

Semestría – Plataforma de planificación de materias



Semestría – Plataforma de planificación de materias

Juan Felipe Rojas Hernández

Santiago Barros Hernández

Facultad De Ingeniería, Universidad EAN

Proyecto De Grado

Profesor John Jairo Porras Vega

Bogotá, 31 de mayo de 2025

Contenido

Resumen Ejecutivo	6
Abstract.....	7
Introducción	8
Objetivos.....	9
Objetivos Generales	9
Objetivos específicos	9
Definición del Problema	10
Justificación	12
Marco Teórico	14
Conceptos Claves y Definiciones	14
Antecedentes y Estado del Arte	15
Teorías y Modelos Relevantes.....	16
Síntesis del enfoque teórico para el proyecto	17
Restricciones	18
Restricciones ambientales	19
Restricciones económicas	19
Restricciones legales.....	19
Restricciones Socioculturales.....	20
Restricciones internas	20
Análisis de Requerimientos	21
Metodología Para el Desarrollo de la Solución	29
Desarrollo de la Solución.....	30
Análisis de costos	32
Arquitectura del Software.....	34

Swagger.....	41
Conclusiones.....	42
Referencias.....	43

Índice de tablas

Tabla 1 <i>Historias De Usuario - HU001</i>	21
Tabla 2 <i>Historias De Usuario - HU002</i>	22
Tabla 3 <i>Historias De Usuario - HU003</i>	23
Tabla 4 <i>Historias De Usuario - HU004</i>	24
Tabla 5 <i>Historias De Usuario - HU005</i>	25
Tabla 6 <i>Historias De Usuario - HU006</i>	25
Tabla 7 <i>Historias De Usuario - HU007</i>	26
Tabla 8 <i>Historias De Usuario - HU008</i>	27
Tabla 9 <i>Historias De Usuario - HU009</i>	28
Tabla 10 <i>Tabla de costos</i>	33
Tabla 11 <i>Servicios públicos</i>	33
Tabla 12 <i>Modelos y Métodos de Semestria</i>	39

Índice de figuras

Figura 1 <i>Diagrama de casos de uso realizado en Draw.io</i>	35
Figura 2 <i>Diagrama de contexto realizado en Draw.io</i>	36

Figura 3 Diagrama de contexto del backend 37

Figura 4 Diagrama de contexto microservicios 38

Resumen Ejecutivo

El presente proyecto propone el diseño y desarrollo de una plataforma web enfocada en mejorar la experiencia de pre-inscripción de materias para estudiantes universitarios. La propuesta busca optimizar el tiempo y la facilidad de uso, permitiendo a los estudiantes visualizar, filtrar y seleccionar asignaturas según su carrera, horarios y docentes de manera intuitiva. El desarrollo se centra en una interfaz de usuario atractiva e interactiva utilizando React para el frontend, complementada con un backend en FastAPI que facilita la carga de datos por parte del personal administrativo. El sistema contempla dos perfiles: estudiante, con acceso a la consulta y selección de materias; y administrativo, encargado de la carga y actualización de la información académica. Metodológicamente, se seguirá un enfoque AGILE-SCRUM y se usará el ciclo de vida del desarrollo de software (SDLC) que permita iterar el diseño y funcionalidad basados en retroalimentación continua. Se espera como resultado una herramienta que reduzca drásticamente el tiempo de inscripción, mejore la usabilidad del proceso y ofrezca una gestión académica más eficiente.

Palabras Claves: Inscripción de materias, plataforma web, React, FastAPI, visualización de datos, usabilidad.

Abstract

This project proposes the design and development of a web platform focused on improving the course registration experience for university students. The proposal seeks to optimize time and ease of use, allowing students to intuitively view, filter, and select courses based on their major, schedule, and instructors. Development focuses on an attractive and interactive user interface using React for the frontend, complemented by a backend in FastAPI, that manages information and facilitates data entry by administrative staff. The system includes two profiles: student, with access to course consultation and selection; and administrative, responsible for entering and updating academic information. Methodologically, an AGILE-SCRUM approach will be followed, and the software development life cycle (SDLC) will be used to iterate the design and functionality based on continuous feedback. The expected result is a tool that drastically reduces enrollment time, improves the usability of the process, and offers more efficient academic management.

Keywords: course registration, web platform, React, FastAPI, data visualization, usability.

Introducción

La inscripción y organización de unidades de estudios, junto a la planificación de lo que va a ser su rutina durante todo el semestre, es uno de los procesos más importantes en la vida académica de los estudiantes universitarios, pues define la planificación del semestre, la carga académica y extracurricular junto a la experiencia educativa en general. En la Universidad EAN, este proceso depende principalmente de dos herramientas: un archivo Excel que contiene la oferta académica y la plataforma institucional SAP, encargada de registrar oficialmente las inscripciones. Aunque este método ha funcionado durante años, presenta limitaciones importantes, especialmente en términos de actualización de la información, facilidad de consulta y construcción de horarios.

Diversos estudiantes han reportado dificultades causadas por datos desactualizados, errores en los horarios y la falta de filtros eficientes en el archivo Excel. Asimismo, SAP, aunque robusto en su función administrativa, no ofrece herramientas que faciliten la visualización, comparación y planificación previa de las asignaturas. Esta situación genera demoras, reprocesos y una experiencia de usuario poco óptima.

En respuesta a estas necesidades, en Semestria proponemos como solución el desarrollo de una plataforma web que integre y organice la oferta académica de manera más clara, dinámica y accesible. La propuesta se enmarca en el campo de la ingeniería de software aplicada a la gestión académica, y busca aportar una solución que mejore la planeación estudiantil, reduzca errores de planeación de horario académico y personal.

Finalmente, este documento se estructura en sección es que abordan los objetivos del proyecto, la justificación, el marco teórico, la metodología aplicada, el desarrollo de la solución propuesta y los resultados obtenidos, culminando con las conclusiones y recomendaciones para futuros trabajos

Objetivos

Objetivos Generales

Desarrollar los componentes de back-end de SEMESTRIA para producto mínimo viable de plataforma digital. Desarrollar los componentes del Front-end para el producto mínimo viable del sistema Semestria (plataforma digital) que facilita la planificación académica universitaria.

Objetivos específicos

Identificar las principales limitaciones que enfrentan los estudiantes en la organización de sus horarios.

Diseñar la arquitectura del backend empleando microservicios, asegurando escalabilidad, modularidad y capacidad de integración con sistemas académicos institucionales.

Desarrollar un sistema académico simulado que reproduzca el flujo de información entre SAP y la plataforma, facilitando pruebas controladas y escenarios de validación.

