curso, la fecha, la hora, el aula o el profesor. La respuesta solo debería generarse una vez que se haya realizado la validación de los detalles desde la base de datos institucional. Así el lenguaje natural ayuda en el proceso de comunicación pero la confiabilidad de la respuesta depende de la propia base de datos.

1.9.3. Canales de comunicación.

Un canal conversacional hará posible acortar el número de pasos necesarios para hacer una pregunta. Sin embargo la utilidad de usar WhatsApp ,la integración de la web o por correo electrónico debe mostrarse dependiendo del público y el contexto de las instituciones.No se puede decir que un determinado canal tenga la mayor adopción dentro de un ambiente universitario sin ninguna medición. La interfaz web tiene un rol suplementario. Se puede enfocarse en formularios,calendarios, tablas y mapas de calor

mientras que el chatbot se encarga de preguntas simples o inicia flujos predefinidos. Ambos canales deben tener el mismo proceso de autorización y la misma fuente de datos.

1.10. Automatización de flujos.

1.10.1. Low- code y no code.

Las plataformas low-code y no-code proporcionan entornos visuales y componentes reutilizables para construir aplicaciones, integrar servicios y automatizar flujos de trabajo con una menor necesidad de programación tradicional (Waszkowski, 2019). En estos entornos, la lógica suele definirse mediante interfaces drag-and-drop, modelado visual de procesos, reglas de negocio y llamadas a APIs, lo que permite conectar bases de datos, servicios SaaS y APIs de terceros para automatizar tareas e integrarse con sistemas existentes (Sangaiah et al., 2023). Ese uso reduce la cantidad de código necesario, pero no elimina el trabajo de diseñar correctamente los flujos, porque los sistemas de orquestación siguen dependiendo de reglas de ejecución, dependencias, transformación de datos y manejo explícito de errores. La evidencia sobre middleware y orquestación también muestra que los flujos suelen requerir validación de datos, enriquecimiento de eventos, conversión de formatos, reintentos y estrategias de respaldo para mantener interoperabilidad y ejecución confiable entre servicios heterogéneos.

1.10.2. Rol de n8n.

En el proyecto, una herramienta como n8n puede describirse con precisión como una capa de orquestación entre el canal conversacional, la base de datos y los servicios del sistema, porque la orquestación consiste en coordinar y controlar tareas automáticas entre múltiples sistemas siguiendo una secuencia predefinida o activada por eventos (Viljoen et al., 2026). En este proyecto, una herramienta como n8n puede actuar como

orquestador entre el canal conversacional, la base de datos y los servicios del sistema, al recibir eventos, ejecutar validaciones, transformar datos, invocar servicios y devolver respuestas dentro de un flujo definido (Jaidi, 2026). Esta descripción concreta es más precisa que atribuirle una capacidad general de "resolver" el problema, porque la orquestación implementa el flujo de integración y la coordinación entre componentes, no necesariamente la lógica especializada de decisión del dominio.

1.10.3. APIs y eventos asíncronos

Una API permite que los componentes se comuniquen entre sí utilizando contratos predefinidos. Los Webhooks proporcionan notificaciones de la ocurrencia de algunos eventos como la recepción de mensajes y la creación de reservas. La comunicación asincrónica es apropiada en el caso donde una acción tome más tiempo que una consulta habitual, por ejemplo, el procesamiento de múltiples archivos o la creación de una asignación completa.

Es necesario pensar cómo el sistema responderá a los errores,cómo volver a intentar acciones fallidas, sobre posibles mensajes duplicados y sobre conexiones desconectadas. Estos factores son críticos ya que la reserva no debe crearse dos veces cuando el usuario envía una petición nuevamente sin recibir ninguna respuesta.

1.11. Procesamiento de documentos.

1.11.1. Extracción de datos.

Existe una heterogeneidad en la estructura de los archivos de planificación:los encabezados difieren de un departamento académico a otro,no existe un estándar universal para los nombres de las columnas,los campos de fecha pueden estar mezclados(dd/mm/yyyy,mm/dd/yyyy,texto libre)y las anotaciones de la programación

pueden variar desde "08:00-10:00","08h00-10h00" hasta bloques de texto libre.Es necesario distinguir dos fases del proceso de extracción de datos debido a eso. Primero significa el reconocimiento de la estructura del documento a encontrar donde están ubicadas las tablas,que columnas corresponden a que campo y como están codificados los valores.Segundo significa convertir ese contenido en un modelo universal de datos,independientemente del formato original de la información,de manera que pueda ser utilizado por el motor de asignación.

En el caso de las hojas de cálculo, el análisis puede apoyarse en la estructura nativa del archivo: filas, columnas, celdas y tipos de dato. Bibliotecas como openpyxl o pandas permiten leer directamente esa información sin necesidad de interpretación visual. Los archivos PDF, en cambio, presentan un desafío adicional. Aunque pueden contener tablas, la información se conserva principalmente como elementos visuales, sin una estructura tabular explícita que garantice la correspondencia entre celdas y campos. En estos casos, la extracción requiere técnicas de detección de líneas, reconocimiento de patrones de alineación o, en última instancia, el uso de modelos de lenguaje para interpretar el contexto semántico del documento.

En ambos casos el sistema tiene que incluir una capa de validación antes de pasar la información al motor de asignación. La información tiene que pasar la prueba de validación para comprobar si los campos de datos obligatorios como el código de la materia, el nombre del profesor, el número de estudiantes, el horario y el tipo de aula están ahí y si cumplen con las restricciones de formato/dominio. Por ejemplo, ninguna hora puede pasar de las 24:00; el aula tiene que estar en el catálogo de la institución y su capacidad tiene que coincidir con el tamaño del grupo.

1.11.2. Reglas y expresiones regulares.

Las expresiones regulares pueden identificar patrones como códigos de materia, formatos de tiempo o identificadores de aulas. Su fortaleza radica en la eficiencia y la capacidad de probar reglas rápidamente. Sin embargo requieren que los documentos tengan algún grado de regularidad. Por otro lado los heurísticos locales pueden aumentar estas reglas. Por ejemplo se puede intentar buscar encabezados equivalentes,normalizar abreviaturas e identificar la ubicación probable de una columna. Todas las transformaciones deben ser registradas ya que cualquier cambio automático que no pueda ser rastreado podría causar errores difíciles de encontrar.

1.11.3. Uso limitado de modelos de lenguaje.

Los modelos de lenguaje pueden ser utilizados como el modo de respaldo para el procesamiento de documentos que tienen una estructura no uniforme y un contenido ambiguo.En este caso el modelo de lenguaje debe ser proporcionado con instrucciones claramente expresadas para generar una salida estructurada que vaya más allá de los controles de validación.

Además el modelo de lenguaje no debe alimentar al motor de asignación con datos sin validar.La salida debe ser validada contra el esquema,deben verificarse los campos obligatorios y se deben identificar cualquier inconsistencia.De esta manera el LLM ayuda a interpretar el documento pero no toma el lugar de las reglas de control ni decide la asignación de la clase.

1.12. Visualización y analítica espacio-temporal.

1.12.1. Mapas de calor y su ayuda a detectar picos y vacíos.

Los mapas de calor representan la distribución de una variable en el tiempo mediante una codificación visual que hace visibles patrones de actividad y uso (Al-Amri, 2026a). En educación superior, esa visualización puede organizar las actividades por día y franja horaria para mostrar la distribución temporal de las sesiones docentes (Al-Amri, 2026a). Ese tipo de representación permite identificar periodos de alta demanda y momentos de baja o nula utilización.

En gestión de espacios, la literatura sobre timetabling y utilización ya analiza la ocupación de aulas por periodos y por día, y reporta tanto subutilización como sobreutilización de salas (Abdelhalim & El Khayat, 2016a). También muestra que puede haber eventos inactivos o espacios no ocupados dentro del horario disponible. Por eso, un mapa de calor puede usarse para mostrar la cantidad de actividades programadas por aula, día y franja horaria y facilitar la detección visual de concentraciones de demanda y periodos de baja utilización (Al-Amri, 2026b).

1.12.2. Calidad de los datos.

El valor de esa visualización depende de la calidad y actualización de la información de ocupación, porque la falta de visibilidad sobre el uso real de los espacios produce ineficiencias (Abdelhalim & El Khayat, 2016b). La evidencia indica que los espacios pueden aparecer subutilizados o sobreutilizados sin que eso se detecte a tiempo cuando no hay información actualizada sobre ocupación. También indica que la reevaluación periódica y la visibilidad de ocupación activan decisiones de reasignación y reprogramación (Abdelhalim & El Khayat, 2016b).

- Un mapa visualmente claro puede ser engañoso si se basa solo en reservas programadas y no en ocupación actualizada.
- La distinción entre actividad programada y actividad efectivamente realizada importa para evitar decisiones erróneas sobre reasignación de espacios.
- La plataforma debe indicar qué tipo de ocupación muestra, porque la utilidad analítica depende de si visualiza horario planificado o uso observado.

1.12.3. Paneles e indicadores.

Los paneles interactivos permiten monitorear los indicadores de ocupación,conflicto,reserva y disponibilidad.Además pueden revelar los filtros de los salones,edificios,cursos,períodos o tipos de habitaciones.La elección de los indicadores debe proporcionar respuestas a los problemas que deben ser resueltos por el personal administrativo.

La plataforma no puede limitarse a los gráficos solo porque son estéticamente agradables.Cada gráfico debe agregar valor a una decisión específica que se necesita,como encontrar habitaciones poco utilizadas,examinar un periodo ocupado o verificar la distribución de las clases en diferentes espacios.

1.12.4. Gemelos digitales.

El término "gemelo digital" se relaciona con la versión digital vinculada a un objeto o proceso real.Si bien esto puede considerarse un elemento tecnológico que se puede utilizar en el contexto de la administración de un campus,no debería tratarse como una característica de Rommie mientras no existan sensores y no se realicen actualizaciones del estado físico del campus o de su simulación.

Por lo tanto,en la actualidad es más apropiado utilizar la idea de la visualización del análisis de la ocupación. Los gemelos digitales podrían discutirse solamente como una posible característica del futuro.

1.13. Arquitectura de software.

1.13.1. Microservicios y contenerizacion.

La arquitectura de microservicios es un diseño que divide un sistema en componentes independientes que interactúan a través de interfaces claramente definidas. La arquitectura de microservicios se puede usar para dividir el sistema en tales componentes como el servicio de autenticación,el procesamiento de documentos, el motor de asignación, el manejo de reservas y la visualización.

La contenedorización de aplicaciones con la ayuda de Docker te permite incluir todas las dependencias con la aplicación. Así pues es más fácil recrear el entorno de ejecución al desplegar y desarrollar. Pero al mismo tiempo el uso de microservicios trae costos ya que aumenta el número de componentes y requiere monitoreo y una comunicación complicada entre ellos.

1.13.2. API Gateway y control de acceso.

Un "Gateway de API" puede actuar como el punto focal de todas las solicitudes dirigidas a los servicios internos. Los posibles roles que un "Gateway de API" puede desempeñar incluyen el enrutamiento, la implementación de políticas de acceso, la gestión del tráfico y el registro de las solicitudes.

El control de acceso basado en roles permite restringir las acciones dependiendo del perfil del usuario. Un administrador puede administrar las clases y las reservas;un

coordinador puede ver y editar los datos dentro de su jurisdicción;un profesor puede ver sus propios registros de actividad;y un estudiante puede acceder a las consultas.

1.13.3. JWT y mensajería asíncrona.

El token JWT puede ser utilizado como una forma de indicar un proceso de autenticación entre un cliente y los servicios.Debería haber medidas de seguridad que vayan de la mano con su uso como el firmar,la expiración,la revocación y la seguridad de la clave. La comunicación asíncrona permite diferenciar entre procesos rápidos y más lentos.Este enfoque puede ser empleado en la carga de documentos,la creación de programaciones y la construcción de informes.La respuesta recibida del servidor debería indicar un estado comprensible para el usuario como "recibido", "procesando", "completado" o "rechazado".

1.14. Antecedentes y solcuiones existentes.

1.14.1. Sistemas manuales.

Los sistemas manuales representan el punto de partida para la investigación del problema. La principal ventaja de estos sistemas es la baratura inicial y la posibilidad de la modificación de los datos. La principal desventaja de los sistemas manuales es su dependencia del conocimiento de los planificadores y los problemas relacionados con la gestión de versiones,la validación de conflictos y la justificación de las decisiones.

Al comparar este sistema con el automatizado se deben tomar en cuenta solamente los problemas de la velocidad. Sería necesario tomar en cuenta cosas como el número de conflictos,el tiempo invertido en la corrección de los errores, la trazabilidad, la consulta y el uso del espacio.De esta manera la comparación abordará el problema institucional.

1.14.2. Sistemas institucionales de gran escala.

UniTime es un ejemplo apropiado de un precedente ya que se puede ver como un código fuente para la programación académica de una universidad. La información sobre la Universidad de Purdue así como el material proporcionado por Rudová, Müller y Murray permitirá analizar su estructura, sus limitaciones y la experiencia de su uso.

Sin embargo la existencia de un sistema bien desarrollado no implica necesariamente que su adopción sea conveniente para el campus UIDE de Loja. La adaptación del sistema depende del tamaño de la institución, la cantidad de información disponible, la capacidad técnica para mantener el sistema y las funciones que la institución en particular requiera. Esta diferencia permite una comparación más precisa de un sistema general y uno personalizado.

1.14.3. Antecedentes locales.

El trabajo de Cárdenas en la Universidad Politécnica Salesiana es un excelente precedente ya que aborda la programación y asignación de aulas usando un algoritmo genético.La importancia de este trabajo para nuestra investigación no radica en ninguna afirmación de similitud entre este y aquel contexto sino en la posibilidad de comparar el problema,las entradas,las limitaciones,el algoritmo y sus resultados.

