

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

**DESARROLLO DE UN SIMULADOR DE CLASES
PERSONALIZADAS CON IA GENERATIVA PARA EL
APRENDIZAJE UNIVERSITARIO**

DISEÑO DE LA INTERFAZ Y EXPERIENCIA DE USUARIO (UX/UI)

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN CIENCIAS DE LA COMPUTACIÓN**

HERNÁN DARÍO SÁNCHEZ TENELANDA
hernan.sanchez@epn.edu.ec

DIRECTOR: ENRIQUE ANDRÉS LARCO AMPUDIA, PhD.
andres.larco@epn.edu.ec

DMQ, diciembre 2025

CERTIFICACIONES

Yo, **HERNÁN DARÍO SÁNCHEZ TENELANDA** declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

Asimismo, declaro que he utilizado la herramienta de inteligencia artificial [CHATGPT] únicamente como apoyo en [BÚSQUEDA Y SÍNTESIS DE INFORMACIÓN], sin atribuirle autoría, y que todo el contenido derivado ha sido revisado, validado y es de mi exclusiva responsabilidad.

HERNÁN DARÍO SÁNCHEZ TENELANDA

Certifico que el presente trabajo de integración curricular fue desarrollado por **HERNÁN DARÍO SÁNCHEZ TENELANDA**, bajo mi supervisión.

ENRIQUE ANDRÉS LARCO AMPUDIA, PhD
DIRECTOR

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

HERNÁN DARÍO SÁNCHEZ TENELANDA

ENRIQUE ANDRÉS LARCO AMPUDIA, PhD.

CARLOS ANDRÉS CÓRDOVA ACARO

MICHAEL ANDRÉS PILLAGA SOSA

CARLA ANNAI RUIZ ACEVEDO

DEDICATORIA

(Opcional)

AGRADECIMIENTOS

(Opcional)

ÍNDICE DE CONTENIDO

CERTIFICACIONES	I
DECLARACIÓN DE AUTORÍA	I
DEDICATORIA	III
AGRADECIMIENTOS	IV
ÍNDICE DE CONTENIDO	V
RESUMEN	VII
ABSTRACT	VIII
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO	1
1.1 Objetivo general	2
1.2 Objetivos específicos	2
1.3 Alcance	2
1.4 Marco Teórico	3
1.4.1 Experiencia de Usuario (UX)	3
1.4.2 Interfaz de Usuario (UI)	5
1.4.3 Inteligencia Artificial Generativa	6
1.4.4 APIs	7
1.4.5 IA y Educación en Universidades	8
1.4.6 Interacción Humano–Computador (HCI)	9
2 METODOLOGÍA	12
2.1 Metodología SCRUM	12
2.1.1 Definición y asignación de roles	15
2.1.2 Planificación y organización de los sprints	16
2.2 Tecnologías Utilizadas	17
2.2.1 Herramientas Utilizadas	18
2.3 Etapa de Desarrollo	19
2.3.1 Sprint 0: Análisis inicial y definición del enfoque UX/UI	19
2.3.2 Sprint 1: Análisis de herramientas de IA generativa y levantamiento de requerimientos	19

2.3.3 Sprint 2: Diseño de prototipos de baja fidelidad y estructura de navegación	25
3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES	27
3.1 Resultados	27
3.2 Conclusiones	28
3.3 Recomendaciones	28
4 REFERENCIAS BIBLIOGRÁFICAS	29
5 ANEXOS	32
ANEXO I: Aplicación Móvil	32
ANEXO II: Aplicación Web	33
ANEXO III: Página Web	34

RESUMEN

Antecedentes:

Objetivo:

Metodología:

Resultados:

Conclusiones:

Cuando termines de escribir tu resumen, elimina las 5 palabras anteriores que están en negrita.

(Maximo 250 palabras)

PALABRAS CLAVE: palabra 1, palabra 2,....., palabra 6.