Integrar un calendario interactivo en el frontend que consuma los servicios del backend y muestre la oferta académica en tiempo real, con opciones de filtrado y preinscripción.

Definición del Problema

En la Universidad EAN, el proceso de inscripción de materias constituye uno de los momentos más importantes del ciclo académico de los estudiantes, ya que define la organización del semestre, la carga de trabajo y, en muchos casos, el éxito en el cumplimiento de los objetivos académicos. Actualmente, este proceso depende de dos herramientas principales: un archivo Excel que contiene la oferta académica y la plataforma SAP, en la que se formaliza la inscripción. Sin embargo, esta dinámica presenta múltiples limitaciones que repercuten negativamente en la experiencia del estudiante y en la eficiencia administrativa de la institución.

Por un lado, el método que se usa en la Universidad para dar a conocer la oferta académica a los estudiantes carece de actualización automática o filtros dinámicos que faciliten la búsqueda, teniendo en cuenta que se hace a través de un archivo de Excel. Esto obliga a los estudiantes a realizar consultas manuales extensas, con un alto riesgo de error y desactualización de la información o en casos peores, que el estudiante no tenga conocimientos o dominios de la herramienta Excel, lo que hace del proceso de organización de horario una completa molestia y que peor aún, genere una alta carga académica. En encuestas realizadas, el 87% de los estudiantes reportaron errores en los horarios publicados, y el 85% identificaron información desactualizada. Por otro lado, SAP, aunque es una herramienta robusta en términos administrativos, no ofrece una interfaz intuitiva ni funcionalidades que permitan a los estudiantes planificar sus horarios de manera flexible y eficiente.

En el caso particular de la EAN, la ausencia de un sistema unificado que integre y muestre la información académica de forma clara y dinámica constituye el núcleo del problema. Esta carencia incrementa la incertidumbre en los estudiantes, afecta la planeación de su semestre y aumenta la carga del área de Unidad Académica Administra, junto a un colapso en la mesa de

ayuda y el área de Meliturno, que deben atender un volumen alto de consultas, quejas, reclamos y correcciones por parte de los estudiantes.

El problema se aborda en el marco temporal del período académico 2026-1, tomando como unidad de análisis los estudiantes de pregrado de la sede Bogotá. Se busca responder hasta qué punto una plataforma web con backend especializado, base de datos relacional y un sistema académico simulado puede contribuir a reducir errores, optimizar tiempos de consulta y mejorar la experiencia de usuario en el proceso de inscripción de asignaturas. Finalmente identificamos 3 problemáticas a la hora de organizar nuestro horario para el siguiente semestre académico:

- La información entre SAP y el Excel entregado por la universidad no se encuentra actualizada.
- Los estudiantes requieren de hacer consultas manuales, esto puede generar el planeamiento erróneo de las unidades de estudio que el estudiante planea ver el siguiente semestre
- La inexistente herramienta que permita realizar una preinscripción de unidades de estudio

Justificación

El proceso de inscripción de materias en la educación superior no solo constituye un trámite administrativo, sino un factor decisivo en la planificación académica de los estudiantes y en la eficiencia de los procesos institucionales. En la Universidad EAN, la dependencia de archivos Excel y de la plataforma SAP ha evidenciado limitaciones que afectan la experiencia de los usuarios y aumentan la carga operativa. Estas dificultades hacen pertinente la búsqueda de una solución tecnológica innovadora que, mediante un backend robusto, una base de datos relacional y un sistema académico simulado, permita optimizar la gestión de información académica y apoyar la toma de decisiones de los estudiantes en un entorno más confiable y eficiente.

La conveniencia del proyecto se fundamenta en la tendencia global hacia la transformación digital en las instituciones de educación superior. Universidades de distintos contextos han demostrado que la implementación de sistemas académicos interactivos y de apoyo previo a la inscripción mejora la satisfacción estudiantil, reduce errores administrativos y aporta a la eficiencia organizacional. Desde una perspectiva económica y de mercado, el fortalecimiento de los procesos digitales constituye una ventaja competitiva para la universidad, al consolidar una experiencia académica diferenciada que puede mejorar la retención estudiantil y la percepción de calidad institucional.

En el ámbito empresarial y organizacional, este proyecto contribuye a la evolución de los procesos internos, reduciendo tiempos de atención, consultas y reprocesos asociados a errores de inscripción. De esta manera, no solo beneficia a los estudiantes, sino que también disminuye la carga operativa de los departamentos académicos y administrativos, liberando recursos humanos para tareas de mayor valor estratégico. El valor teórico de la investigación radica en su aporte a

la ingeniería de software aplicada a la gestión académica, al explorar la integración de un backend escalable, una base de datos relacional eficiente y un entorno simulado que permita validar funcionalidades antes de una implementación completa. Este enfoque no solo permite comprobar la viabilidad técnica de la solución, sino también generar conocimiento aplicable en el campo de los sistemas de información y la transformación digital universitaria.

La relevancia social se refleja en la mejora de la experiencia de los estudiantes, quienes contarán con una herramienta intuitiva para planificar su carga académica de manera más clara, rápida y eficiente. Esto impacta positivamente en la organización del tiempo, la disminución del estrés asociado a la inscripción y la reducción de errores que puedan afectar su trayectoria académica.

En términos de alcance, el proyecto es viable gracias a la disponibilidad de recursos técnicos y humanos, así como al acceso a información institucional necesaria para desarrollar y probar el sistema en un contexto simulado. Su implementación se realizará en el marco temporal del período académico 2026-1, centrado en los estudiantes de pregrado de la sede Bogotá.

Finalmente, este trabajo se enmarca en el campo de la Ingeniería, dentro del grupo de investigación en Ingeniería de Software y Sistemas de Información, y en la línea de investigación en transformación digital y sistemas de información aplicados a la educación. De esta forma, responde tanto a las necesidades institucionales actuales como a la filosofía investigativa de la Universidad EAN, orientada a la innovación, la sostenibilidad y la mejora continua de la experiencia académica.

Marco Teórico

Conceptos Claves y Definiciones

A continuación, se dará la definición de conceptos comúnmente usados en el transcurso de la redacción de algunos artículos de este marco teórico con el fin de evitar ambigüedades.