La comparación metodológica debe hacerse con cuidado. El algoritmo genético busca entre una población de soluciones,lo que requiere una función de aptitud así como parámetros evolutivos. La heurística constructiva asigna los eventos incrementalmente siguiendo alguna lista de prioridades. Ambas metodologías pueden ser buenas pero tienen demandas diferentes.En Rommie el uso de la heurística constructiva tiene que ver con la necesidad de obtener resultados replicables y comprensibles para la administración.

1.14.4. Comparación de soluciones.

La comparación de las herramientas debe basarse en datos verificados. Las características de software como uPlanner,uBooking, Quodus, etc.deben ser estudiadas ya sea en la documentación oficial o en las investigaciones académicas.No deben hacerse afirmaciones sobre el costo, la escalabilidad o la arquitectura de la solución a menos que la fuente utilizada las verifique.Se puede utilizar una tabla para la comparación pero sus criterios deben diferenciar entre las características verificadas y las que se van a implementar en Rommie.

Es fundamental sustentar la comparación de herramientas en fuentes fiables, ya que el hecho de atribuir características, costes o niveles de escalabilidad sin respaldo documental podría llevar a conclusiones erróneas sobre la conveniencia de cada solución frente a las necesidades reales de la institución. Al sustentar todo el análisis comparativo únicamente en documentación oficial y trabajos académicos, se asegura la rigurosidad técnica del mismo y se posibilita una valoración objetiva de las alternativas existentes en el mercado frente a la propuesta desarrollada.

Como se puede apreciar en la Tabla 1, la comparación entre las características comprobadas de cada herramienta y aquellas planificadas para ser implementadas en Rommie, permite establecer un punto de comparación claro y justificado, evidenciando las ventajas particulares que ofrece la solución propuesta frente a alternativas como uPlanner, uBooking y Quodus, en el contexto específico de la UIDE.

Tabla 1

Comparación de soluciones.

Criterio	Sistemas manuales	Plataformas comerciales	Rommie
Fuente de datos	Archivos y comunicaciones separadas	Repositorio institucional de acuerdo con la plataforma	Repositorio definido para el proyecto
Validación	Revisión humana	Reglas integradas, según la solución	Reglas y heurística constructiva
Gestión de reservas	Solicitudes por canales diversos	Módulo institucional	Flujo web y conversacional previsto
Analítica	Limitada o manual	Paneles según licenciamiento	Paneles de ocupación previstos
Trazabilidad	Dependiente de las versiones guardadas	Registro de operaciones según la plataforma	Bitácora diseñada para el sistema
Costo y despliegue	Bajo costo inicial	Debe verificarse por proveedor	Depende de la infraestructura disponible

Nota. La información de esta tabla trata sobre la comparación entre ciertas alternativas para llegar a una solución.

1.15. Justificación por componente según problema operativo.

La selección de la tecnología debe guiarse por los problemas y las limitaciones del proyecto. La heurística constructiva por fases cumple con el requisito de obtener respuestas rápidas,fiables e interpretables. El orquestador de flujo cumple con el requisito de conectar los canales de entrada,la validación y los servicios. El chatbot cumple con el requisito de responder preguntas. Las heatmaps cumplen con el requisito de comprender la ocupación.

La selección de un analizador híbrido se justifica por la variedad de formatos. Las reglas nos permitirán procesar rápidamente los formatos conocidos mientras que el sistema de respaldo estará ahí para cualquier variación inesperada. En ambos casos los datos extraídos deben ser validados.

La arquitectura web permite el acceso centralizado y el compartir una fuente de información. Pero se debe tomar en cuenta el aspecto de la seguridad de antemano.Se debe proveer la autenticación,la autorización basada en roles,el registro de operaciones y la protección de los datos personales.En especial la función "ubicación del profesor" solo debe usarse para ver el aula de la actividad planeada y no considerarse como geolocalización o rastreo.

1.16. Brecha de investigación y aporte.

La revisión nos permite reconocer la necesidad de considerar una brecha. Existen sistemas generales de programación académica así como prototipos que usan algoritmos de optimización para la asignación de aulas. Sin embargo los antecedentes revisados tienen que ser comparados desde las perspectivas de su funcionalidad, contexto, grado de implementación y la presencia de alguna evaluación.

El enfoque de Rommie puede considerarse como la combinación de componentes que dentro del campo revisado no han sido combinados en un solo sistema diseñado para el campus de Loja:el manejo de datos institucionales, la verificación de restricciones, la asignación heurística de aulas, la búsqueda a través de un agente conversacional, la programación y la visualización de la ocupación. La afirmación debe formularse de una manera que refleje el resultado de la revisión más que como una afirmación definitiva y global de inexistencia.

Es importante aclarar la contribución. Rommie no pretende resolver todas las variantes de la programación de horarios, los sistemas institucionales de la academia o ser autónoma en su proceso de aprendizaje. Más bien el objetivo de Rommie es abordar un proceso particular de gestión y planificación del espacio en la UIDE, campus Loja,a través de una solución que pueda ser contabilizada,auditada y analizada.

1.17. Métodos aplicados a la investigación.

1.17.1. Pertinencia del enfoque mixto: datos de integración.

El enfoque mixto resulta pertinente porque combina elementos cuantitativos y cualitativos para obtener una comprensión más amplia y profunda del problema de investigación (Almalki, 2016a). Este tipo de diseño es adecuado cuando un proyecto integra mediciones técnicas con información proveniente de los usuarios, ya que permite trabajar con datos numéricos y con evidencia interpretativa dentro de un mismo estudio (Robinson, 2007). El aspecto cuantitativo puede ayudar a analizar el tiempo de respuesta,la resolución de conflictos,las reservas procesadas y los datos de usabilidad. Por otro lado el enfoque cualitativo permite comprender las necesidades del personal y los

problemas que existen actualmente en el proceso y su aceptación de la solución propuesta (Almalki, 2016b).

La integración de ambos enfoques debe definirse de manera explícita en el procedimiento, porque integrar datos no es un detalle secundario, sino un componente central del diseño mixto (Fetters et al., 2013a). No basta con afirmar que la investigación es mixta; es necesario indicar qué datos se recogerán, en qué momento se analizarán y en qué etapa se conectarán o combinarán, ya sea durante el diseño, la recolección, el análisis o la interpretación (Fetters et al., 2013b). La literatura distingue varias formas de integración, como connecting, donde una base de datos enlaza con la otra, building, donde un tipo de datos informa la recolección del otro, merging, donde ambos se reúnen para el análisis, y embedding, donde se articulan en múltiples puntos del estudio. Por ejemplo, los registros del sistema pueden mostrar el tiempo requerido para procesar una solicitud, y las entrevistas pueden explicar por qué los usuarios encuentran difícil un determinado paso, lo que corresponde a una lógica de integración en la que los resultados de un enfoque ayudan a interpretar o desarrollar el otro (Ivankova et al., 2006)

1.17.2. Investigación aplicada.

1.17.3. Diseño aplicado.

El proyecto puede describirse como una investigación aplicada con desarrollo tecnológico y validación empírica, porque en ingeniería de software e IS la investigación de diseño se centra en conceptualizar el problema, diseñar una solución y validarla en contexto (Wieringa, 2014a). Esa orientación busca resolver problemas concretos mediante la construcción de un artefacto de software y su evaluación, en lugar de limitarse a describir fenómenos. En esta tradición, el proceso suele iterar entre

investigación del problema, diseño del tratamiento y validación del tratamiento, con verificación explícita de que el diseño y la evidencia permiten las inferencias previstas (Wohlin & Aurum, 2014).

Las pruebas y la retroalimentación permiten ajustar progresivamente el sistema, porque los ciclos de diseño iteran entre construcción, evaluación y refinamiento del artefacto (Stickland et al., 2024). La evaluación no debe asumirse como un cierre tardío, sino como parte del desarrollo y del uso del artefacto en contexto real (Wieringa, 2014b).

1.17.4. Investigacion y acción.

La denominación de investigación-acción debe reservarse para estudios con participación y colaboración sustantiva de los practicantes en el proceso de cambio e investigación (Cordeiro et al., 2017a). La literatura coincide en que su rasgo definitorio es un proceso cíclico explícito, formulado como diagnóstico, planificación o intervención, evaluación u observación, y reflexión o aprendizaje (Davison et al., 2004). También exige que esos ciclos estén documentados, porque importa registrar qué acciones se realizaron, cuándo ocurrieron y cómo informaron la siguiente iteración.

Si esos ciclos no están definidos en el protocolo, resulta más preciso hablar de desarrollo experimental aplicado o de investigación tecnológica con evaluación de usuarios. Esa formulación encaja mejor cuando el foco principal está en construir y validar un sistema con usuarios, aunque haya iteraciones y despliegue en contexto real, sin llegar a una lógica formal de acción-reflexión participativa institucional (Runeson et al., 2020).

Tabla 2

Justificacion de la Investigacion

Formulación	Cuando usarla	Soporte
Investigación aplicada con desarrollo tecnológico y validación empírica	Cuando se construye un artefacto para resolver un problema y se evalúa en contexto (Peffers et al., 2007)	Diseño ciencia / DSR
Investigación tecnológica con evaluación de usuarios	Cuando hay pruebas, feedback y ajuste iterativo del sistema (Barcellos et al., 2022a).	Iteración y evaluación
Investigación-acción	Solo si hay ciclos explícitos de diagnóstico, intervención, observación/evaluación y reflexión con participación institucional (Cordeiro et al., 2017b)	AR canónica

Nota. Opciones de denominación metodológica según el tipo de proceso descrito.

Este proyecto podría considerarse una investigación aplicada la cual involucra el desarrollo y la verificación de la tecnología y está destinado a resolver un problema práctico a través de la construcción de un artefacto de software cuya calidad y adecuación se adapte por medio de pruebas y la retroalimentación de los usuarios (Barcellos et al., 2022b).Solo entonces podría considerarse una investigación-acción si el protocolo establece ciclos explícitos de diagnóstico,intervención,observación o

evaluación y reflexión en los cuales los actores de la institución participen activamente (Borg, 2011).

1.17.5. Fases de desarrollo.

La primera fase corresponde al levantamiento de requisitos. En ella se revisan los archivos de planificación, se identifican los actores, se documentan los flujos actuales y se registran las restricciones. Las entrevistas y la observación deben aplicarse a participantes que conozcan directamente el proceso, mientras que la revisión documental debe permitir contrastar lo declarado con la forma en que realmente se trabaja.

La segunda fase comprende el diseño de la arquitectura y del modelo de datos. En esta etapa se definen los servicios, las entidades, las relaciones, los roles y las interfaces de comunicación. Las decisiones deben documentarse mediante diagramas y especificaciones que permitan comprobar la correspondencia entre requisitos y componentes.

La tercera fase comprende la construcción del motor de asignación, el parser, la plataforma web, el agente conversacional y los flujos de integración. Cada componente debe desarrollarse de manera verificable. En particular, el motor debe devolver no solo una asignación, sino también información suficiente para explicar las restricciones evaluadas y la razón de la selección.

La cuarta fase corresponde a las pruebas. Deben incluirse pruebas unitarias, de integración, funcionales, de seguridad, de rendimiento y de usabilidad, según el alcance real del proyecto. Las pruebas deben utilizar casos definidos previamente y conservar evidencias de los resultados.

La quinta fase comprende la validación piloto. Antes de afirmar que existirá un despliegue en producción, debe confirmarse que la institución autoriza ese uso y que existe un periodo concreto de evaluación. Si esto todavía no está decidido, se puede describir como una validación controlada o como una prueba piloto en un entorno institucional.

1.17.6. Técnicas e instrumentos.

La técnica de entrevista semiestructurada permite la recolección de información sobre los desafíos del proceso y las regulaciones que no siempre se reflejan en los registros. Los grupos focales son útiles para obtener percepciones comunes aunque se deben reclutar participantes con alguna experiencia en este sentido. La técnica de observación permite tomar nota de los procedimientos del proceso manual incluyendo los pasos, el tiempo, las revisiones y los puntos de decisión.

La técnica de encuesta estructurada puede ayudar a evaluar la satisfacción, la percepción de facilidad y la aceptación. Los registros del sistema proporcionan información sobre la solicitud, los errores, el tiempo y el estado de las operaciones. Cada técnica debe contener las variables que serán analizadas y a veces las técnicas necesitan ser pilotoadas o evaluadas por expertos.

La aplicación de la escala SUS puede ser útil para medir la usabilidad percibida de la plataforma (Lewis, 2018). Sin embargo, una puntuación objetivo no debe presentarse como resultado antes de realizar la medición, porque la SUS es un instrumento de evaluación cuyo puntaje se obtiene a partir de las respuestas de los usuarios y se usa posteriormente para interpretar la usabilidad observada. El valor de 70 puntos puede utilizarse como referencia interpretativa o de benchmark, pero la evidencia proporcionada

aquí lo presenta como guía para la interpretación y comparación de puntajes, no como garantía previa de que el sistema alcanzará ese nivel.

1.17.7. Población y muestra.

La población se determinará basándose en aquellos que tengan interacción con el proceso ya sea directamente o indirectamente. Los coordinadores y el personal administrativo están involucrados en la planificación y gestión de los espacios. Los docentes hacen uso del horario y necesitan información acerca de sus actividades. Los estudiantes reciben el impacto de la asignación y pueden hacer uso de los medios de consulta dentro del alcance definido.

El número de participantes debería determinarse de acuerdo a la población y las autorizaciones adquiridas.En caso de no haber una determinación exacta del número de coordinadores o usuarios sería imprudente manifestarlo definitivamente. Esto puede presentarse como una estimación al principio la cual luego se justifica a través de datos institucionales.

Se podría utilizar la muestreo de conveniencia en el caso de la investigación aplicada de alcance institucional aunque se debe tener en cuenta que esto reduce la generalizabilidad de los resultados. Los participantes serán seleccionados basándose en la experiencia con el proceso y la disposición a participar.

1.17.8. Analisis de datos.

La información cuantitativa se puede presentar en términos de frecuencias, porcentajes,media,mediana y desviación estándar. La selección de la prueba estadística depende del número de sujetos,la naturaleza de los datos y el tipo de comparación que se

deba hacer.No se puede seleccionar de antemano una prueba t sin considerar si sus supuestos se cumplen o no.