ABSTRACT

Background:

Objective:

Methods:

Results:

Conclusions:

When you finish writing your abstract, delete the previous five bold words.
(Maximo 250 palabras).

KEYWORDS: word1, word2, . . . , word6.

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

El componente desarrollado estuvo centrado al diseño e implementación de la interfaz gráfica y de la experiencia de usuario (UX/UI) del Simulador de Clases Personalizadas con Inteligencia Artificial Generativa. Debido a que esta herramienta está pensada para apoyar a estudiantes universitarios con diferentes niveles de conocimiento y formas de aprendizaje, fue necesario plantear una interfaz que resultara clara, accesible y cómoda de utilizar desde el primer uso.

Desde las primeras etapas del proyecto se tuvo presente que la interfaz no debía limitarse a un diseño visual atractivo. Su función principal era servir como un puente entre el estudiante y el sistema, facilitando la interacción y el acceso a los contenidos sin generar confusión o dificultades innecesarias. Con este enfoque, se inició una fase de análisis en la que se revisaron distintas plataformas educativas y herramientas basadas en inteligencia artificial generativa, con el objetivo de identificar prácticas efectivas y elementos que pudieran adaptarse al contexto del simulador.

A partir de este análisis se definieron las funcionalidades principales de la plataforma y se desarrollaron prototipos con un enfoque centrado en el usuario. Estos diseños preliminares priorizaron una navegación sencilla y una estructura organizada, permitiendo que los estudiantes accedieran de forma rápida a secciones clave como las clases personalizadas, la retroalimentación inmediata y los ejercicios adaptativos. Se prestó especial atención a aspectos como la legibilidad de los textos, el contraste de colores, el uso adecuado de íconos y la distribución clara de la información, buscando una experiencia de uso coherente y comprensible.

Durante el proceso de desarrollo se realizaron pruebas de usabilidad con estudiantes universitarios, las cuales resultaron fundamentales para evaluar el funcionamiento real de la interfaz. Estas pruebas permitieron observar directamente cómo los usuarios interactuaban con la plataforma, identificar dificultades recurrentes y recoger sugerencias de mejora. A partir de estos resultados, se realizaron ajustes en distintos elementos del diseño, como la ubicación de ciertos componentes, la claridad de botones e instrucciones y la optimización de los tiempos de carga.

La interfaz final permite a los estudiantes explorar de manera sencilla el contenido generado por la inteligencia artificial, solicitar explicaciones adicionales cuando lo consideran necesario y practicar mediante ejercicios que se adaptan a su ritmo de aprendizaje. Todo

ello se presenta en un entorno visual ordenado y funcional, diseñado para reducir la carga cognitiva y facilitar el proceso de aprendizaje.

Este componente fue esencial para el correcto funcionamiento del simulador, ya que permitió transformar la complejidad técnica del sistema en una herramienta accesible y fácil de utilizar. Además, cada etapa del desarrollo fue documentada de forma detallada, lo que facilitará futuras mejoras, ajustes o integraciones con otras plataformas educativas.

1.1 Objetivo general

Diseñar una interfaz gráfica accesible y funcional para el Simulador de Clases Personalizadas con IA generativa, que facilite la experiencia de usuario y la interacción entre estudiantes y el sistema.

1.2 Objetivos específicos

1. Investigar y analizar herramientas de inteligencia artificial generativa para identificar buenas prácticas en diseño de interfaces.
2. Diseñar e implementar una interfaz gráfica que permita navegación fluida, interacción y visualización del contenido generado.
3. Realizar pruebas de usabilidad con estudiantes universitarios, para mejorar la experiencia de usuario e identificar posibles mejoras.

1.3 Alcance

El alcance del componente desarrollado incluyó la construcción y puesta en funcionamiento de la interfaz gráfica y la experiencia de usuario (UX/UI) del simulador de clases personalizadas con inteligencia artificial generativa, de acuerdo con las directrices establecidas en el Plan de Trabajo de Integración Curricular. Este módulo fue concebido como el principal punto de interacción entre los estudiantes y el sistema, por lo que su diseño se orientó a ofrecer una experiencia clara, accesible y alineada con los objetivos formativos del proyecto.