- **Backend:** capa lógica del sistema que gestiona las reglas de negocio, exposición de APIs, autenticación, autorización, validaciones de horario y coordinación entre servicios.
- **Base de datos relacional:** sistema de almacenamiento estructurado que permite consultas eficientes, integridad referencial y soporte transaccional (propiedades ACID).
- **Simulador académico:** módulo que genera datos ficticios sobre asignaturas, horarios y cupos para probar el sistema sin depender del entorno real institucional
- **Preinscripción:** proceso en que los estudiantes seleccionan materias de forma anticipada, antes del registro oficial, con validaciones automáticas de conflictos de horario.
- **Calendario interactivo:** representaciones visuales de horarios (vista semanal o diaria) que permiten al usuario ver cómo se distribuyen sus materias y detectar choques de horario.
- **Usabilidad:** grado en que un sistema puede ser usado por usuarios específicos para lograr objetivos con eficacia, eficiencia y satisfacción en un contexto determinado.

- Aceptación tecnológica: grado en que los usuarios adoptan un sistema, influenciada por la percepción de facilidad de uso y utilidad.

Antecedentes y Estado del Arte

La inscripción de asignaturas —o el registro académico— es un proceso común en universidades que tradicionalmente ha dependido de sistemas manuales o sistemas institucionales rígidos que carecen de flexibilidad y de buena experiencia de usuario. Investigaciones recientes muestran que estos sistemas tradicionales suelen generar errores, conflictos de horario y baja satisfacción de los usuarios (Ramírez & Ochoa, 2022).

Por ejemplo, el estudio Automated Class Scheduling System evaluó cómo un sistema automatizado frente a un método manual mostraba mejores resultados en precisión, seguridad y manejo de datos en el contexto universitario filipino. Concluyó que el sistema automatizado resolvía problemas fundamentales del proceso manual, como errores y uso intensivo de tiempo (Gamale, Anuta & Sayson, 2012).

En el ámbito de plataformas educativas, el estudio “Enhancing user experience in e-learning systems: A new approach” explora cómo el diseño centrado en el usuario puede elevar la usabilidad de sistemas educativos, enfatizando que muchas veces los desarrollos tecnológicos priorizan funciones más que la experiencia real de los estudiantes (Mostefai et al., 2025).

Asimismo, la literatura sobre usabilidad en tecnología educativa evidencia que muchos desarrollos descuidan la evaluación formal de usabilidad, recurriendo a mediciones subjetivas o cualitativas que no capturan las falencias del sistema. Por ello, se requiere un enfoque sistemático para evaluar la efectividad y facilidad de uso de plataformas educativas (Usability research in educational technology: A state-of-the-art systematic review)

En cuanto al diseño de horarios académicos, estudios han propuesto métodos automáticos que consideren restricciones académicas, preferencias del estudiante y eficiencia espacial.

Por ejemplo, el trabajo *Designing a smart system for building student's schedule* propone un sistema que analiza el rendimiento y restricciones del estudiante para sugerir conjuntos de materias que minimicen conflictos horarios. Allí se integran reglas de prioridad, pre requisitos y límites de carga horaria (Saeed & Fahim, 2021)

Estos antecedentes muestran que la problemática de la no organización del horario académicos es global, y que las soluciones tecnológicas más efectivas combinan diseño centrado en el usuario y arquitecturas flexibles que faciliten integración y mantenimiento.

Teorías y Modelos Relevantes

Para sustentar teóricamente la propuesta, se deben revisar modelos comunes y ya establecidos por universidades y escuelas alrededor del mundo.

Tenemos como un primer ejemplo el modelo DeLone y McLean, usado frecuentemente como punto de partida en estudios de sistema de información. La base de este propone que, la calidad del sistema, información y servicio influyen en el uso del mismo y la satisfacción de los usuarios sobre el mismo. En el contexto de esta investigación, la calidad del backend (rendimiento, fiabilidad), la calidad de los datos (oferta académica correcta) y la calidad de la interfaz (experiencia del estudiante) serán determinantes para el éxito del sistema.

Otra teoría clave es la Teoría de la Aceptación de la Tecnología (TAM, Davis, 1989), que postula que la percepción de utilidad y la percepción de facilidad de uso influyen en la intención de uso de un sistema. En proyectos educativos, muchos estudios extienden el TAM incluyendo

factores como la usabilidad medida con escalas estandarizadas (por ejemplo, el uso de la System Usability Scale para medir la facilidad percibida) (Revythi & Tselios,2017)

Por último, se plantearán 3 ejemplos de modelos de diseños de sistemas académicos que incluyen arquitectura de software (Microservicios, API REST), bases de datos y simuladores de sistemas institucionales.

- En el lado del backend y arquitectura, Newman (2021) plantea que los microservicios ofrecen modularidad, escalabilidad y facilidad de mantenimiento, lo que es especialmente útil cuando se espera evolucionar el sistema en fases sucesivas.
- En cuanto al diseño de bases de datos, Elmasri & Navathe (2016) ofrecen fundamentos sobre la normalización y el manejo eficiente de transacciones.
- En el ámbito de simulación institucional, los entornos virtuales que replican la lógica de sistemas reales (por ejemplo, SAP) permiten validar hipótesis sin comprometer datos reales (Bass, Clements & Kazman, 2013).

Síntesis del enfoque teórico para el proyecto

Este marco teórico orienta la forma como abordamos el desarrollo del sistema:

1. Desde el modelo de DeLone y McLean estarás atento a la calidad de cada componente del sistema (datos, tecnología, servicio) y su impacto en la satisfacción del estudiante y en el uso del sistema.

2. La TAM te ayudará a formular hipótesis sobre adopción del sistema: un estudiante adoptará la plataforma si la percibe útil y fácil de usar.
3. Las dimensiones de usabilidad (navegabilidad, interactividad, capacidad de respuesta, etc.) te servirán de guía para el diseño de la interfaz y para estructurar las pruebas de usabilidad.
4. En el plano técnico, optarán por una arquitectura modular (microservicios o servicios desacoplados) con APIs para facilitar futuras integraciones, mientras que la base de datos relacional será diseñada con normalización y soporte ACID para garantizar integridad y eficiencia.
5. El entorno de simulación te permitirá evaluar tempranamente las funcionalidades del sistema (preinscripción, validación, calendario) sin depender del sistema real institucional y sin riesgo de afectar datos reales

Restricciones

Ahora, hablando del desarrollo de una solución tecnológica en forma de plataforma de apoyo y simulación en el proceso de inscripción de horario académico y organización de materias, está sujeto a restricciones de tipo ambiental, económico, legal, social y técnico. Estas deben ser consideradas desde la parte de diseño para garantizar una viabilidad y sostenibilidad del proyecto.

Restricciones ambientales

- La eficiencia energética constituye una restricción relevante en el diseño y operación de la plataforma propuesta. Aunque se trata de un sistema académico de alcance limitado, su desarrollo e implementación deben considerar el consumo de recursos computacionales tanto en el entorno local como en servidores externos, pues es bien sabido, que muchas de los recursos que se utilizan para mantener los servicios, consumen un alto nivel de recursos naturales (agua y energía).