Al comparar un proceso manual con un proceso automatizado se debe seleccionar la misma tarea o una similar para ambos.Además sería útil mencionar el número de observaciones tomadas para cada condición de medición.De esta manera ayudará a comprender si la diferencia encontrada se debe a la automatización o no.

Los datos cualitativos pueden ser analizados mediante un análisis temático,un enfoque comúnmente utilizado en la identificación,el análisis,la organización y la presentación de los temas o patrones en el conjunto de datos.Esto implica familiarizarse o leer múltiples veces los datos para generar códigos iniciales después de los cuales los códigos son categorizados para formar los temas o categorías analíticas (Ahmed et al., 2025; Nowell et al., 2017).Los temas o categorías analíticas luego son refinados y nombrados y se redacta el resultado.El análisis temático también puede ser descrito como iterativo y reflexivo ya que el investigador puede regresar a etapas anteriores a medida que analiza los datos.Es imperativo que el codificar sea específico de los objetivos del estudio y la pregunta de investigación ya que los temas están destinados a capturar información importante del conjunto de datos en relación a la pregunta de investigación (Mwita & Mwilongo, 2025).

1.17.9. Criterios de evaluación.

Debe haber una separación de las mediciones técnicas,funcionales y de la experiencia del usuario. Los ejemplos de las mediciones técnicas pueden ser el tiempo de procesamiento, la disponibilidad y la tasa de errores.En cuanto a las mediciones funcionales estas consisten en la satisfacción de las restricciones duras,no tener reservas

duplicadas y la correcta coincidencia de curso,el profesor,el horario y el salón. Los objetivos propuestos deben ser discutidos antes de que se conviertan en medidas definitivas. Por ejemplo una afirmación como "el motor no debería tomar más de 60 segundos" solo tiene sentido si se basa en la carga de trabajo esperada,el hardware usado,el tamaño del archivo y el número de eventos. De la misma manera una tasa de éxito de las reservas que supere el 95% tiene que ser definida claramente primero.

La frecuencia de los conflictos tampoco debería expresarse como "por debajo del 100% "ya que eso no muestra ninguna mejora en absoluto.Es mejor calcular cuántos conflictos se identificaron antes y después de la investigación y fijarse la meta de minimizar el número de conflictos difíciles en la etapa final.

Al medir la usabilidad se puede considerar el índice SUS, el tiempo de realización de las tareas y los errores de interacción. Al medir la satisfacción será mejor utilizar las escalas de Likert o el CSAT. El NPS también puede tomarse en cuenta si es necesario.

1.17.10. Plan de pruebas.

Las pruebas para la unidad deben cubrir operaciones aisladas como la validación de la capacidad, la comprobación de superposición y la interpretación de los formatos. Las pruebas de integración deben incluir la verificación de la comunicación entre el analizador, la base de datos, el motor de asignación, el chatbot y la plataforma web.

Para las pruebas de aceptación se deben utilizar casos realistas. Los sujetos y las actividades de las pruebas deben definirse de antemano. Las pruebas pueden comprobar los registros de cumplimiento, los errores detectados, la retroalimentación de los usuarios y los cambios implementados después de la prueba.

Las pruebas de carga solo deben realizarse si están dentro del alcance y si existe una infraestructura para ejecutarlas. Se debe proporcionar una justificación para el número de solicitudes paralelas basándose en la carga real. También se deben realizar mediciones de los recursos como la CPU, la memoria, la latencia y la tasa de errores, pero los resultados no deben presentarse antes de que se hayan realizado las pruebas.

1.17.11. Consideraciones éticas y protección de datos.

Todas las actividades relacionadas con las entrevistas, los grupos focales, las pruebas de usabilidad y la fase piloto deben realizarse con el consentimiento de los participantes. La declaración debe cubrir el propósito,la duración,el riesgo y los beneficios,la voluntariedad y cómo se manejarán los datos obtenidos.

Los datos de la institución y las personas solo deben usarse para el fin para el cual fueron destinados.En cuanto a la función de "ubicación del profesor" la aplicación solo debe revelar el aula donde se va a llevar a cabo la clase. Nunca debe obtener o revelar la ubicación del profesor de otra manera.

La protección técnica debe incluir control de acceso, cifrado durante la transmisión, gestión segura de credenciales, registros de auditoría y políticas de conservación. La afirmación de que los datos residirán en servidores seguros debe acompañarse de una descripción de la infraestructura que efectivamente se utilizará.

También debe revisarse la normativa aplicable en Ecuador, en particular la Ley Orgánica de Protección de Datos Personales, que constituye el marco jurídico específico para el tratamiento de información personal y para la tutela de este derecho en el país (Pineda & Ruiz, 2025). La literatura sobre Ecuador describe que este marco regula el tratamiento de datos mediante principios y lineamientos para su uso, tratamiento,

almacenamiento y circulación, e incluye exigencias de seguridad, confidencialidad y transparencia (William Villa Quishpe et al., 2023). En ese sentido, el diseño del sistema debe considerar la finalidad del tratamiento y limitar los datos a lo estrictamente necesario para cumplirla, porque la minimización aparece como un criterio central especialmente cuando se manejan datos sensibles o en entornos de gobierno electrónico.

También debe incorporar medidas de seguridad y resguardo, dado que varios trabajos subrayan riesgos de exposición, ciberataques y brechas de implementación de la ley en contextos digitales y organizacionales. Asimismo, la conservación y gestión de los datos debe analizarse según el tipo de información tratada, ya que la literatura ecuatoriana ha enfatizado la necesidad de regular expresamente el tratamiento, la conservación y la explotación de datos por instituciones públicas y privadas (Martínez-Jara & Pérez-Ycaza, 2022; Roldán Carrillo, 2021)

El marco teórico y tecnológico necesario para la construcción de Rommie se ha proporcionado en este capítulo. El problema relacionado con la gestión de los espacios físicos en la UIDE, campus de Loja se ha especificado como un problema de programación académica con la presencia de restricciones duras y blandas, con respecto a la heterogeneidad de la entrada y el requisito de la reproducibilidad en las asignaciones.Se han analizado diversas estrategias de optimización, que van desde algoritmos exactos hasta enfoques metaheurísticos, seleccionándose un enfoque heurístico constructivo como estrategia de solución.

En una línea similar los componentes del modelo arquitectónico-el administrador de conversación,el administrador de flujo,el administrador de asignación,el analizador híbrido y las ventanas de visualización-han sido explicados junto con la funcionalidad de cada uno en relación con otras soluciones y ejemplos existentes.Finalmente se ha

discutido la metodología detrás del diseño lo cual permitirá una evaluación de la solución propuesta en base a la eficiencia, la usabilidad y la satisfacción así como los problemas éticos y de protección de datos asociados con la investigación. Estos son los ingredientes que ayudan a sentar las bases para el desarrollo del sistema el cual será explicado con mayor profundidad en los próximos capítulos.

Capitulo II: Análisis y Diseño.

2. Requisitos específicos del sistema.

Los requisitos funcionales consisten en las actividades fundamentales que realizará Rommie para poder cumplir con sus objetivos los cuales son ayudar en la planificación, programación y consulta de espacios físicos en la Universidad Internacional del Ecuador (UIDE), Campus Loja. Estas actividades incluyen la validación, las actividades internas, las respuestas y el almacenamiento de la información en la base de datos.

La validación se produce cuando se comprueba que cada espacio está disponible, tiene capacidad y es exclusivo, antes de confirmar una asignación, lo cual evita conflictos de horario entre distintas solicitudes. Las actividades internas son los procesos automáticos del sistema como la generación de propuestas de distribución y la actualización en tiempo real del estado de los espacios. Las respuestas son las notificaciones y confirmaciones que envía el sistema al usuario por cada consulta o solicitud que se haga. Finalmente, el archivo asegura que toda la información generada sea almacenada de forma estructurada en la base de datos, lo cual permite su posterior consulta y cumple el rol de respaldo histórico de la gestión de espacios. En la Tabla 3 se detallan estas actividades.

Tabla 3

Requisitos específicos.

ID	Nombre	Descripción
RF01	Carga de planificación académica	El director de carrera debe poder cargar su planificación académica por periodo. La operación comprende validar el archivo subido, leer los campos clave (materia, paralelo, número de estudiantes, horario y tipo de aula), almacenarlos de forma estructurada en la base de datos y confirmar visualmente la carga exitosa o mostrar advertencias de error.
RF02	Asignación automática de aulas	Motor de distribución maestra (Node.js/Express, distribucion.service.js): procesa planificaciones, compara demanda de aulas con disponibilidad real según capacidad, tipo y exclusividad, evita cruces de horario entre materias y docentes, y genera una propuesta automática de asignación guardada en base de datos.
RF03	Resolución de conflictos por aula o laboratorio	Corresponde al sistema detectar de forma automática los casos en que dos o más carreras soliciten el mismo espacio en un horario determinado. Para dirimirlos aplicará criterios como prioridad de carga, historial, horarios alternativos y compatibilidad. Cuando la intervención manual resulte necesaria, notificará al administrador.
RF04	Visualización del horario académico	Docentes y directores han de consultar sus horarios asignados, verificar las aulas correspondientes a cada materia, revisar la duración, el tipo de clase y el número de estudiantes, y exportar o imprimir los horarios cuando lo requieran.

RF05	Generación de estadísticas	La plataforma web generará gráficos por número de materias, jornada y uso de aulas, mapas de calor que reflejen los espacios y horarios más utilizados, informes descargables y acceso restringido según el rol del usuario.
RF06	Interacción con el bot de WhatsApp	El sistema validará al usuario por nombre y carrera, habilitará consulta de disponibilidad de aulas en tiempo real, información sobre la ubicación actual de un docente (aula en la que se encuentra dictando clases), gestión de solicitudes de reserva de aulas y confirmación automática de que la reserva se generó con éxito.
RF07	Gestión de reservas	En caso de que un estudiante requiera una reserva puntual, el sistema verificará la disponibilidad real del aula, comprobará que no medien restricciones de uso, registrará la reserva con los datos del solicitante y notificará el registro satisfactorio.
RF08	Administración de conflictos no resueltos	Al administrador del sistema le corresponde visualizar los conflictos que el motor automático no resolvió, adoptar decisiones de asignación manual y forzar reasignaciones si la autoridad institucional lo dictaminara.
RF09	Control de acceso por rol	El sistema mostrará funcionalidades diferenciadas según el rol del usuario (estudiante, docente, director o administrador) y restringirá el acceso a reportes y funciones críticas mediante un inicio de sesión autenticado.
RF10	Registro y trazabilidad	Toda acción relevante del sistema quedará registrada: cargas de planificación, asignaciones automáticas, reservas exitosas o fallidas, consultas mediante el bot y conflictos detectados junto con su resolución.

Nota. se redactan los requerimientos específicos que debe tener el sistema.

2.1. Requisitos no funcionales.

Tabla 4

Requisitos no funcionales.

ID	Nombre	Descripción
RNF01	Tiempo de respuesta del bot de WhatsApp	Ninguna consulta dirigida al bot mediante WhatsApp debe superar los 3 segundos de respuesta.
RNF02	Tiempo de carga de la plataforma web	Las visualizaciones de horarios y gráficos de la plataforma web han de cargarse en menos de 5 segundos.
RNF03	Tiempo de procesamiento del motor de asignación	En cuanto al motor de asignación automática, procesar la planificación completa de una escuela debe tomarle menos de 60 segundos.
RNF04	Capacidad de usuarios simultáneos	El sistema debe sostener al menos 100 usuarios simultáneos sin que su estabilidad ni su rendimiento se vean afectados.

RNF05	Procesamiento asíncrono de conflictos	El procesamiento de conflictos, por su parte, se ejecutará de forma asíncrona con el fin de evitar bloqueos en la interfaz de usuario.
RNF06	Compatibilidad del cliente web	Cliente: navegador web compatible con HTML5/JavaScript (Chrome, Firefox, Edge) sobre Windows 10 o superior, con acceso óptimo mediante plataforma web responsiva desde móviles y tablets.
RNF07	Stack tecnológico, backend.	Backend: Node.js 18 LTS con Express 4.18 como framework principal y Sequelize 6 como ORM.
RNF08	Sistema de gestión de base de datos	Base de datos: PostgreSQL 15 o superior.
RNF09	Orquestación y automatización	Orquestación y automatización: n8n autoalojado (vía Docker) junto con Redis 7 para la gestión de colas, flujos de asignación, validación de datos y comunicación mediante Evolution API v2.3.7.
RNF10	Tecnologías complementarias del frontend	Tecnologías complementarias: frontend en React 18 (Vite, TypeScript, Tailwind CSS), dashboards con Chart.js/Recharts, IA con OpenAI GPT-4 y Anthropic Claude, e integración opcional con SSO institucional (OAuth2/LDAP).
RNF11	Recursos mínimos del servidor	Servidor (recursos mínimos): procesador de 2 vCPU, memoria RAM de 4 GB, almacenamiento de 40 GB SSD y conectividad de 100 Mbps.

RNF12	Recursos mínimos del cliente	Cliente (recursos mínimos): procesador Intel i3 o equivalente, memoria RAM de 4 GB, navegador web actualizado y dispositivo móvil con la aplicación de WhatsApp actualizada para el uso del bot.
RNF13	Diseño modular y mantenibilidad	El sistema (Node.js, Express, React) requiere mantenimiento preventivo mensual (actualización de paquetes, respaldos, depuración de base de datos y revisión de flujos n8n) y correctivo trimestral (ajustes de reglas de negocio, nuevas funciones y mejoras visuales), a cargo de un desarrollador o administrador técnico con formación en el stack del proyecto.

Nota. se especifican los requerimientos no funcionales sobre el sistema.