Dentro de las actividades realizadas se contempló, en primer lugar, la revisión de distintas

herramientas de inteligencia artificial generativa, con el fin de analizar su posible integración a nivel de interfaz. Paralelamente, se llevó a cabo un análisis de las necesidades de interacción de estudiantes universitarios, lo que permitió identificar patrones de uso y expectativas frente a este tipo de plataformas. Con base en esta información, se definieron los requerimientos técnicos y funcionales que guiaron el desarrollo del componente.

A partir de los requerimientos establecidos, se elaboraron prototipos de baja y alta fidelidad, los cuales sirvieron para validar progresivamente las decisiones de diseño. Durante esta etapa se priorizó la facilidad de uso, una navegación estructurada de manera lógica y una presentación visual clara del contenido generado por la inteligencia artificial, procurando que el acceso a las funcionalidades del simulador resultara intuitivo para el usuario.

La fase de implementación se llevó a cabo utilizando tecnologías web actuales, lo que permitió una integración adecuada con los módulos principales del simulador, tales como la generación de clases, la retroalimentación automática y el sistema de evaluación adaptativa. Adicionalmente, se desarrolló una interfaz responsiva que garantiza un funcionamiento correcto en distintos dispositivos, incluyendo computadoras de escritorio, teléfonos móviles y tabletas, favoreciendo su uso en diversos contextos universitarios.

Como parte del proceso de desarrollo, se realizaron evaluaciones de usabilidad con estudiantes universitarios, orientadas a observar su desempeño en tareas específicas y a identificar posibles dificultades de interacción. Los resultados obtenidos sirvieron como base para aplicar mejoras de manera iterativa, ajustando elementos visuales, optimizando los tiempos de carga y reforzando la claridad de los componentes interactivos. Estas acciones contribuyeron a fortalecer la experiencia general ofrecida por la plataforma.

1.4 Marco Teórico

1.4.1 Experiencia de Usuario (UX)

La Experiencia de Usuario (UX, por sus siglas en inglés) hace referencia a las percepciones, emociones y respuestas que una persona desarrolla antes, durante y después de interactuar con un sistema digital. Este concepto no se limita a la facilidad de uso, sino que incorpora factores cognitivos, emocionales y contextuales que influyen directamente en la aceptación y efectividad de una solución tecnológica [1].

De acuerdo con la norma ISO 9241-210, la UX se define como el resultado de las percepciones y respuestas del usuario derivadas del uso real o anticipado de un sistema, considerando emociones, creencias, preferencias, comodidad y logros personales. Esta definición destaca que la experiencia no depende únicamente de la interfaz gráfica, sino también del contexto de uso y del estado interno del usuario, como sus conocimientos previos y expectativas [2].

En el desarrollo de software, la experiencia de usuario se ha convertido en un factor clave de calidad. Un sistema puede ser técnicamente correcto, pero si genera confusión, frustración o sobrecarga cognitiva, su adopción será limitada. En entornos educativos digitales, una mala UX puede afectar la motivación del estudiante y reducir la continuidad del proceso de aprendizaje, mientras que una experiencia bien diseñada favorece la participación y el compromiso académico [3].

Es importante diferenciar la UX de la usabilidad y de la interfaz de usuario (UI). La usabilidad se enfoca en qué tan fácil resulta aprender y utilizar un sistema, mientras que la UX abarca una visión más amplia que incluye la utilidad percibida, la satisfacción emocional y el valor que el usuario atribuye a la interacción. De esta manera, la interfaz actúa como un medio, pero no garantiza por sí sola una experiencia positiva.