Restricciones económicas

- A tratarse de un proyecto de grado e independiente desarrollado por 4 personas, se cuenta con poco presupuesto, por lo que se propone desarrollar la idea usando herramientas de código abierto y libres para evitar costos elevados en licencias. Cualquier fluctuación económica, como alzas en costos de hosting, puede afectar el alcance del proyecto.

Restricciones legales

- El sistema debe cumplir con la Ley 1581 de 2012 sobre protección de datos personales en Colombia y con las políticas internas de la Universidad EAN respecto al manejo de información académica. El uso de datos reales de estudiantes estará sujeto a consentimientos informados y protocolos de seguridad. Además, cualquier integración con SAP debe realizarse bajo los términos legales y contractuales vigentes de la universidad.

Restricciones Socioculturales

- La solución debe adaptarse a los hábitos y preferencias de los estudiantes. Si la plataforma es demasiado compleja o no resulta intuitiva, será rechazada por los usuarios. Además, debe contemplar la diversidad cultural y el acceso a la tecnología: estudiantes que no cuentan con computadores de alto rendimiento o con conexiones rápidas a internet también deben poder utilizarla desde dispositivos móviles de gama media.

Restricciones internas

- El equipo de desarrollo está compuesto por estudiantes, lo que implica limitaciones en tiempo, experiencia técnica y recursos humanos. Esto obliga a modular el proyecto, priorizar funcionalidades críticas y aplicar metodologías ágiles para avanzar de manera incremental.

Análisis de Requerimientos

Tabla 1 *Historias De Usuario - HU001*

HISTORIA DE USUARIO		
HU001	Visualizar oferta académica	
Puntos de historia estimados		3
Prioridad	Alta	
Responsable	Desarrollador	
Descripción		
Como estudiante, quiero visualizar toda la oferta académica disponible, para poder elegir las materias que deseo inscribir.		
Numero de criterio	Criterio de aceptación	Atributo de calidad
1	El sistema debe mostrar todas las materias con nombre, NRC, profesor, salón, días y horas.	Usabilidad
2	El usuario debe poder filtrar por nombre, facultad o tipo de materia.	Fiabilidad
3	El sistema debe mostrar únicamente materias con información válida cargada desde el sistema académico.	usabilidad

Tabla 2 *Historias De Usuario - HU002*

HISTORIA DE USUARIO		
Identificador	Nombre	
HU002	Realizar preinscripción de materias	
Puntos de historia estimados		5
Prioridad	Alta	
Responsable	Desarrollador	
Descripción		
Como estudiante, quiero poder seleccionar las materias que deseo ver durante el semestre, para construir mi preinscripción del semestre.		
Numero de criterio	Criterio de aceptación	Atributo de calidad
1	El sistema debe permitir agregar una materia con un solo clic.	Usabilidad
2	Debe validarse disponibilidad de cupo antes de agregar la materia.	Fiabilidad
3	La materia debe aparecer inmediatamente en el horario del estudiante.	Usabilidad

Tabla 3 *Historias De Usuario - HU003*

HISTORIA DE USUARIO		
Identificador	Nombre	
HU003	Detección de choques de horario	
Puntos de historia estimados		5
Prioridad	Alta	
Responsable	Desarrollador	
Descripción		
Como estudiante, quiero que el sistema me notifique si existe un choque de horario al agregar una materia.		
Numero de criterio	Criterio de aceptación	Atributo de calidad
1	Si el día y hora ya están ocupados, el sistema debe impedir la preinscripción.	Funcionalidad
2	Debe mostrarse un mensaje claro indicando el conflicto.	Usabilidad
3	La materia no debe agregarse si existe un choque	Fiabilidad

Tabla 4 Historias De Usuario - HU004

HISTORIA DE USUARIO		
Identificador	Nombre	
HU004	Crear y gestionar hasta cuatro horarios	
Puntos de historia estimados		8
Prioridad	Media	
Responsable	Desarrollador	
Descripción		
Como estudiante, quiero poder crear y gestionar hasta cuatro horarios alternativos.		
Numero de criterio	Criterio de aceptación	Atributo de calidad
1	El usuario debe poder seleccionar en qué horario (1 a 4) desea agregar una materia.	Usabilidad
2	Cada horario debe mostrarse de forma independiente.	Fiabilidad
3	El estudiante debe poder eliminar materias sin afectar otros horarios.	Usabilidad

Tabla 5 Historias De Usuario - HU005

Identificador	Nombre	
HU005	Iniciar sesión	
Puntos de historia estimados		3
Prioridad	Alta	
Responsable	Desarrollador	
Descripción		
Como estudiante, quiero iniciar sesión con mi usuario institucional para acceder a mi información y guardar mis horarios.		
Numero de criterio	Criterio de aceptación	Atributo de calidad
1	El sistema debe validar correo y contraseña registrados.	Seguridad
2	Si los datos son correctos, el sistema debe permitir el ingreso.	Usabilidad
3	Si los datos son incorrectos, se mostrará mensaje de error.	usabilidad

Tabla 6 Historias De Usuario - HU006

HISTORIA DE USUARIO	
Identificador	Nombre

HU006	Ver horario en un calendario visual	
Puntos de historia estimados		5
Prioridad	Media	
Responsable	Front end	
Descripción		
Como estudiante, quiero ver mi horario organizado visualmente en un calendario tipo tabla.		
Numero de criterio	Criterio de aceptación	Atributo de calidad
1	El calendario debe mostrar días como columnas y horas como filas.	Usabilidad
2	Las materias deben visualizarse como bloques.	Usabilidad
3	El horario debe actualizarse automáticamente al agregar o eliminar materias.	Fiabilidad

Tabla 7 Historias De Usuario - HU007

HISTORIA DE USUARIO	
Prioridad	Alta
Responsable	Backend responsable de programación académica
Descripción	
Como responsable de programación académica, quiero cargar la oferta académica desde un archivo Excel.	