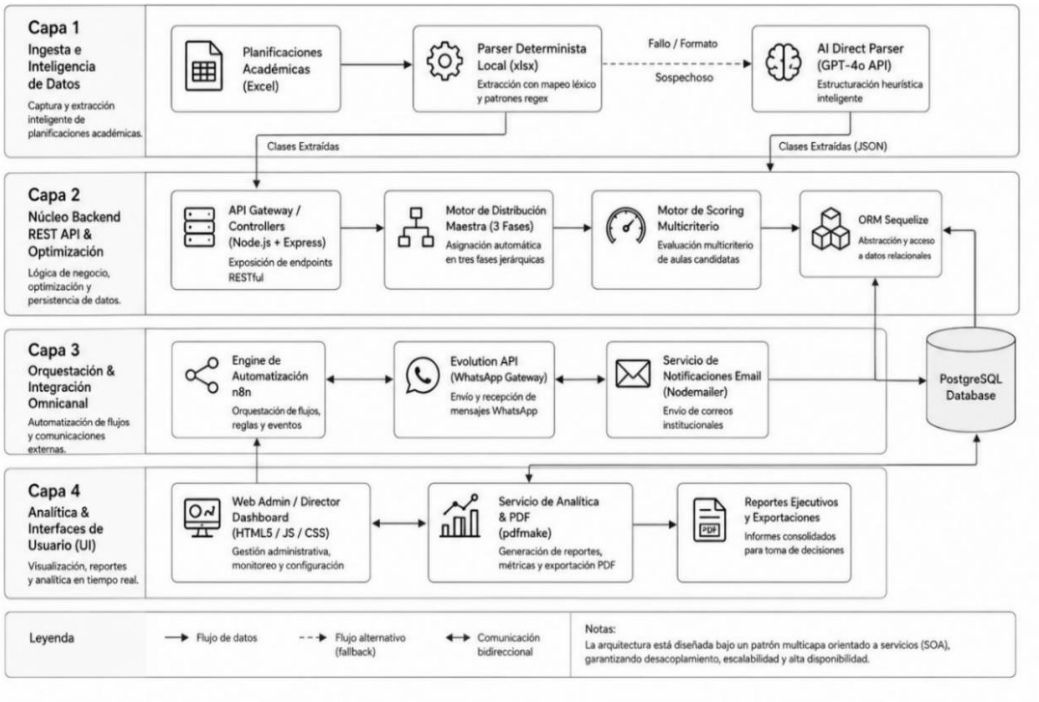
2.2. Arquitectura del sistema y el diseño de la base de datos.

Rommie es un marco de trabajo orientado hacia la planificación académica, la automatización del aula y la optimización, la gestión de reservas así como el análisis espacial y temporal de la estructura física de la universidad.Este capítulo describe el estado actual del sistema arquitectónico el cual está estructurado de acuerdo a un enfoque de arquitectura orientada a servicios (SOA) con procesamiento desacoplado e integración de la inteligencia artificial junto con el diseño de la base de datos relacional en PostgreSQL.

2.2.1. Arquitectura general del sistema.

Figura 2

Arquitectura de cuatro capas.



Nota. En la figura 2 se especifica la arquitectura multicapa del sistema.

2.2.2. Arquitectura general del sistema.

Rommie adopta una arquitectura multicapa orientada a servicios (SOA), caracterizada por el procesamiento desacoplado y la comunicación omnicanal en tiempo real. El sistema se organiza en cuatro capas, cada una con responsabilidades diferenciadas:

- La primera capa se ocupa de la ingesta y el procesamiento inteligente de los datos. En condiciones normales, emplea un analizador sintáctico determinista instalado localmente para extraer las planificaciones académicas desde archivos con un formato predefinido. No obstante, cuando el documento presenta desviaciones o variaciones imprevistas en su estructura, el sistema activa un mecanismo de

contingencia basado en el modelo de lenguaje GPT-4o, el cual se encarga de interpretar la organización del contenido y estandarizar los datos para asegurar su consistencia.

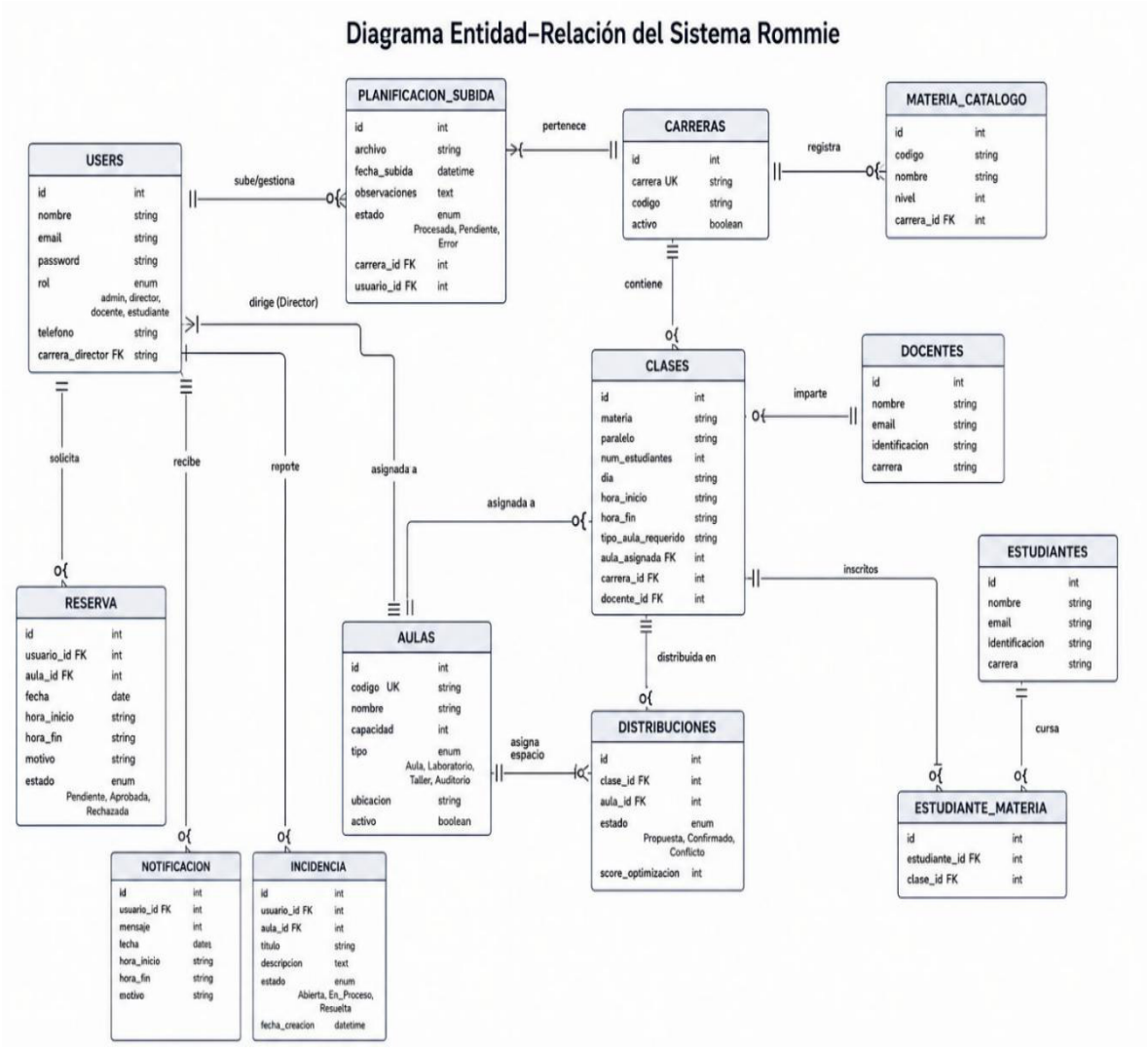
- El segundo nivel constituye el núcleo del backend, donde confluyen la API REST y los procesos de optimización. Esta capa está compuesta por un API Gateway desarrollado en Node.js/Express, un motor de distribución principal que ejecuta su flujo en tres fases sucesivas, un motor de puntuación multicriterio que sopesa las distintas alternativas de adjudicación y, por último, el ORM Sequelize, encargado de gestionar la persistencia y el modelo de datos en PostgreSQL.
- El tercer nivel se centra en la orquestación y la integración multicanal. Para ello, se apoya en el motor de automatización n8n, que sincroniza y gestiona los flujos de comunicación entre la Evolution API de WhatsApp, el sistema de notificaciones vía correo electrónico y los eventos que emite el *backend*. Gracias al enfoque asíncrono, las solicitudes se procesan en segundo plano, lo que garantiza que la interfaz de usuario no se vea afectada ni pierda capacidad de respuesta.
- El cuarto nivel se dedica al análisis de datos y a la interacción con los usuarios. Agrupa un panel de gestión web que posibilita la consulta de horarios, distribuciones de aulas e incidencias, junto con un módulo de análisis estadístico y generación de informes en formato PDF. Los cuadros de mando incluyen visualizaciones dinámicas, entre las que sobresalen los mapas de calor, que permiten apreciar el nivel de ocupación de los espacios según los bloques horarios y la tipología del aula.

La figura 3 representa gráficamente esta arquitectura de cuatro capas. El desglose por componente y mecanismo de integración figura en el capítulo III; se trata, por tanto, de una misma arquitectura descrita en dos niveles a detalle.

2.3. Diseño y modelo de base de datos.

Figura 3

Modelo de entidad-relacion de la base de datos.



En la figura 3 se detalla gráficamente el modelo entidad-relacion de la base de datos.

El siguiente esquema representa el modelo entidad-relación del sistema de gestión académica y asignación de aulas. Su diseño abarca desde el registro de usuarios

y la estructura curricular hasta la planificación de clases, la asignación de espacios y la gestión de incidencias y notificaciones. A continuación, se detalla cada entidad y su propósito dentro del ecosistema.

2.3.1. Gestion de usuario y roles.

- USERS.

  Almacena las credenciales de acceso al sistema incluyendo nombre, correo electrónico y contraseña. Esta tabla constituye la base para la autenticación y el control de roles. Los perfiles específicos (docente, estudiante, administrador) se gestionan en entidades separadas que referencian a esta tabla mediante una clave foránea.

- DOCENTES.

  Contiene la información personal y académica de los profesores, tales como nombre, correo electrónico, número de identificación y carrera de pertenencia. Se relaciona con la entidad CLASES para indicar quién imparte cada asignatura, estableciendo una relación uno a muchos entre docente y clases asignadas.

- ESTUDIANTES.

  Registra los datos de los alumnos, incluyendo nombre, correo electrónico, número de identificación y carrera. Su vínculo con las reservas y con la matrícula se establece a través de otras tablas, como RESERVA o CURSA, que modelan la relación entre estudiantes y actividades académicas.

2.3.2. Catalogo académico.

- CARRERAS.

  Define las carreras ofertadas por la institución. Cada registro incluye un código

único (clave única), un nombre descriptivo y un indicador de estado (activo/inactivo). Esta entidad constituye la raíz para organizar las materias y los estudiantes, ya que todas las asignaturas y matrículas se asocian a una carrera específica.

- MATERIA_CATALOGO.

  Contiene el listado maestro de asignaturas disponibles. Cada registro especifica código, nombre, nivel académico y una clave foránea (carrera_id) que la asocia a la carrera correspondiente. Esta tabla funciona como fuente de verdad para las materias que posteriormente se programan en la entidad CLASES.

- CURSA.

  Tabla puente que relaciona estudiantes con carreras, modelando la relación de pertenencia o matrícula. En el diagrama actual solo se muestra carrera_id, pero se recomienda incorporar estudiante_id para asegurar la trazabilidad alumno-carrera y completar la relación muchos a muchos entre ambas entidades.

2.3.3. Planificación de clases y asignación de aulas.

- AULAS.

  Registra los espacios físicos disponibles para actividades académicas. Cada registro incluye código único, nombre descriptivo, capacidad máxima, tipo de espacio (teórica, laboratorio, taller, etc.), ubicación física y estado (activo/inactivo). Esta entidad representa el recurso principal que se asigna en los procesos de distribución.

- CLASES.

  Constituye la entidad central de la planificación academica. Agrupa la infromacion de cada sesión programada, incluyendo:

  - Datos de la materia (nombre, paralelo, número de estudiantes).
  - Franja horaria (día, hora de inicio y hora de fin).
  - Requerimientos de aula (tipo_aula_requerido).
  - Claves foráneas hacia la carrera, el docente y el aula finalmente asignada (aula_asignada).

  Cada registro de CLASES representa una oferta de clase concreta en un periodo académico determinado. Se recomienda normalizar el campo materia para que referencie a MATERIA_CATALOGO mediante clave foránea en lugar de almacenarlo como texto.

- DISTRIBUCIONES.

  Almacena el resultado del proceso automático de asignación de aulas. Cada registro incluye una referencia a la clase (clase_id), un estado (Proyectado, Confirmado, con Conflictos) y una puntuación de optimización (score_optimizacion). Esta información permite evaluar y auditar la calidad de la distribución generada por el motor de asignación.

- DISPONIBILIDADES.

  Esta tabla presenta una estructura similar a DISTRIBUCIONES (clase_id, estado, score_optimizacion), lo que sugiere una posible duplicación o un uso diferenciado, como disponibilidad temporal versus asignación definitiva. Se recomienda aclarar su propósito específico o evaluar su fusión con DISTRIBUCIONES para evitar redundancias en el esquema.

2.3.4. Reservas y eventos puntuales.

- RESERVA.

  Permite a los estudiantes solicitar un aula para un uso extraordinario fuera de la planificación regular. Cada registro almacena el usuario solicitante (usuario_id), la fecha solicitada, el intervalo horario, el motivo de la reserva y el estado actual (pendiente, aprobada, rechazada). Esta tabla opera de manera independiente de la asignación regular de clases, aunque debe verificar la disponibilidad del espacio antes de confirmar la reserva.

2.3.5. Incidencias y notificaciones.

- INCIDENCIA.