Desde un enfoque multidisciplinario, la experiencia de usuario integra aportes de la interacción humano-computador, la psicología cognitiva, el diseño de interacción y la arquitectura de la información. Esta convergencia permite diseñar sistemas alineados con los modelos mentales de los usuarios y reducir la carga cognitiva, aspecto especialmente relevante en plataformas educativas apoyadas por inteligencia artificial.

En el contexto del simulador de clases personalizadas con IA generativa propuesto en este proyecto, la UX cumple un rol estratégico. Una experiencia clara, coherente y accesible facilita que los estudiantes comprendan el funcionamiento del sistema, confíen en la personalización del contenido y utilicen de forma efectiva la retroalimentación automática y la evaluación adaptativa. Por ello, la UX debe entenderse como un proceso continuo de evaluación y mejora, basado en la retroalimentación de usuarios reales.

1.4.2 Interfaz de Usuario (UI)

La Interfaz de Usuario (UI, por sus siglas en inglés) constituye el punto de contacto directo entre las personas y los sistemas informáticos. Es el espacio donde se materializa la interacción humano-computador y donde el usuario interpreta qué puede hacer un sistema y cómo hacerlo. Desde esta perspectiva, la interfaz no es un elemento accesorio, sino un componente esencial que condiciona la comprensión, el uso y la aceptación de cualquier producto digital [4].

Diversos autores coinciden en que una interfaz bien diseñada permite que el usuario actúe de manera natural, sin necesidad de comprender los aspectos técnicos que subyacen al sistema. Para el usuario final, la interfaz representa el sistema en su totalidad; si la interacción resulta confusa o poco clara, el sistema será percibido como ineficiente, independientemente de su correcto funcionamiento interno [5].

La UI se compone de elementos visuales e interactivos como botones, menús, iconos, textos y mecanismos de retroalimentación que guían al usuario en la ejecución de tareas. Estos elementos deben organizarse de forma coherente y consistente, facilitando la creación de un modelo mental claro que permita anticipar el comportamiento del sistema. Cuando la interfaz respeta las expectativas del usuario, la interacción se vuelve fluida y reduce la carga cognitiva durante el uso [6].

Un aspecto clave del diseño de interfaz es reconocer la diversidad de usuarios. Las personas difieren en sus capacidades físicas, cognitivas y en sus niveles de experiencia con la tecnología. Por ello, una UI efectiva debe ser flexible, accesible y tolerante al error, permitiendo tanto a usuarios novatos como experimentados interactuar con el sistema de manera segura y eficiente [7].

Asimismo, la interfaz cumple un rol comunicativo. A través de metáforas visuales, colores, jerarquías y mensajes claros, la UI transmite el estado del sistema y orienta al usuario en cada paso de la interacción. Una retroalimentación adecuada fortalece la sensación de control y confianza, elementos fundamentales para una experiencia de uso positiva.

En el contexto del simulador de clases personalizadas con inteligencia artificial propuesto en este proyecto, la interfaz de usuario adquiere especial relevancia. Una UI clara y profesional permite que los estudiantes comprendan las funcionalidades del sistema, accedan al contenido de forma intuitiva y aprovechen los recursos de personalización sin sentirse abrumados.

mados por la tecnología. De esta manera, la interfaz actúa como un puente que humaniza la tecnología y facilita el proceso de aprendizaje.

1.4.3 Inteligencia Artificial Generativa

La Inteligencia Artificial Generativa (IAG) es una rama de la inteligencia artificial orientada a la creación automática de contenido nuevo a partir del aprendizaje de grandes volúmenes de datos. A diferencia de otros enfoques de la IA, cuyo objetivo principal es clasificar, predecir o recomendar, la IAG se caracteriza por su capacidad para producir textos, imágenes, audio, video o código que resultan coherentes y comparables a los generados por seres humanos [8].