Numero de criterio	Criterio de aceptación	Atributo de calidad
1	El sistema debe leer correctamente las columnas y contenido de estas del archivo Excel.	Fiabilidad
2	Las materias deben almacenarse en la base de datos.	Fiabilidad
3	Los datos deben estar disponibles inmediatamente para el frontend.	Usabilidad

Tabla 8 Historias De Usuario - HU008

HISTORIA DE USUARIO	
Identificador	Nombre
HU008	Acceso restringido a usuarios registrados
Puntos de historia estimados	
3	

Prioridad	Alta	
Responsable	Backend / Responsable de programación académica	
Descripción		
Como responsable de programación académica, quiero que el sistema consulte la oferta académica en tiempo real.		
Numero de criterio	Criterio de aceptación	Atributo de calidad
1	El backend debe exponer un servicio que entregue la oferta académica procesada.	Fiabilidad
2	Los cambios deben reflejarse en el frontend sin recargar manualmente la base de datos.	Usabilidad

Tabla 9 Historias De Usuario - HU009

HISTORIA DE USUARIO	
Identificador	Nombre
HU009	Consultar oferta académica en tiempo real
Puntos de historia estimados	5

Prioridad	Alta	
Responsable	Backend / Responsable de programación académica	
Descripción		
Como responsable de programación académica, quiero que el sistema consulte la oferta académica en tiempo real.		
Numero de criterio	Criterio de aceptación	Atributo de calidad
1	El aplicativo debe poder integrarse con los usuarios en el directorio activo de la universidad.	Integrabilidad
2	Los endpoints sensibles deben exigir autenticación.	Seguridad

Metodología Para el Desarrollo de la Solución

Para el desarrollo de la solución propuesta, se tomó como referencia el ciclo de vida del software, empleando un enfoque estructurado que permitiera organizar, planificar y ejecutar de manera ordenada cada una de las actividades necesarias para la construcción del sistema. Este ciclo se implementó siguiendo una metodología ágil, utilizando principalmente los principios del marco de trabajo Scrum, lo que facilitó la adaptabilidad y la mejora continua durante el proceso.

Scrum se caracteriza por organizar el trabajo en periodos cortos llamados sprints, dentro de los cuales el equipo define, desarrolla y entrega funcionalidades específicas del producto. Esta metodología fomenta la comunicación constante mediante reuniones breves diarias, planificación colaborativa y revisiones periódicas, lo que permitió detectar problemas de manera temprana, reorientar tareas y asegurar que cada avance estuviera alineado con los objetivos del proyecto.

Además, Scrum integra actividades de retrospectiva en cada sprint, lo que contribuyó a evaluar el desempeño del equipo, optimizar procesos y consolidar una cultura de mejora continua.

La implementación de esta metodología fue especialmente útil dado el carácter evolutivo del proyecto. La plataforma debía integrarse con la oferta académica oficial proporcionada en un archivo Excel actualizado por la Universidad EAN, lo que implicó la necesidad de realizar ajustes frecuentes en la estructura de datos, los modelos de base de datos y los servicios del backend. Gracias al enfoque ágil, fue posible incorporar cambios sobre la marcha, modificar funcionalidades según los nuevos requerimientos institucionales y validar constantemente el comportamiento correcto del sistema.

Durante el desarrollo se llevaron a cabo varias reuniones de seguimiento donde se revisaron avances, se priorizaron tareas y se reorganizaron las actividades según las necesidades emergentes. Esto permitió mantener un ritmo de trabajo estable, asegurar que los entregables fueran consistentes y cumplir los objetivos dentro del tiempo estimado.

En conjunto, la metodología adoptada permitió gestionar adecuadamente la complejidad del proyecto, promover la colaboración efectiva entre los integrantes y garantizar una entrega progresiva y validada del prototipo final.

Desarrollo de la Solución

El desarrollo de la plataforma de preinscripción académica se llevó a cabo con el propósito de facilitar a los estudiantes de la Universidad EAN la organización de sus horarios antes de la inscripción oficial en el sistema institucional.

El proyecto se construyó bajo una arquitectura de microservicios, utilizando FastAPI para el backend y React para la interfaz gráfica, lo que permitió trabajar de forma modular y flexible.

En una primera etapa, se definieron los requerimientos funcionales y no funcionales del sistema, estableciendo qué debía hacer la aplicación y en qué condiciones debía operar. Luego, se diseñó la base de datos con SQLite, contemplando las entidades principales: materias, horarios, carreras, usuarios y preinscripciones.

Posteriormente, se desarrolló un servicio encargado de leer y procesar la oferta académica oficial de la universidad, que se encuentra en un archivo Excel. Este servicio limpia los datos, elimina información duplicada y la estructura de forma que puedan ser utilizados dentro del sistema.

Con esta base, se implementó el módulo principal: el servicio de preinscripción, el cual permite a los estudiantes registrar materias, visualizar horarios y detectar automáticamente conflictos cuando dos asignaturas coinciden en el mismo día y hora.

Durante el proceso de desarrollo se incorporaron nuevos campos, como las fechas de inicio y finalización de cada materia, lo que permitió mejorar la precisión del calendario académico. Además, se actualizaron los modelos de datos para evitar duplicaciones y mantener la base de datos organizada.

Las pruebas se realizaron de manera local, utilizando SQLite como base de datos y herramientas como Postman y Swagger UI para validar los endpoints. Esto permitió comprobar la comunicación entre el backend y el frontend, la correcta detección de choques de horario y la funcionalidad de eliminar o modificar preinscripciones.

Análisis de costos

Para estimar el esfuerzo necesario para el desarrollo de la plataforma, se analizaron los costos involucrados desde una perspectiva integral. En primer lugar, se consideran los costos directos, que corresponden a los recursos empleados de manera inmediata en la creación del sistema. Aquí se incluyen el tiempo dedicado al desarrollo del backend y el frontend, la construcción y configuración de la base de datos, la limpieza y organización del archivo Excel institucional y la integración de los diferentes módulos del proyecto. También forman parte de este grupo los equipos utilizados y la conexión a internet requerida durante todo el proceso de implementación.

Además de estos gastos, existen costos que permanecen constantes sin importar el ritmo de avance del proyecto. Estos costos fijos se relacionan con el consumo eléctrico de los equipos, el mantenimiento básico del hardware y el uso de herramientas de colaboración y control de versiones como GitHub, las cuales son necesarias tanto en las fases activas de trabajo como en los momentos de planificación.

A estos se suman los gastos generales, que abarcan diversas actividades de apoyo indispensables para que el proyecto avance de manera organizada. Entre ellas se encuentran la planificación del desarrollo, la elaboración de la documentación técnica, las reuniones de coordinación con estudiantes y docentes, y la verificación de los requerimientos y resultados. Aunque no influyen directamente en la programación, son fundamentales para asegurar que la solución cumpla con las expectativas y necesidades de los usuarios.