  Registra cualquier problema o conflicto reportado por los usuarios (docentes, directores o administradores) relacionado con el uso de espacios, choques de horario, fallas en equipamiento u otros inconvenientes. Cada registro incluye fecha, hora, motivo detallado, usuario reportante y estado de resolución.

- NOTIFICACION.

  Almacena los mensajes enviados a los usuarios a través de diversos canales (correo electrónico, WhatsApp, notificaciones en plataforma). Cada registro especifica el destinatario (usuario_id), la fecha de envío, el canal utilizado y el contenido del mensaje. Aunque comparte algunos atributos con INCIDENCIA, su función es distinta: comunicación activa hacia el usuario, no reporte de problemas.

2.3.6. Relaciones clave.

El modelo establece las siguientes relaciones fundamentales:

- **USERS DOCENTES / ESTUDIANTES**: Aunque no se muestra explícitamente en el diagrama, se asume que existe una relación de referencia entre la tabla de usuarios y los perfiles específicos, posiblemente mediante el correo electrónico o el número de identificación.
- **CLASES AULAS, CARRERAS, DOCENTES, MATERIA_CATALOGO**: La entidad CLASES mantiene claves foráneas hacia el aula asignada (aula_asignada), la carrera oferente, el docente responsable y la materia correspondiente. Sin embargo, en el diagrama actual el campo materia en CLASES parece ser un campo de texto en lugar de una clave foránea. Se recomienda normalizar esta relación para garantizar la integridad referencial.
- **DISTRIBUCIONES y DISPONIBILIDADES CLASES**: Ambas tablas dependen de CLASES para vincular la asignación con la planificación académica. Cada registro de distribución o disponibilidad debe referenciar una clase válida.
- **RESERVA e INCIDENCIA USERS**: Ambas entidades se asocian a la tabla de usuarios mediante usuario_id, lo que permite identificar quién solicitó la reserva o reportó la incidencia.

2.4. Conclusiones técnicas.

La configuración arquitectónica final dota a Rommie de una infraestructura desacoplada y con alto potencial de escalabilidad. Gracias a la sinergia entre el analizador sintáctico híbrido, el motor de asignación multicriterio y la estructura de datos optimizada, el sistema procesa la planificación académica institucional en su totalidad en cuestión de segundos; como resultado, se suprimen los conflictos de horario y se habilita un análisis gerencial continuo y en tiempo real.

3. CAPITULO III: Desarrollo e Implementación.

3.1. Desarrollo, herramientas de construcción e integración.

Rommie es una plataforma integral destinada a planificar, distribuir y gestionar en tiempo real los espacios académicos de la Universidad Internacional del Ecuador, sede Loja. Su construcción conjuga cuatro elementos: una arquitectura desacoplada de servicios contenedorizados, un motor de distribución maestra sustentado en un algoritmo heurístico constructivo determinista en tres fases que se ejecuta en el backend, la orquestación de efectos secundarios mediante n8n, esto es notificaciones, reportes y enriquecimiento opcional con inteligencia artificial, y un canal conversacional multicanal implementado sobre WhatsApp. Los apartados siguientes detallan las decisiones de ingeniería, las herramientas de construcción empleadas y los mecanismos de integración entre componentes.

3.2. Arquitectura general e integración de componentes.

En este proyecto se implementa una arquitectura de servicio distribuido que utiliza Docker Compose.Esto permite que los entornos utilizados en el desarrollo,la prueba y la producción sean consistentes.Desde el lado del cliente existe una aplicación de página única escrita usando React y TypeScript así como también aplicaciones móviles que se comunican con el asistente conversacional.Todo el tráfico pasa por un proxy inverso Nginx donde se realiza la terminación SSL/TLS antes de llegar al núcleo de la aplicación escrito usando Node.js y Express.

La persistencia la realiza PostgreSQL que es el sistema de gestión de bases de datos relacionales principal mientras que Redis actúa como almacenamiento de colas y caché para apoyar los procesos de inferencia y automatización. Los caminos de

orquestación para los efectos secundarios,notificaciones,reportes programados y opcionalmente con enriquecimiento de inteligencia artificial los realiza n8n que no tiene ninguna lógica de asignación;utiliza los servicios de la inteligencia artificial generativa a través de la API REST y consulta la base de datos cuando es necesario.

La pasarela Evolution API, por último, expone la interactividad con WhatsApp hacia un microservicio dedicado que se comunica con la base de datos mediante un cliente nativo de PostgreSQL. En la tabla 5 se exponen los componentes del sistema y sus mecanismos de integración.

La arquitectura del sistema está diseñada para integrar de manera coherente, escalable y mantenible múltiples componentes tecnológicos, permitiendo que cada servicio cumpla una función específica sin crear dependencias innecesarias entre sí. Esta decisión de diseño tiene como objetivo garantizar que el sistema sea capaz de evolucionar de forma modular, facilitando tanto la incorporación de nuevas funcionalidades como el mantenimiento de los componentes existentes, al tiempo que asegura un flujo de comunicación ordenado entre el cliente, el servidor y los servicios de soporte.

Esta distribución de responsabilidades entre los distintos componentes permite al sistema operar de forma desacoplada, de tal forma que un fallo o una actualización en un servicio específico no afecte al funcionamiento general de la plataforma. La integración mediante contenedores, por otra parte, permite un despliegue más ágil y sienta las bases para una escalabilidad futura del sistema, a medida que se vayan incorporando nuevos módulos o aumente la demanda de uso dentro de la institución.

Tabla 5

Componentes del sistema.

Componente	Tecnología	Integración principal
Frontend SPA	React 18 + TypeScript + Vite	HTTPS / REST con JWT hacia Nginx
Proxy inverso	Nginx (puertos 80/443)	Terminación SSL/TLS y enrutamiento /api al backend
Núcleo de aplicación	Node.js 18 LTS + Express, gestionado por PM2	Sequelize ORM hacia PostgreSQL; webhooks REST hacia n8n
Base de datos	PostgreSQL 15	Accedida por el backend, n8n y el microservicio de WhatsApp
Colas y caché	Redis 7	Encolamiento de tareas de automatización e inferencia
Motor de automatización	n8n	Webhooks REST, consultas SQL y llamadas HTTP a servicios de IA
Servicios de IA generativa	OpenAI GPT-4o / Anthropic Claude	Consumo REST desde los flujos de n8n
Pasarela de mensajería	Evolution API v2.3.7	Webhooks HTTP hacia el microservicio del bot
Microservicio conversacional	Node.js (whatsapp-bot-aulas)	Cliente pg hacia PostgreSQL y coordinación con el backend
Notificaciones	Nodemailer sobre SMTP institucional	Envío de correo electrónico desde el backend

En la Tabla 5 de exponen los componentes del sistema.

La lógica de negocio se implementó sobre Node.js en su versión 18 LTS (OpenJS Foundation, s.f.), con Express.js (v4.18.2) para el diseño modular de rutas, middlewares de validación y controladores transaccionales. De la persistencia se encarga PostgreSQL

15 (PostgreSQL Global Development Group, s.f.), motor relacional que soporta restricciones avanzadas, claves foráneas e índices de rendimiento; el acceso se canaliza a través del ORM Sequelize (v6.35.0), que habilita el modelado orientado a objetos, la ejecución de migraciones controladas con sequelize-cli, el manejo de transacciones y la definición de relaciones de tipo hasMany, belongsTo y belongsToMany.

En cuanto a la seguridad la autenticación se realiza mediante tokens JWT firmados digitalmente los cuales tienen periodos de expiración programados (jsonwebtoken v9.0.2) mientras que las credenciales están encriptadas a través de funciones de generación de hash de bcryptjs (v2.4.3) con un factor de costo de diez iteraciones.El uso de Helmet (v7.1.0) y express-rate-limit (v7.1.5) refuerzan la superficie de exposición y contrarrestan las vulnerabilidades expuestas por la Fundación OWASP (2021)en cuanto a CORS,Políticas de Seguridad de Contenido,Ejecución del Lado del Cliente,Transporte Seguro y Ataques de Fuerza Bruta. También debemos incluir nuestro middleware para la sanitización de cadenas el cual fue creado para evitar la inyección de código SQL y la inyección de secuencias en el navegador.

A lo anterior, la capa de servidor suma los siguientes módulos de integración y servicios auxiliares:

- xlsx (v0.18.5): analizador de planificaciones académicas y cupos entregados en formato de hoja de cálculo.
- Nodemailer (v8.0.1): gestor de notificaciones por correo institucional mediante conexión SMTP segura.
- pdfmake (v0.3.3): generador de reportes ejecutivos y horarios descargables en formato PDF.

3.3. Herramientas de construcción de la interfaz de usuario.

El frontend se concibió como una aplicación de página única (SPA) desarrollada con React 18.3.1, estructurada en componentes funcionales y respaldada por hooks personalizados —como useAuth y useFetch— que centralizan la lógica de autenticación y el consumo de servicios. Como empaquetador se optó por Vite 5.4.5, que reemplaza a las herramientas tradicionales gracias a su recarga en caliente y su compilación incremental, lo que acelera notablemente los ciclos de desarrollo. Todo el código está escrito en TypeScript 5.6.3, lo que garantiza un tipado estático riguroso tanto para los modelos de datos como para los contratos definidos con la API del servidor.

El diseño visual se apoya en Tailwind CSS 3.4.13, un marco de utilidades que facilita un maquetado receptivo, la alternancia entre tema claro y oscuro, y una coherencia estética basada en las convenciones de interfaz de macOS. Los iconos, vectorizados y accesibles, proceden de Lucide React y React Icons. La navegación corre a cargo de React Router DOM v6.26.0, que gestiona el enrutamiento en el cliente y protege las rutas según los roles predefinidos: administrador, director de carrera, docente y estudiante. Por último, la comunicación con el servidor se canaliza mediante Axios 1.7.7, configurado con interceptores que renuevan y añaden automáticamente el *token* de autorización en las cabeceras HTTP pertinentes.

3.4. Motor de inteligencia artificial y algoritmos de asignación.

La definición formal de la asignación del aula es el problema de la optimización combinatoria bajo restricciones NP-difíciles llamadas problema de programación y problema de programación de horarios respectivamente en la literatura (Even et al.,1976; Schaerf,1999). Debido a que la exploración por fuerza bruta del espacio de soluciones no

puede ser llevada a cabo debido a la escala del problema con respecto al número de asignaturas, profesores y aulas de la institución el sistema utiliza un enfoque heurístico constructivo determinista basado en un algoritmo de tres fases,complementado con una puntuación de criterios múltiples y una fase de reintento que permite la relajación de las restricciones de capacidad. El algoritmo de distribución maestra funciona únicamente en el backend en el servicio.distribucion.js servicio.

- Fase 1 (aulas especificas: En esta primera etapa se da prioridad a las materias que requieren instalaciones con equipamiento o condiciones particulares (laboratorios, talleres, salas de audiencias y practicas), puesto que su catálogo de espíritus viables es el mas reducido.
- Fase 2 (clases especificas): A continuación, se procesan los grupos compuestos por treinta y cinco o más estudiantes, ordenados de mayor a menor número de alumnos. De este modo, las secciones con mayor aforo acceden en primer lugar a los espacios con capacidad más amplia.
- Fase 3 (clases estándar): Finalmente, el resto de las clases con matrícula regular se distribuyen entre las aulas que aún permanecen libres tras las dos fases anteriores.
- Fase de reintento con relajación de capacidad: Aquellas clases que no lograron asignación bajo el criterio de capacidad estricta son evaluadas nuevamente, pero esta vez admitiendo un sobrecupo controlado, cuyo límite máximo se fija en el 125 % de la capacidad nominal del aula.
- Funcion de puntuación: La indoneidad de cada aula cnadidata se determina mediante un puntaje que se minimiza:

$$puntuacion = diferenciaCapacidad - bonusPrioridad - bonusSugerencia.$$

- *diferenciaCapacidad* es el valor absoluto de la diferencia entre el aforo del aula y el número de estudiantes
- *bonusSugerencia* suma 2000 puntos si el aula coincide con la propuesta en la planificación.

  Al elegir la puntuación más baja, el sistema privilegia el aula sugerida explícitamente; en su ausencia, la que tiene prioridad para la carrera; y, si ninguna de esas condiciones se da, aquella cuya capacidad se ajusta mejor al número de alumnos.

3.5. Reglas de exclusividad y exclusiones fijas:

Los espacios designados como exclusivos para una carrera actúan como veto absoluto, por lo que se descartan sin llegar a calcular su puntuación para cualquier clase ajena a esa carrera. Por otra parte, el auditorio y la sala magna están completamente vetados de la distribución automática, al estar reservados para eventos de carácter institucional.

3.6. IA generativa: apoyo opcional, invocada desde n8n.

La IA generativa no interviene en el proceso de asignación. Su uso se limita a complementar el analizador sintáctico durante la ingesta de datos y a generar recomendaciones textuales dirigidas al administrador sobre aquellos conflictos que el motor no pudo resolver de forma automática.

3.7. Integración del canal conversacional.

Para ofrecer un acceso inmediato al sistema sin que los usuarios tengan que instalar aplicaciones extra, se optó por integrar el asistente conversacional Roomie directamente en WhatsApp. Esta funcionalidad se materializa en un microservicio autónomo denominado whatsapp-bot-aulas, desarrollado en Node.js, que se comunica de forma directa con PostgreSQL y está orquestado junto con el núcleo de la aplicación. La conexión con la plataforma de mensajería se canaliza a través de la pasarela Evolution API (v2.3.7), un componente empaquetado en Docker que gestiona la interacción con WhatsApp Web mediante webhooks HTTP que notifican al microservicio. Este canal posibilita las siguientes prestaciones:

- La identificación del usuario se realiza de manera automática, valiéndose de su número de teléfono o de su cédula de identidad.
- Es posible verificar en tiempo real el espacio asignado a un profesor en concreto o a una sesión de clase determinada.
- Las peticiones de reserva de aulas se gestionan con una verificación instantánea de la ocupación vigente.
- Se facilita la notificación de incidencias técnicas detectadas en los espacios físicos.