El desarrollo reciente de la IA generativa ha sido posible gracias a la combinación de tres factores clave: el aumento de la capacidad de cómputo, la disponibilidad de grandes conjuntos de datos y el avance de arquitecturas de aprendizaje profundo, especialmente los modelos basados en redes neuronales de tipo Transformer. Estos modelos aprenden patrones complejos del lenguaje, la imagen o el sonido, y posteriormente los utilizan para generar nuevas salidas a partir de instrucciones expresadas en lenguaje natural, conocidas como *prompts* [9].

Uno de los ejemplos más representativos de la IA generativa aplicada al lenguaje natural es ChatGPT, un modelo de lenguaje desarrollado por OpenAI basado en la arquitectura GPT (Generative Pre-trained Transformer). Este tipo de sistemas es capaz de mantener diálogos, responder preguntas, generar explicaciones, redactar textos académicos preliminares o crear ejercicios educativos, lo que ha impulsado su adopción en ámbitos como la educación superior y la formación continua [10]. Otros ejemplos similares incluyen Gemini (Google) y Claude (Anthropic), que persiguen objetivos equivalentes en cuanto a generación de texto y asistencia conversacional.

En el ámbito visual, la IA generativa ha dado lugar a herramientas capaces de crear imágenes originales a partir de descripciones textuales. Modelos como DALL·E, Midjourney o Stable Diffusion utilizan redes generativas entrenadas con grandes colecciones de imágenes para sintetizar ilustraciones, diagramas o material gráfico con distintos estilos. Estas aplicaciones han ampliado las posibilidades creativas en diseño, comunicación y educación, al permitir la generación rápida de recursos visuales adaptados a contextos específicos [11].

Asimismo, la IA generativa se ha extendido a la producción de música, audio y código fuente. Herramientas como Jukebox o MusicLM permiten generar composiciones musicales, mientras que sistemas como GitHub Copilot asisten en la escritura de código, sugiriendo fragmentos de programas basados en el contexto y las buenas prácticas aprendidas durante su entrenamiento. Estos ejemplos evidencian el carácter transversal de la IAG y su impacto en múltiples dominios profesionales [9].

En el contexto educativo, la inteligencia artificial generativa presenta un potencial significativo para la personalización del aprendizaje. Diversos estudios han demostrado que estas tecnologías pueden utilizarse para generar explicaciones adaptadas al nivel del estudiante, proponer ejemplos prácticos, ofrecer retroalimentación inmediata y apoyar procesos de autoaprendizaje. En educación superior, el uso controlado de sistemas como ChatGPT ha mostrado altos niveles de satisfacción estudiantil y una percepción positiva de su utilidad como apoyo al estudio, siempre que se acompañe de orientación pedagógica y pensamiento crítico [12].

No obstante, la adopción de la IA generativa también plantea desafíos relevantes. Entre ellos se encuentran la posible generación de información incorrecta, los sesgos derivados de los datos de entrenamiento, las implicaciones éticas y legales, y el riesgo de una dependencia excesiva por parte de los usuarios. Por este motivo, diversos autores coinciden en la necesidad de integrar estas tecnologías de manera responsable, promoviendo su uso como herramienta de apoyo y no como sustituto del juicio humano [13].

1.4.4 APIs

Las Interfaces de Programación de Aplicaciones, comúnmente conocidas como APIs (Application Programming Interfaces), constituyen un componente fundamental en la arquitectura del software moderno. Una API puede entenderse como un conjunto de reglas, definiciones y protocolos que permiten que distintos sistemas informáticos se comuniquen entre sí de forma estructurada y controlada. En lugar de interactuar directamente con la implementación interna de una aplicación, los sistemas consumidores utilizan la API como un intermediario que expone únicamente las funcionalidades necesarias, garantizando así un mayor nivel de abstracción y desacoplamiento [14].