Finalmente, se contempla la inversión inicial, que corresponde al tiempo y esfuerzo invertido en las etapas de análisis, diseño de la arquitectura, definición de los esquemas de datos y configuración de los entornos de desarrollo. Esta fase también incluye las primeras pruebas

funcionales que permiten validar la coherencia entre los componentes del sistema. En conjunto, todo el proceso de construcción de la plataforma tomó aproximadamente entre 8 y 10 semanas, dependiendo de la disponibilidad del equipo de trabajo y de los ajustes necesarios en cada etapa.

En suma, a pesar de los diferentes tipos de costos involucrados, la solución resultante demuestra ser eficiente y de bajo costo, con un impacto significativo en la organización académica de los estudiantes y una notable reducción de errores y tiempos en el proceso de inscripción.

Tabla 10 Tabla de costos

<i>Item</i>	Horas trabajadas	Valor de la hora	Total
Horas de análisis y diseño	40	\$ 120.000	\$ 4.800.000
Horas de desarrollo del front-end	180	\$ 100.000	\$ 18.000.000
Horas de desarrollo del backend	100	\$ 100.000	\$ 10.000.000
Horas de diseño de UI/UX	120	\$ 70.000	\$ 8.400.000
Valor total			\$ 41.200.000

Tabla 11 Servicios públicos

<i>Nombre</i>	<i>Valor</i>
Servicio de internet	\$ 75.000
Pago de servicios públicos	\$ 68.000
Valor total	\$ 41.375.700

Arquitectura del Software

La arquitectura del software constituye el eje central que permite comprender cómo los diferentes componentes del sistema interactúan entre sí para ofrecer una solución funcional, escalable y mantenible. En esta sección se presenta la estructura general del proyecto, destacando la división lógica del backend, la gestión de datos, la comunicación entre servicios y la relación con el frontend. A través de diagramas y representaciones visuales, se busca ilustrar de forma clara cómo se organizó el sistema, cómo fluyen los datos desde la extracción de la oferta académica hasta la preinscripción de materias, y cómo cada módulo cumple un rol específico dentro de la solución. Esta descripción arquitectónica no solo facilita el entendimiento técnico, sino que también evidencia la coherencia y solidez del diseño implementado.

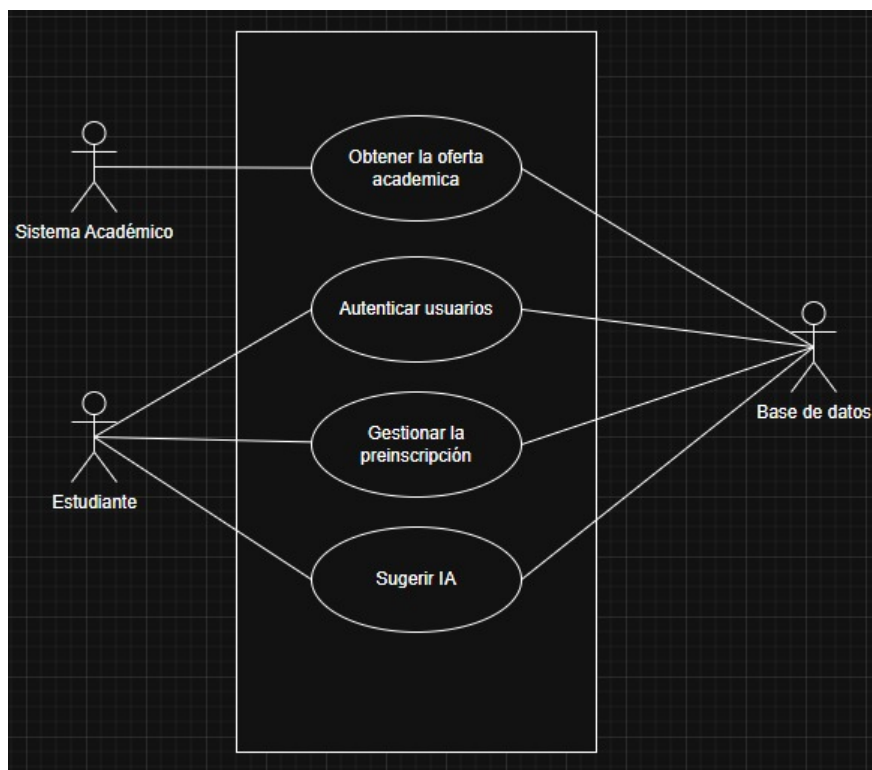


Figura 1 Diagrama de casos de uso realizado en Draw.io

El gráfico ilustra las funcionalidades más importantes del backend del sistema Semestria. El Sistema Académico y el Estudiante, que interactúan con los distintos módulos del sistema, son dos actores externos que se identifican en él. Cuatro procesos esenciales son centralizados por el backend: gestionar la preinscripción, autenticar a los usuarios, obtener la oferta académica y crear sugerencias basadas en inteligencia artificial. La dependencia del sistema con respecto a la información institucional para asegurar la integridad, disponibilidad y consistencia de las operaciones se manifiesta en que cada uno de estos casos de uso tiene comunicación directa con la base de datos.