3.8. Infraestructura, despliegue y mantenimiento.

El despliegue de la producción se realiza usando Docker Compose (Docker Inc., sin fecha) a partir de los archivos docker-compose.yml y docker-compose.prod.yml. Por lo tanto la base de datos, Redis, el servidor de la aplicación, la interfaz de usuario y el motor de automatización se ejecutan en contenedores separados usando redes virtuales

internas.Este enfoque asegura que la reproducción del ambiente sea posible y permite agregar nuevos servicios sin afectar otros servicios existentes.

La gestión de procesos de Node.js se realiza con la herramienta PM2 con la ayuda del archivo ecosystem.config.js el cual permite el reinicio automático de esos procesos en caso de cualquier excepción inesperada o un uso anormal de los recursos. Perímetro Nginx actúa como un proxy inverso que servirá los archivos precompilados por Vite, redirigirá cualquier solicitud para la ruta /api al servidor de la aplicación y obligará a las conexiones HTTPS a través de SSL/TLS en los puertos 80 y 443.

En su totalidad esta clara segregación de responsabilidades entre el servidor de aplicación,la interfaz de usuario,los flujos de automatización y el portal de mensajería hace que sea fácil que los diferentes componentes se actualicen individualmente.El uso de técnicas de "annealamiento simulado" y de aprendizaje automático en el proceso de asignación de los recursos resulta en una reducción del tiempo de planificación de varias semanas a menos de sesenta segundos para cada asignación. También trae la ventaja adicional de aumentar el alcance de las operaciones al utilizar el poder de la plataforma web moderna junto con el del asistente de WhatsApp.

3.9. Plan de pruebas y protocolos de verificación final.

Para garantizar que Rommie satisface los requisitos funcionales y no funcionales definidos en el Capítulo II y que se comporta de forma correcta, segura y eficiente en condiciones reales de operación, se diseñó un plan de pruebas orientado a su despliegue e implementación en la Universidad Internacional del Ecuador, sede Loja. El plan articula pruebas unitarias, de integración, de extremo a extremo, no funcionales y de aceptación

de usuario, y culmina en un protocolo de verificación final que determina la aptitud del sistema para entrar en producción.

3.10. Objetivos y alcance de las pruebas.

El objetivo principal es asegurarse de que cada parte del sistema (servidor de aplicación,interfaz de usuario,motor de asignación inteligente,flujos de automatización n8n y canal de conversación de WhatsApp) funcione bien y de manera efectiva por sí sola y cuando se combinen. En donde los objetivos específicos que se plantearon son:

- Verificar el cumplimiento de los requisitos funcionales y no funcionales especificados en el Capítulo II.
- Comprobar la robustez del sistema ante las cargas concurrentes propias de un periodo de matrícula.
- Evaluar la exactitud y el tiempo de ejecución del motor de asignación de aulas.
- Confirmar la aceptación de la solución por parte de los usuarios finales de la institución.

El alcance abarca todos los módulos desarrollados y se realiza en el entorno de prueba que existe en el entorno UIDE utilizando datos representativos de la planificación académica (grados,asignaturas,profesores,capacidad y aulas del campus). Fuera de alcance están cualquier otra integración con el sistema de una institución de terceros que no exista en el prototipo y las pruebas en cualquier dispositivo o navegador que no esté listado en los requisitos. Cada caso de prueba está asociado con uno o más requisitos del Capítulo II usando la matriz de trazabilidad requisito; casos de pruebas; resultados. Esta matriz está anexa y proporciona la cobertura de todos los requisitos priorizados.

3.11. Entorno y estrategia.

Para poder reproducir con exactitud las condiciones de producción el sistema se instala en un entorno de prueba en el UIDE,utilizando la orquestación de Docker Compose, la cual es la misma que en producción.En este caso la base de datos PostgreSQL,el caché Redis,el servidor Node.js, la interfaz React entregada a través de Nginx, el motor n8n y el gateway de la API Evolution funcionarán con versiones e configuraciones idénticas.

Los datos utilizados para la fase de pruebas son los de la planificación académica de los cursos en curso en Loja (cursos, profesores,número de plazas) en cantidades suficientes para el funcionamiento realista del algoritmo.Fueron anonimizados donde fue necesario y se importaron mediante scripts.

La estrategia de pruebas avanza pro niveles progresivos:

1. Pruebas unitarias:Comprobar la lógica de los controladores,los servicios y las funciones del algoritmo de asignación por separado.
2. Pruebas de integración:Verificar la interacción del servidor de la aplicación,la base de datos,los flujos de n8n y el microservicio de WhatsApp.
3. Pruebas de extremo a extremo (E2E):Ejecutar todo el flujo de trabajo del usuario a través de la interfaz.
4. Pruebas no funcionales: Probar el rendimiento,la carga,la escalabilidad,la seguridad y la resiliencia.
5. Pruebas de aceptación del usuario (UAT):Prueba de la solución usando los perfiles reales de la organización.

Tabla 6

Herramientas de prueba.

Nivel de prueba	Herramienta	Propósito
Unitarias (backend)	Jest	Validar controladores, servicios y funciones del motor de asignación
API / integración	Supertest y Postman / Newman	Verificar los endpoints REST y su contrato de datos
Unitarias y de componentes (frontend)	Vitest y React Testing Library	Comprobar componentes React y ganchos personalizados
Extremo a extremo (E2E)	Cypress	Recorrer flujos completos de usuario en la interfaz
Carga y rendimiento	k6	Simular concurrencia y medir tiempos de respuesta y tasa de error
Seguridad	OWASP ZAP + revisión manual	Detectar vulnerabilidades comunes (OWASP Top 10) y validar controles
Aceptación de usuario	Guiones de prueba y formularios de registro	Validar la solución con los usuarios finales de la UIDE

Nota. La Tabla 3 resume las herramientas empleadas en cada nivel.

3.12. Pruebas funcionales.

Las siguientes pruebas verifican requisitos en autenticación, roles, importación de programación, asignación de aulas, consulta de espacios y asistente conversacional; cada caso incluye identificador, precondiciones, pasos, datos y resultado esperado.

Tabla 7

Casos de prueba

ID	Caso de prueba	Precondiciones / Entrada	Resultado esperado
CF-01	Inicio de sesión con credenciales válidas	Usuario y contraseña correctos registrados en el sistema	Emisión de token JWT y acceso al panel según el rol del usuario
CF-02	Inicio de sesión con credenciales inválidas	Contraseña incorrecta	Rechazo del acceso, mensaje de error y no emisión de token
CF-03	Control de acceso por rol	Usuario con rol Docente intenta acceder a un módulo de administración	Denegación del acceso y redirección a una vista permitida
CF-04	Carga de planificación académica	Archivo Excel con estructura válida (materias, docentes, cupos)	Importación correcta de los registros y validación de la estructura

CF-05	Carga de archivo con formato inválido	Archivo Excel con columnas faltantes o datos inconsistentes	Rechazo del archivo con detalle de los errores detectados
CF-06	Ejecución del motor de asignación	Solicitud de asignación para una carrera con datos cargados	Distribución de aulas sin choques de horario ni sobrecupos
CF-07	Reserva de un espacio disponible	Aula libre en la franja horaria solicitada	Confirmación de la reserva y actualización inmediata de la disponibilidad
CF-08	Reserva de un espacio ocupado	Aula ya reservada en la franja solicitada	Rechazo de la reserva con aviso claro de conflicto
CF-09	Consulta de aula por el bot de WhatsApp	Número telefónico registrado y comando de consulta válido	Respuesta con el aula asignada al docente o clase
CF-10	Generación de reporte en PDF	Solicitud de horario por parte del administrador	Descarga de un PDF con la planificación correcta y completa

Nora. La Tabla 4 presenta una selección representativa de los casos ejecutados.

3.13. Pruebas no funcionales.

Las pruebas no funcionales evalúan la calidad del sistema en cuanto al rendimiento, la carga, la seguridad y la estabilidad características que son muy importantes en el momento de los registros cuando el número de usuarios simultáneos es máximo.

Se realizaron pruebas de carga y rendimiento con la herramienta k6, emulando un acceso concurrente con un aumento progresivo de usuarios y midiendo el tiempo de respuesta (promedio,p95,p99),la tasa de errores y el consumo de recursos. Uno de los objetivos fue asegurar que todo el proceso del motor de asignación para la finalización de la carrera sea inferior a 60 segundos. Se realizaron pruebas de seguridad con la ayuda de OWASP ZAP y verificando los mecanismos de protección aplicados (JWT,bcryptjs,Helmet,limitación de tasa,sanitización de entrada).

Este tipo de pruebas adquiere especial relevancia en un sistema como Rommie, donde existen periodos de máxima demanda (como los procesos de matrícula o registro al inicio de cada curso) que concentran un volumen elevado de usuarios accediendo de forma simultánea. La evaluación del comportamiento del sistema bajo estas condiciones permite anticipar posibles cuellos de botella, prevenir caídas del servicio y asegurar que la infraestructura sea capaz de sostener la carga real de la institución, sin comprometer la experiencia del usuario ni la integridad de la información gestionada.

Las pruebas confirmaron tiempos de respuesta adecuados y estabilidad ante alta concurrencia, además de mecanismos de protección efectivos frente a vulnerabilidades comunes, respaldando la confiabilidad del sistema para su operación productiva en la UIDE.

Tabla 8

Casos de prueba no funcionales.

ID	Atributo	Condición de prueba	Resultado esperado
CNF-01	Rendimiento	Consulta de disponibilidad con carga nominal	Tiempo de respuesta promedio < 2 s y p95 < 3 s
CNF-02	Rendimiento del motor	Asignación completa de aulas de una carrera	Ejecución finalizada en menos de 60 segundos
CNF-03	Escalabilidad	Incremento progresivo de peticiones	Degradación controlada y estable de los tiempos de respuesta
CNF-04	Seguridad: autenticación	Acceso a un recurso protegido sin token JWT	Respuesta 401 Unauthorized y bloqueo del acceso
CNF-05	Seguridad: fuerza bruta	Intentos repetidos de inicio de sesión	Bloqueo temporal por limitación de peticiones (rate limiting)
CNF-06	Seguridad: inyección	Envío de cadenas maliciosas SQL/XSS	Neutralización de la entrada mediante saneamiento de datos
CNF-07	Disponibilidad / resiliencia	Caída forzada de un proceso Node.js	Reinicio automático del proceso mediante PM2
CNF-08	Cifrado del tráfico	Petición al sistema por canal no seguro	Redirección forzada a HTTPS mediante Nginx

Nota. La Tabla 8 resume los casos de prueba no funcionales.

3.14. Pruebas de aceptación de usuario.

Se realizaron pruebas de aceptación por parte de los usuarios en el entorno implementado en la UIDE donde participaron las personas que utilizan el sistema a diario.Estas incluyen al personal administrativo y de gestión para la planificación, profesores, estudiantes y el departamento de tecnología de la información y de la biblioteca.

A cada perfil se le asignaron guiones de prueba correspondientes a sus tareas habituales. Las observaciones se registraron en formularios de aceptación que sirvieron de base para los ajustes finales.

Es fundamental hacer estas pruebas con la comunidad académica para comprobar que el sistema no sólo cumple los requisitos técnicos establecidos sino que también responde eficazmente a las necesidades operativas y a las expectativas de quienes van a utilizarlo en su trabajo diario. El hecho de involucrar a distintos tipos de usuarios permite descubrir posibles inconsistencias en la usabilidad, el flujo de trabajo o la interpretación de funcionalidades que difícilmente se detectarían haciendo pruebas técnicas internas, lo cual garantiza así una validación más completa y cercana al contexto real de uso.

Los resultados de esta fase de pruebas de aceptación permitieron identificar oportunidades puntuales de mejora, antes de la puesta en producción definitiva del sistema, reforzando la confiabilidad de la solución desarrollada y asegurando que su implementación responda de manera adecuada a las dinámicas de trabajo de cada uno de los perfiles involucrados dentro de la institución.

Tabla 9

Casos de prueba.

ID	Perfil	Tarea evaluada	Criterio de aceptación
UAT-01	Administración / Dirección	Cargar la planificación y ejecutar la asignación de aulas	La asignación se completa sin conflictos y dentro del tiempo previsto
UAT-02	Administración / Dirección	Resolver un conflicto con apoyo de las sugerencias de la IA	Las sugerencias son pertinentes y facilitan la reasignación
UAT-03	Docente	Consultar el aula asignada desde la web y desde WhatsApp	La información mostrada es correcta y coincide en ambos canales
UAT-04	Docente	Reportar una incidencia técnica de un aula	La incidencia se registra y es visible para el personal de soporte
UAT-05	Estudiante	Consultar el horario y las aulas de sus materias	El estudiante accede a su información de forma clara e intuitiva
UAT-06	TI / Biblioteca	Supervisar el estado de los servicios y atender incidencias	El personal gestiona la operación sin dificultades relevantes

Nota. En la tabla 9 se ven los casos de prueba de aceptación por perfil de usuario.

3.15. Criterios de aceptación y protocolo de verificación final.

El protocolo de verificación incluye un análisis de todas las pruebas que se han realizado al sistema. El proceso de verificación se declara completado si se cumplen simultáneamente los siguientes requisitos:

- Se ha verificado el cumplimiento exitoso de todos los casos de prueba funcionales considerados críticos.
- Los tiempos de respuesta del sistema, así como la ejecución del motor de asignación, se mantienen dentro de los límites establecidos, considerando también los percentiles p95.
- Las auditorías de seguridad no han detectado vulnerabilidades clasificadas con severidad alta.
- Los distintos perfiles de usuario de la UIDE han expresado su conformidad con la solución a través de los formularios de aceptación.
- Al momento de proceder con el despliegue, no permanece abierto ningún defecto de severidad crítica o alta.

El procedimiento también incluye un manejo estructurado de las incidencias detectadas. Cada hallazgo se documenta, se categoriza según su nivel de gravedad (Crítico, Alto, Medio o Bajo) y se subsana antes de dar paso a un nuevo ciclo de verificación. Únicamente cuando se satisfacen la totalidad de los criterios estipulados se otorga la aprobación definitiva, momento en el cual el sistema queda autorizado para su puesta en marcha en la Universidad Internacional del Ecuador, sede Loja.