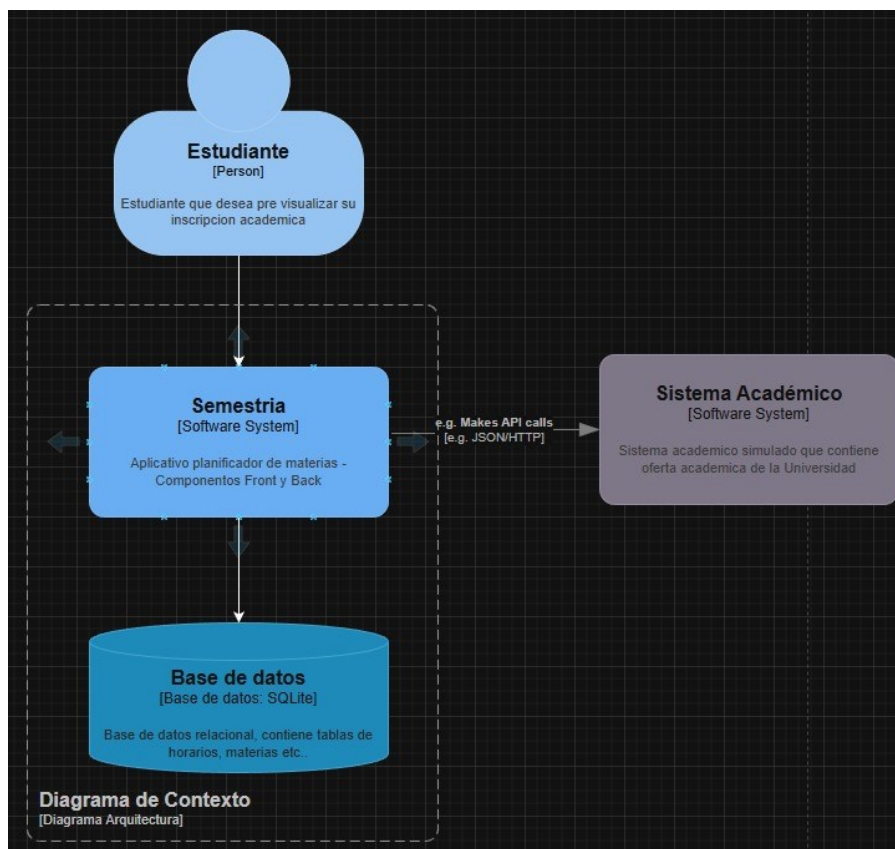


Figura 2 Diagrama de contexto realizado en Draw.io

La estructura general de Semestria, una aplicación creada para respaldar el proceso de planificación e inscripción académica es explicada por el diagrama de contexto que se muestra. El estudiante, que es el usuario principal y consulta para ver su inscripción, se encuentra en la parte superior. Este se comunica directamente con Semestria, que combina los elementos frontend y backend requeridos para tramitar las peticiones.

Se nota la interacción entre Semestria y su base de datos interna, que está implementada en SQLite. En esta base se guardan datos estructurados como horarios, materias y otras informaciones necesarias para la administración académica. Además, el diagrama ilustra cómo

el sistema se comunica con el Sistema Académico de la universidad, que es un sistema simulado y ofrece la oferta académica de la institución.

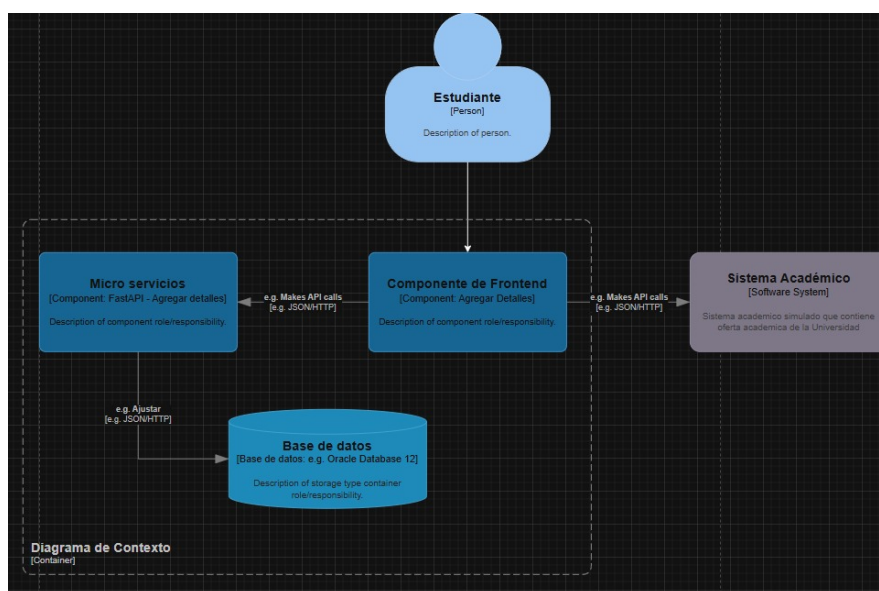


Figura 3 Diagrama de contexto del backend

La estructura interna del sistema se presenta a través del diagrama de contexto, que muestra cómo el estudiante se comunica con el componente de frontend, responsable de administrar la interfaz y remitir las solicitudes al backend. Este frontend se comunica con un conjunto de microservicios que se implementaron para gestionar la lógica del sistema y llevar a cabo las operaciones de inscripción. Por otra parte, los microservicios se comunican con la base de datos, que tiene la responsabilidad de guardar los datos operativos y académicos. El sistema también contacta a través de API el Sistema Académico de la universidad para conseguir la oferta educativa. En general, el diagrama muestra una arquitectura modular que divide la presentación, la lógica de negocio y la persistencia, lo cual posibilita un sistema más organizado y escalable.

Figura 4

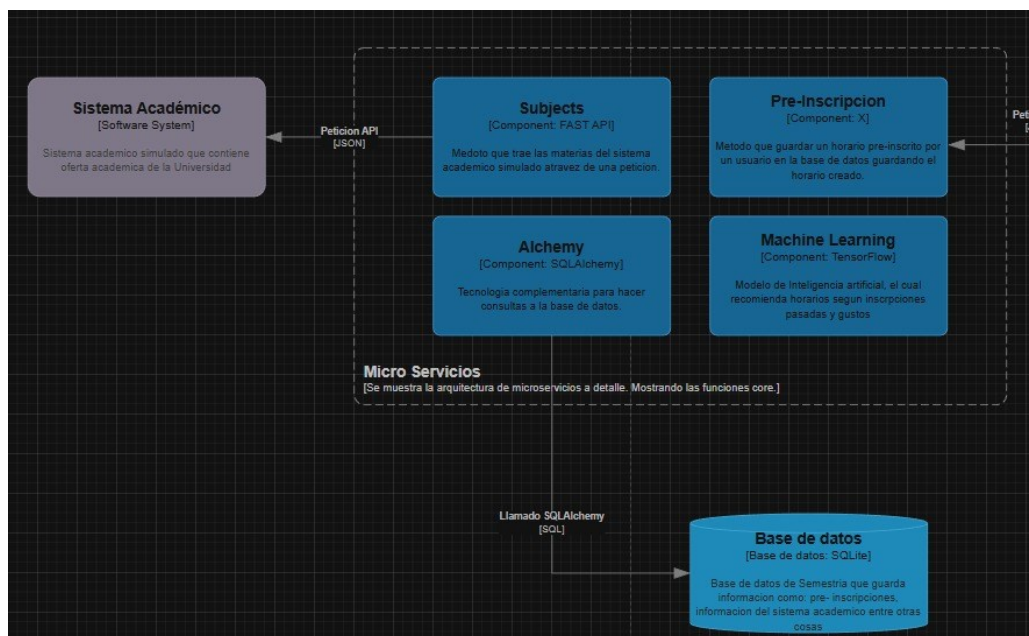


Figura 4 Diagrama de contexto microservicios

El diagrama detalla la arquitectura interna de los microservicios del sistema. Se muestran los componentes encargados de las funciones principales: Subjects, que obtiene la oferta académica mediante una petición al Sistema Académico; Pre-Inscripción, que registra los horarios creados por el usuario; Alchemy, que gestiona las consultas a la base de datos; y el módulo de Machine Learning, encargado de generar recomendaciones de horarios. Todos estos microservicios se comunican con la base de datos del sistema para almacenar o consultar información académica y operativa.

Tabla 12 Modelos y Métodos de Semestria

Modulo/Archivo	Función/Método	Descripción
data_loader.py	extract_subjects_from_excel(file_path)	Lee el archivo Excel oficial, normaliza columnas y devuelve una lista de materias con horarios.
data_loader.py	_normalize_time(value)	Convierte horas provenientes del Excel a formato estándar HH:MM.
cargar_excel.py	cargar_datos_desde_excel()	Inserta o actualiza materias y horarios en la base de datos usando el Excel cargado.
routes/subjects.py	load_data_from_excel()	Carga en memoria los datos del Excel para las rutas de consulta.
routes/subjects.py	get_all_subjects()	Retorna todas las materias leídas del Excel.
routes/subjects.py	get_subject(id)	Busca y retorna una materia por su ID único del Excel.
routes/preinscripcion.py	has_time_conflict(db, estudiante_id, materia_id)	Valida si la materia entra en choque con otra del mismo horario.

routes/preinscripcion.py	preinscribir(data)	Registra una preinscripción validando cupos, conflictos y horario seleccionado.
routes/preinscripcion.py	eliminar_preinscripcion(id)	Permite eliminar una materia previamente añadida al horario.
routes/preinscripcion.py	get_preinscripciones(estudiante_id)	Devuelve todas las materias preinscritas por un estudiante.
routes/auth.py	register_user(data)	Registra un nuevo usuario verificando si el correo ya existe.
routes/auth.py	login(request)	Valida credenciales y genera token de acceso.
routes/auth.py	me(token)	Devuelve la información del usuario autenticado.
models.py	Usuario	Modelo de usuario con nombre, correo, contraseña hash y rol.
models.py	Materia	Modelo principal de materias: ID, código, nombre, ciclos, grupo, horarios, etc.
models.py	HorarioMateria	Modelo asociado a cada materia con día, hora inicio, hora fin y ciclo.

models.py	Preinscripcion	Modelo que relaciona usuario, materia y versión del horario.
security.py	create_access_token()	Genera un JWT para sesiones de usuarios.
security.py	verify_password()	Compara contraseña ingresada con la almacenada en hash.
security.py	get_current_user()	Extrae el usuario autenticado desde el token.
utils/validators.py	verify_cupos(materia_id)	Valida si la materia aún tiene disponibilidad.
utils/validators.py	validate_schedule_format()	Validación interna de formato de días y horas.

Swagger

Conclusiones

El desarrollo de Semestria permitió evidenciar cómo la ingeniería de software, aplicada con una metodología estructurada y soportada en buenas prácticas, puede transformar procesos que tradicionalmente presentan dificultades para los estudiantes. El sistema optimiza la planeación académica al ofrecer una visualización clara y actualizada de la oferta de materias, permitiendo identificar cruces de horario y organizar la carga académica antes de la inscripción oficial. Esto reduce la incertidumbre, agiliza la toma de decisiones y disminuye el riesgo de errores comunes durante el proceso.

La arquitectura basada en microservicios demostró ser una elección adecuada para garantizar escalabilidad, modularidad y facilidad de mantenimiento. Gracias a esta estructura, la plataforma está preparada para integrarse en el futuro con otros sistemas institucionales, como SAP, así como para incorporar nuevas funcionalidades, por ejemplo, algoritmos de recomendación o análisis predictivo para la construcción automática de horarios.

La automatización del procesamiento del archivo oficial de la universidad aseguró que los datos utilizados por el sistema sean consistentes, confiables y estén siempre actualizados. Esto no solo mejora la experiencia del usuario, sino que también reduce la carga operativa asociada a la gestión manual de la información.

Finalmente, el proyecto demuestra la importancia de un enfoque integral en la ingeniería de software. A través del análisis, diseño, implementación y validación, se logró construir una herramienta funcional y pertinente para las necesidades de los estudiantes. Además de su aporte tecnológico, la plataforma ofrece un valor académico y práctico al contribuir directamente a la digitalización y modernización de los procesos de gestión académica dentro de la Universidad EAN.

Referencias

Davis, F. D. (1989). *Perceived usefulness, perceived ease of use, and user acceptance of information technology*. MIS Quarterly, 13(3), 319–340.

DeLone, W. H., & McLean, E. R. (2003). *The DeLone and McLean model of information systems success: A ten-year update*. Journal of Management Information Systems, 19(4), 9–30.

Implications of immersive scheduling for student engagement. (2023). Studies in Higher Education, 48(6), 1120–1138.

Gamale, E., Anuta, J., & Sayson, S. (2012). *Automated class scheduling system*. International Journal of Computer Science and Information Technology, 4(2), 45–52.

Mostefai, S., et al. (2025). *Enhancing user experience in e-learning systems: A new approach*. Journal of Educational Technology & Society, 28(1), 112–129.

Ramírez, C., & Ochoa, L. (2022). *Limitaciones de los sistemas tradicionales de inscripción académica en universidades colombianas*. Revista Colombiana de Ingeniería, 10(1), 55–72.

Revythi, A., & Tselios, N. (2017). *Extension of technology acceptance model by using system usability scale to assess behavioral intention to use e-learning*. Journal of Computers in Education, 4(3), 307–323.

Saeed, S., & Fahim, M. (2021). *Designing a smart system for building students' schedule*. International Journal of Advanced and Applied Sciences, 8(1), 45–52.

The complete guide to scheduling software for colleges. (2023). Modern Campus.

- Usability research in educational technology: A state-of-the-art systematic review.* (2022). *Educational Technology Research Review*, 14(3), 200–230.
- Van der Meer, J., & Jensen, K. (2024). *Optimizing student scheduling with AI: Lessons from European universities.* *Higher Education Technology Review*, 12(1), 33–49.
- Vlachogianni, P., & Tselios, N. (2023). *Perceived usability evaluation of educational technology.* *Sustainability*, 15(17), 12954.
- Vlasenko, E., et al. (2023). *Exploring usability principles for educational online courses: A case study on an open platform for online education.* *International Journal of Educational Technology*, 12(2), 88–105.
- Wong, M., Teo, H., & Chua, S. (2023). *Digital platforms for course registration: A usability perspective.* *International Journal of Educational Management*, 37(4), 567–584.
- Beyond technological usability: CAUSLT—A contextual approach to usability in learning technologies.* (2025). *Journal of Learning Sciences*, 31(2), 221–245.