CONCLUSIONES.

- El motor de asignación heurístico, que funciona en un marco de tres fases junto con una función de puntuación de múltiples criterios y una capacidad de sobre reserva del 125%,pudo asignar con una tasa de éxito del 94% (47 de 50 clases),dejando solo 3 casos que requieren de una intervención manual adicional.Esto demuestra que con la capacidad actual del campus de Loja los métodos deterministas y auditables son suficientes para automatizar la programación de clases sin el uso de modelos estocásticos y de aprendizaje automático.
- Sin embargo el analizador de ingestión híbrida logró resolver el problema relacionado con la variabilidad estructural de los archivos recibidos de distintos colegios debido a que se combinó una capa determinista con un modelo de lenguaje basado en un respaldo.Esto ayudo a evitar la necesidad de obligar a las organizaciones académicas a utilizar un formato específico,lo cual era uno de los principales obstáculos para la adopción.
- Un SOA (Arquitectura Orientada a Servicios) que se encuentra contenedorizado usando Docker ya ha demostrado su valor porque mantiene la paridad entre todos los entornos al permitir la evolución individual de cada uno de los componentes.Al restringir la lógica de asignación importante al backend y confiando a n8n con efectos secundarios como las notificaciones, los informes y un posible enriquecimiento con Inteligencia Artificial el proceso es independiente de un punto externo.

RECOMENDACIONES.

- Mejora del analizador mediante retroalimentación continua. Registrar cada vez que el analizador híbrido recurre al modelo de lenguaje y mejorar el diccionario determinista gracias a los datos generados de esa actividad,lo cual reducirá la dependencia de los servicios de Inteligencia Artificial de terceros.
- Automatización integral del plan de pruebas. Implementar una cadena de integración continua que ejecute automáticamente los cuatro niveles de verificación ante cada cambio en el código, e incluir pruebas de regresión sobre casos de asignación históricos para detectar desviaciones del motor antes del despliegue.
- Ampliación del canal conversacional y medición de su impacto. Extender el bot de WhatsApp a otras funcionalidades institucionales (reserva de laboratorios, reporte de incidencias, consulta de disponibilidad docente) y realizar mediciones periódicas de satisfacción, de modo que la inversión en la pasarela de mensajería genere valor más allá de la asignación de aulas.

BIBLIOGRAFIA.

Abdelhalim, E. A., & El Khayat, G. A. (2016a). A Utilization-based Genetic Algorithm for Solving the University Timetabling Problem (UGA). *Alexandria Engineering Journal*, *55*(2), 1395–1409. https://doi.org/10.1016/J.AEJ.2016.02.017

Abdelhalim, E. A., & El Khayat, G. A. (2016b). An information visibility-based university timetabling for efficient use of learning spaces (IVUT). *Egyptian Informatics Journal*, *17*(3), 315–325. https://doi.org/10.1016/J.EIJ.2016.04.003

Abdipoor, S., Yaakob, R., Goh, S. L., & Abdullah, S. (2023a). Meta-heuristic approaches for the University Course Timetabling Problem. *Intelligent Systems with Applications*, *19*, 200253. https://doi.org/10.1016/j.iswa.2023.200253

Abdipoor, S., Yaakob, R., Goh, S. L., & Abdullah, S. (2023b). Meta-heuristic approaches for the University Course Timetabling Problem. *Intelligent Systems with Applications*, *19*, 200253. https://doi.org/10.1016/J.ISWA.2023.200253

Ahmad, I. R., Sufahani, S., Ali, M., & Razali, S. N. A. M. (2018). A Heuristics Approach for Classroom Scheduling Using Genetic Algorithm Technique. *Journal of Physics: Conference Series*, *995*(1), 012050. https://doi.org/10.1088/1742-6596/995/1/012050

Ahmed, S. K., Mohammed, R. A., Nashwan, A. J., Ibrahim, R. H., Abdalla, A. Q., M. Ameen, B. M., & Khdhir, R. M. (2025). Using thematic analysis in qualitative research. *Journal of Medicine, Surgery, and Public Health*, *6*, 100198. https://doi.org/10.1016/J.GLMEDI.2025.100198

Al-Amri, A. (2026a). *A Heatmap-Based Analytical Framework for Evaluating and Enhancing Institutional Scheduling in Higher Education*. 1–6. https://doi.org/10.1109/ICSEDI66420.2026.11568344

Al-Amri, A. (2026b). *A Heatmap-Based Analytical Framework for Evaluating and Enhancing Institutional Scheduling in Higher Education*. 1–6. https://doi.org/10.1109/ICSEDI66420.2026.11568344

Algethami, H., & Laesanklang, W. (2021). A mathematical model for course timetabling problem with faculty-course assignment constraints. *IEEE Access*, *9*, 111666–111682. https://doi.org/10.1109/ACCESS.2021.3103495

Alhijawi, B., & Awajan, A. (2024). Genetic algorithms: theory, genetic operators, solutions, and applications. *Evolutionary Intelligence*, *17*(3), 1245–1256. https://doi.org/10.1007/S12065-023-00822-6/METRICS

Almalki, S. (2016a). Integrating Quantitative and Qualitative Data in Mixed Methods Research—Challenges and Benefits. *Journal of Education and Learning*, *5*(3), p288. https://doi.org/10.5539/JEL.V5N3P288

Almalki, S. (2016b). Integrating Quantitative and Qualitative Data in Mixed Methods Research—Challenges and Benefits. *Journal of Education and Learning*, *5*(3), p288. https://doi.org/10.5539/JEL.V5N3P288

Arias-Osorio, J., & Mora-Esquivel, A. (2020). A solution to the university course timetabling problem using a hybrid method based on genetic algorithms. *DYNA*, *87*(215), 47–56. https://doi.org/10.15446/DYNA.V87N215.85933

Avella, P., & Vasil'ev, I. (2005). A Computational Study of a Cutting Plane Algorithm for University Course Timetabling. *Journal of Scheduling 2005 8:6*, *8*(6), 497–514. https://doi.org/10.1007/S10951-005-4780-1

Barcellos, M., Santos, G., Conte, T., Trinkenreich, B., & Matsubara, P. (2022a). Organizing Empirical Studies as Learning Iterations in Design Science Research

Projects. *ACM International Conference Proceeding Series*. https://doi.org/10.1145/3571473.3571474

Barcellos, M., Santos, G., Conte, T., Trinkenreich, B., & Matsubara, P. (2022b). Organizing Empirical Studies as Learning Iterations in Design Science Research Projects. *ACM International Conference Proceeding Series*. https://doi.org/10.1145/3571473.3571474

Bashab, A., Ibrahim, A. O., Hashem, I. A. T., Aggarwal, K., Mukhlif, F., Ghaleb, F. A., & Abdelmaboud, A. (2022). Optimization Techniques in University Timetabling Problem: Constraints, Methodologies, Benchmarks, and Open Issues. *Computers, Materials & Continua*, *74*(3), 6461–6484. https://doi.org/10.32604/CMC.2023.034051

Bettinelli, A., Cacchiani, V., Roberti, R., & Toth, P. (2015). An overview of curriculum-based course timetabling. *TOP 2015 23:2*, *23*(2), 313–349. https://doi.org/10.1007/S11750-015-0366-Z

Beyrouthy, C., Burke, E. K., Landa-Silva, D., McCollum, B., McMullan, P., & Parkes, A. J. (2009). Towards improving the utilization of university teaching space. *Journal of the Operational Research Society*, *60*(1), 130–143. https://doi.org/10.1057/PALGRAVE.JORS.2602523

Bonutti, A., De Cesco, F., Di Gaspero, L., & Schaerf, A. (2010). Benchmarking curriculum-based course timetabling: formulations, data formats, instances, validation, visualization, and results. *Annals of Operations Research 2010 194:1*, *194*(1), 59–70. https://doi.org/10.1007/S10479-010-0707-0

Borg, S. (2011). Doing Action Research in English Language Teaching. A Guide for Practitioners. *ELT Journal*, *65*(4), 485–487. https://doi.org/10.1093/ELT/CCR052

Chen, M. C., Sze, S. N., Goh, S. L., Sabar, N. R., & Kendall, G. (2021). A Survey of University Course Timetabling Problem: Perspectives, Trends and Opportunities. *IEEE Access*, *9*, 106515–106529. https://doi.org/10.1109/ACCESS.2021.3100613

Chen, Y. (n.d.). *A Comprehensive Investigation of Genetic Algorithms*.

Colajanni, G., & Daniele, P. (2020). A new model for curriculum-based university course timetabling. *Optimization Letters 2020 15:5*, *15*(5), 1601–1616. https://doi.org/10.1007/S11590-020-01588-X

Computational study of a cutting plane algorithm for University Course Timetabling – Optimization Online. (n.d.). Retrieved August 30, 2026, from https://optimization-online.org/2003/07/685/

Constantino, A. A., Landa-Silva, D., de Melo, E. L., de Mendonça, C. F. X., Rizzato, D. B., & Romão, W. (2013). A heuristic algorithm based on multi-assignment procedures for nurse scheduling. *Annals of Operations Research 2013 218:1*, *218*(1), 165–183. https://doi.org/10.1007/S10479-013-1357-9

Consulting, H. (n.d.). *Space utilization and Optimization Report for the University of Colorado Boulder*.

Cordeiro, L., Soares, C. B., & Rittenmeyer, L. (2017a). Unscrambling method and methodology in action research traditions: theoretical conceptualization of praxis and emancipation. *Qualitative Research*, *17*(4), 395–407. https://doi.org/10.1177/1468794116674771;SUBPAGE:STRING:ABSTRACT;WEBSITE:WEBSITE:SAGE;JOURNAL:JOURNAL:QRJA;ISSUE:ISSUE:DOI

Cordeiro, L., Soares, C. B., & Rittenmeyer, L. (2017b). Unscrambling method and methodology in action research traditions: theoretical conceptualization of praxis and emancipation. *Qualitative Research*, *17*(4), 395–407.

https://doi.org/10.1177/1468794116674771;SUBPAGE:STRING:ABSTRACT;WEBSITE:WEBSITE:SAGE;JOURNAL:JOURNAL:QRJA;ISSUE:ISSUE:DOI

Crocker, A. (n.d.). *Running Out of Classroom Space? Maybe It's a Schedule Problem*.

Cyun, D., Fang, J., & Ke, T.-W. (2025). *Information Seeking for Robust Decision Making under Partial Observability*. https://arxiv.org/pdf/2510.01531

Daskalaki, S., Birbas, T., & Housos, E. (2004a). An integer programming formulation for a case study in university timetabling. *European Journal of Operational Research*, *153*(1), 117–135. https://doi.org/10.1016/S0377-2217(03)00103-6

Daskalaki, S., Birbas, T., & Housos, E. (2004b). An integer programming formulation for a case study in university timetabling. *European Journal of Operational Research*, *153*(1), 117–135. https://doi.org/10.1016/S0377-2217(03)00103-6

Davison, R. M., Martinsons, M. G., & Kock, N. (2004). Principles of canonical action research. *Information Systems Journal*, *14*(1), 65–86. https://doi.org/10.1111/J.1365-2575.2004.00162.X;PAGEGROUP:STRING:PUBLICATION

Duncan L. Hill. (2008). *Qualitative Timetabling An Organizational and Qualitative Approach to*.

Empty classrooms, full campuses: rethinking space utilization in higher education. (n.d.). Retrieved August 29, 2026, from https://www.vantagespace.com/blog/empty-classrooms-full-campuses-rethinking-space-utilization-higher

Ezugwu, A. E., Shukla, A. K., Nath, R., Akinyelu, A. A., Agushaka, J. O., Chiroma, H., & Muhuri, P. K. (2021). Metaheuristics: a comprehensive overview and classification along with bibliometric analysis. *Artificial Intelligence Review 2021 54:6*, *54*(6), 4237–4316. https://doi.org/10.1007/S10462-020-09952-0

FERPA | Protecting Student Privacy. (n.d.). Retrieved August 29, 2026, from https://studentprivacy.ed.gov/ferpa

Fetters, M. D., Curry, L. A., & Creswell, J. W. (2013a). Achieving integration in mixed methods designs - Principles and practices. *Health Services Research*, *48*(6 PART2), 2134–2156. https://doi.org/10.1111/1475-6773.12117;REQUESTEDJOURNAL:JOURNAL:14756773;JOURNAL:JOURNAL:14756773;WGROUP:STRING:PUBLICATION

Fetters, M. D., Curry, L. A., & Creswell, J. W. (2013b). Achieving integration in mixed methods designs - Principles and practices. *Health Services Research*, *48*(6 PART2), 2134–2156. https://doi.org/10.1111/1475-6773.12117;REQUESTEDJOURNAL:JOURNAL:14756773;JOURNAL:JOURNAL:14756773;WGROUP:STRING:PUBLICATION

Gama, da, Gradual, D., Han, X., & Wang, D. (2025). Gradual Optimization of University Course Scheduling Problem Using Genetic Algorithm and Dynamic Programming. *Algorithms 2025, Vol. 18, Page 158*, *18*(3), 158. https://doi.org/10.3390/A18030158

Hambali, A. M., Olasupo, Y. A., & Dalhatu, M. (2020). Automated university lecture timetable using Heuristic Approach. *Nigerian Journal of Technology*, *39*(1), 1–14. https://doi.org/10.4314/NJT.V39I1.1

Hasmadi, A. A., Kamaruddin, N., & Salim, N. A. A. (2026). Space Utilisation Rate of Public Higher Education Classrooms in Klang Valley. *Built Environment Journal*, *23*(1), 22–31. https://doi.org/10.24191/bej.v23i1.1106

Hussain, S. A. I., Kalita, R., & Mandal, U. K. (2023). A priority-based heuristic approach for solving flexible flow-shop with parallel machine scheduling in a fuzzy

environment. *Granular Computing 2023 8:6*, *8*(6), 1097–1120. https://doi.org/10.1007/S41066-023-00377-Y

IEC 62443: Ultimate OT Security Guide | Rockwell Automation | UK. (2025, May 19). https://www.rockwellautomation.com/en-gb/company/news/blogs/iec-62443-security-guide.html

Islam, T., Shahriar, Z., Perves, M. A., & Hasan, M. (2016). University Timetable Generator Using Tabu Search. *Journal of Computer and Communications*, *04*(16), 28–37. https://doi.org/10.4236/jcc.2016.416003

Ivankova, N. V., Creswell, J. W., & Stick, S. L. (2006). Using Mixed-Methods Sequential Explanatory Design: From Theory to Practice. *Field Methods*, *18*(1), 3–20. https://doi.org/10.1177/1525822X05282260;WEBSITE:WEBSITE:SAGE;ISSUE:ISSUE:DOI

Jaidi, S. R. (2026). Agentic AI in Enterprise Middleware: Toward Autonomous Integration Orchestration. *International Journal of Computational and Experimental Science and Engineering*, *12*(3), 41–55. https://doi.org/10.22399/IJCESEN.5339

Joseph, S. B., Dada, E. G., Abidemi, A., Oyewola, D. O., & Khammas, B. M. (2022). Metaheuristic algorithms for PID controller parameters tuning: review, approaches and open problems. *Heliyon*, *8*(5). https://doi.org/10.1016/J.HELIYON.2022.E09399

Kahar, M. N. M., & Kendall, G. (2010). The examination timetabling problem at Universiti Malaysia Pahang: Comparison of a constructive heuristic with an existing software solution. *European Journal of Operational Research*, *207*(2), 557–565. https://doi.org/10.1016/J.EJOR.2010.04.011

Kaushik, D., & Nadeem, M. (2025). Batch-enabled randomized parameter tuning for improved metaheuristic performance. *Evolving Systems*, *16*(1), 11-. https://doi.org/10.1007/S12530-024-09633-1/METRICS

Kumar, A. (2026). *Timetable Optimizer: Automated University Schedule Generation Using CP-SAT Constraint Solving and Large Language Model Intelligence Adyasha Nanda*. *2*(1), 145. www.zesterapublications.com

Kumar, M., Husain, Dr. M., Upreti, N., & Gupta, D. (2010). Genetic Algorithm: Review and Application. *SSRN Electronic Journal*. https://doi.org/10.2139/SSRN.3529843

Lewis, J. R. (2018). The System Usability Scale: Past, Present, and Future. *International Journal of Human–Computer Interaction*, *34*(7), 577–590. https://doi.org/10.1080/10447318.2018.1455307

Makalesi Müfredat Tabanlı Ders Çizelgeleme Problemi için Yeni Bir Açgözlü Algoritma Batuhan COŞAR, A., Say, B., Dökeroğlu, T., Mühendisliği Bölümü, B., Fakültesi, M., Üniversitesi, A., Yazılım Mühendisliği Bölümü, A., & Üniversitesi, Ç. (2023). A New Greedy Algorithm for the Curriculum-based Course Timetabling Problem. *Duzce University Journal of Science and Technology*, *11*(2), 1121–1136. https://doi.org/10.29130/DUBITED.1113519

Martínez-Jara, J. N., & Pérez-Ycaza, J. C. (2022). Protección de datos personales en la historia clínica electrónica bajo el marco legal ecuatoriano. *IUSTITIA SOCIALIS*, *7*(1), 776–801. https://doi.org/10.35381/racji.v7i1.2203

Mulvey, J. M. (1982). A classroom/time assignment model. *European Journal of Operational Research*, *9*(1), 64–70. https://doi.org/10.1016/0377-2217(82)90012-1

Mwita, K. M., & Mwilongo, N. H. (2025). Thematic Analysis of Qualitative Research Data: A Seven-Step Guide. *Eminent Journal of Business and Management*, *1*(1), 51–59. https://doi.org/10.70582/4AJW7K14

Nowell, L. S., Norris, J. M., White, D. E., & Moules, N. J. (2017). Thematic Analysis: Striving to Meet the Trustworthiness Criteria. *International Journal of Qualitative Methods*, *16*(1). https://doi.org/10.1177/1609406917733847;WGROUP:STRING:PUBLICATION

Oude Vrielink, R. A., Jansen, E. A., Hans, E. W., van Hillegersberg, J., & van Hillegersberg jvanhillegersberg, J. (2019). Practices in timetabling in higher education institutions: a systematic review. *Ann Oper Res*, *275*, 145–160. https://doi.org/10.1007/s10479-017-2688-8

Paspuel López, J. P., & Naranjo Luzuriaga, E. P. (2017). *Análisis e implementación de un algoritmo de optimización para la asignación automática de aulas clase en el Campus Sur de la Universidad Politécnica Salesiana*. https://dspace.ups.edu.ec/handle/123456789/14531

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, *24*(3), 45–77. https://doi.org/10.2753/MIS0742-1222240302

Pillay, N., & Özcan, E. (2017a). Automated generation of constructive ordering heuristics for educational timetabling. *Annals of Operations Research 2017 275:1*, *275*(1), 181–208. https://doi.org/10.1007/S10479-017-2625-X

Pillay, N., & Özcan, E. (2017b). Automated generation of constructive ordering heuristics for educational timetabling. *Annals of Operations Research 2017 275:1*, *275*(1), 181–208. https://doi.org/10.1007/S10479-017-2625-X

Pillay, N., & Özcan, E. (2017c). Automated generation of constructive ordering heuristics for educational timetabling. *Annals of Operations Research 2017 275:1*, *275*(1), 181–208. https://doi.org/10.1007/S10479-017-2625-X

Pineda, M. J. I., & Ruiz, L. M. M. (2025). Personal data protection in the face of the development of artificial intelligence in Ecuador: analysis and legislative proposals. *Visionario Digital*, *9*(2), 24–48. https://doi.org/10.33262/visionariodigital.v9i2.3366

Pongcharoen, P., Stewardson, D. J., Hicks, C., & Braiden, P. M. (2001). Applying designed experiments to optimize the performance of genetic algorithms used for scheduling complex products in the capital goods industry. *Journal of Applied Statistics*, *28*(3–4), 441–455. https://doi.org/10.1080/02664760120034162

Qiu, T., Herrmann, H., & Afrin, M. (2026). A Hybrid Genetic Algorithm and Simulated Annealing Approach for Optimal University Course Timetabling. *Proceedings of 2026 Australasian Computer Science Week, ACSW 2026*, 64–72. https://doi.org/10.1145/3793811.3793816;SUBPAGE:STRING:BASIC

Razali, M. K. M., Rahman, A. H. A., Ayob, M., Jarmin, R., Yong, L. C., Maaya, M., Izaham, A., & Rahman, R. A. (2024a). An Improved Saturation Degree-Based Constructive Heuristic for Master Surgery Scheduling Problem: Case Study. *IEEE Access*, *12*, 44748–44772. https://doi.org/10.1109/ACCESS.2024.3380160

Razali, M. K. M., Rahman, A. H. A., Ayob, M., Jarmin, R., Yong, L. C., Maaya, M., Izaham, A., & Rahman, R. A. (2024b). An Improved Saturation Degree-Based

Constructive Heuristic for Master Surgery Scheduling Problem: Case Study. *IEEE Access*, *12*, 44748–44772. https://doi.org/10.1109/ACCESS.2024.3380160

Riis, T. (2026). General rights A Comprehensive Study of Educational Timetabling-a Survey. *Downloaded from Orbit.Dtu.Dk On*.

Robinson, P. (2007). Designing and Conducting Mixed Methods Research | By John W.Creswell and Vicki L.Piano Clark. Published by Sage Publications , California , 2007. Paperback, 273 pages. $66. ISBN 1 4129 2792 7. *Australian and New Zealand Journal of Public Health*, *31*(4), 388. https://doi.org/10.1111/J.1753-6405.2007.00096.X

Roldán Carrillo, F. N. (2021). Los ejes centrales de la protección de datos: consentimiento y finalidad. Críticas y propuestas hacia una regulación de la protección de datos personales en Ecuador. *USFQ Law Review*, *8*(1), 175–202. https://doi.org/10.18272/ulr.v8i1.2184

Rudová, H., Müller, T., & Murray, K. (2010). Complex university course timetabling. *Journal of Scheduling 2010 14:2*, *14*(2), 187–207. https://doi.org/10.1007/S10951-010-0171-3

Ruiz-Torrubiano, R. (2024). *Modeling Local Search Metaheuristics Using Markov Decision Processes*.

Runeson, P., Engström, E., & Storey, M.-A. (2020). The Design Science Paradigm as a Frame for Empirical Software Engineering. *Contemporary Empirical Methods in Software Engineering*, 127–147. https://doi.org/10.1007/978-3-030-32489-6_5

Russell, Stuart and Norvig, & Peter. (2020). *Artificial intelligence a modern approach*.

Ryder, J. (n.d.). *A Genetic Algorithm for Scheduling*.

Sangaiah, K., Cai, X., & Sufi, F. (2023). Algorithms in Low-Code-No-Code for Research Applications: A Practical Review. *Algorithms 2023, Vol. 16, Page 108*, *16*(2), 108. https://doi.org/10.3390/A16020108

Schimmelpfeng, K., & Helber, S. (2006). Application of a real-world university-course timetabling model solved by integer programming. *OR Spectrum 2006 29:4*, *29*(4), 783–803. https://doi.org/10.1007/S00291-006-0074-Z

Schlegel, M., Jacobsen, A., Abbas, Z., Patterson, A., White, A., & White, M. (2021). General Value Function Networks. *Journal of Artificial Intelligence Research*, *70*, 497–543. https://doi.org/10.1613/JAIR.1.12105

SPACE UTILIZATION & SPACE NEEDS REPORT. (n.d.).

Stickland, S., Scott, N., & Athauda, R. (2024). *A Hybrid Design-Based Methodology for Cross-Discipline Research Projects Developing Software Artifacts*. 68–72. https://doi.org/10.18848/978-1-963049-18-3/CGP/68-72

Theophily Nsulangi, P., Werneld, ;, Ngongi, E., Mansour, ;, Likamba, R., Omary, ;, Sarehe, B., Miraji, ;, & Mkwande, A. (2024). International Journal of Computer Science and Mobile Computing A Comparative Analysis of Manual and Automatic Timetabling Approaches for Resource Utilisation in Tertiary Higher Learning Institution. *International Journal of Computer Science and Mobile Computing*, *13*, 65–76. https://doi.org/10.47760/ijcsmc.2024.v13i12.007

Torres-Ovalle, C., Montoya-Torres, J. R., Quintero-Araújo, C. L., Sarmiento-Lepesqueur, A., & Castilla-Luna, M. (2014). Programación de horarios y asignación de aulas de clases universitarias. *Ingenieria y Universidad*, *18*(1), 59–76. https://doi.org/10.11144/JAVERIANA.IYU18-1.PHAA

Vega, E., Soto, R., Crawford, B., Peña, J., Contreras, P., & Castro, C. (2022). Predicting population size and termination criteria in metaheuristics: A case study based on spotted hyena optimizer and crow search algorithm. *Applied Soft Computing*, *128*, 109513. https://doi.org/10.1016/J.ASOC.2022.109513

Viljoen, A., Hein, A., & Krcmar, H. (2026). Low-code development platform ecosystems. *Electronic Markets 2026 36:1*, *36*(1), 12-. https://doi.org/10.1007/S12525-025-00848-X

Vincent Recreo, J. B., Rica Villaceran, J. S., Stephen Tedoso, M. D., Gabriel Tayong, D. V, & Ilustrisimo, D. (n.d.). A Feasibility Study on the Effectiveness of Manual Scheduling and Room Allocation. *International Journal of Scientific Research and Engineering Development*. Retrieved August 29, 2026, from www.ijsred.com

Waszkowski, R. (2019). Low-code platform for automating business processes in manufacturing. *IFAC-PapersOnLine*, *52*(10), 376–381. https://doi.org/10.1016/J.IFACOL.2019.10.060

Why Excel for Class Scheduling & Information Silos are Bad for Student Success. (n.d.). Retrieved August 29, 2026, from https://www.coursedog.com/articles/why-information-silos-hurt-scheduling

Wieringa, R. J. (2014a). Design science methodology: For information systems and software engineering. *Design Science Methodology: For Information Systems and Software Engineering*, 1–332. https://doi.org/10.1007/978-3-662-43839-8/SAVE-RESEARCH

Wieringa, R. J. (2014b). Design science methodology: For information systems and software engineering. *Design Science Methodology: For Information Systems and*

Software Engineering, 1–332. https://doi.org/10.1007/978-3-662-43839-8/SAVE-RESEARCH

Wijaya, M. C. (2020). Two stages best first search algorithm using hard and soft constraints heuristic for course timetabling. *Revue d'Intelligence Artificielle*, *34*(4), 413–418. https://doi.org/10.18280/RIA.340405

William Villa Quishpe, M., José Augusto Cadena Moreano, E., Angel Rodrigo Gavilanez Guanoluisa, E., & Steve Centeno Atavallo, C. (2023). Protección de Datos Personales en Ecuador y Colombia: Principios, Ética y Desafíos Actuales. *Revista Científica de Informática ENCRIPTAR - ISSN: 2737-6389.*, *6*(11), 19–34. https://doi.org/10.56124/ENCRIPTAR.V6I11.0002

Wohlin, C., & Aurum, A. (2014). Towards a decision-making structure for selecting a research design in empirical software engineering. *Empirical Software Engineering 2014 20:6*, *20*(6), 1427–1455. https://doi.org/10.1007/S10664-014-9319-7

Yaw Dwamena Quansah, J., Osei, S., & Musah Abudu, A. (n.d.). *Optimization of teaching space potential: A Ghanaian tertiary institution educational leadership perspective*. https://doi.org/10.30918/AERJ.124.24.056