Desde una perspectiva conceptual, las APIs facilitan la interoperabilidad entre aplicaciones heterogéneas, incluso cuando estas han sido desarrolladas en diferentes lenguajes de

programación o se ejecutan en plataformas distintas. Esta capacidad ha sido clave para el crecimiento de ecosistemas digitales complejos, donde servicios web, aplicaciones móviles y sistemas empresariales requieren intercambiar información de manera constante y confiable [15]. En este contexto, las APIs permiten automatizar procesos, reutilizar funcionalidades y reducir significativamente el esfuerzo de desarrollo.

Uno de los estilos más extendidos en el diseño de APIs es el enfoque REST (Representational State Transfer), el cual se apoya en el protocolo HTTP y en el uso de recursos identificados mediante URLs. Las APIs REST emplean métodos estándar como GET, POST, PUT y DELETE para realizar operaciones sobre los recursos, lo que favorece la simplicidad, la escalabilidad y la comprensión por parte de los desarrolladores [16].

En términos prácticos, las APIs se utilizan de manera cotidiana en múltiples escenarios. Por ejemplo, cuando una aplicación móvil consulta datos almacenados en un servidor, cuando una plataforma educativa recupera información de usuarios autenticados o cuando un sistema incorpora servicios externos como mapas, pagos en línea o motores de inteligencia artificial. En todos estos casos, la API actúa como un contrato que define cómo deben enviarse las solicitudes y cómo se estructuran las respuestas, generalmente en formatos como JSON o XML [17].

Desde el punto de vista estratégico, las APIs no solo cumplen una función técnica, sino que también representan un activo de valor para las organizaciones. Una API bien diseñada puede convertirse en un habilitador de innovación, al permitir que terceros desarrollen nuevas aplicaciones o servicios sobre una plataforma existente, al mismo tiempo que mejora la mantenibilidad y evolución de los sistemas [18].

1.4.5 IA y Educación en Universidades

La incorporación de la inteligencia artificial (IA), y en particular de la inteligencia artificial generativa, en la educación universitaria ha generado un profundo debate académico y pedagógico. Las universidades se enfrentan al desafío de integrar estas tecnologías en sus procesos de enseñanza y aprendizaje sin comprometer principios fundamentales como el pensamiento crítico, la autonomía intelectual y la integridad académica. En este contexto, la IA no debe entenderse únicamente como una herramienta técnica, sino como un fenómeno que transforma las dinámicas educativas y redefine el rol tanto del docente como del estudiante [19].

Diversos estudios señalan que la IA generativa puede actuar como un apoyo significativo en la educación superior, especialmente en entornos virtuales y semipresenciales. Herramientas basadas en modelos generativos permiten ofrecer retroalimentación inmediata, generar explicaciones personalizadas y apoyar la gestión del aprendizaje autónomo. Estas capacidades resultan particularmente relevantes en universidades, donde los estudiantes deben asumir un papel activo en la construcción de su conocimiento y en la autorregulación de su proceso formativo [20].

No obstante, la literatura académica advierte que el uso indiscriminado de la IA generativa puede limitar el desarrollo del pensamiento crítico si no existe una mediación pedagógica adecuada. Investigaciones recientes destacan que el verdadero valor educativo de estas tecnologías no reside en la automatización de respuestas, sino en su potencial para estimular la reflexión, el análisis y la metacognición cuando son utilizadas como herramientas de apoyo y no como sustitutos del esfuerzo intelectual del estudiante [21].

Desde la perspectiva institucional, la adopción de IA en universidades exige el desarrollo de competencias tecnopedagógicas en el profesorado, así como la definición de políticas claras sobre su uso ético y responsable. Autores como García-Peña et al. subrayan que muchas de las problemáticas asociadas a la IA en educación —como el plagio, los sesgos algorítmicos o la brecha digital— ya existían antes, pero se ven amplificadas por la potencia de las tecnologías generativas, lo que obliga a replantear estrategias de evaluación y modelos pedagógicos tradicionales [22].

En el ámbito universitario, la IA generativa también se ha explorado como asistente estratégico para el aprendizaje, apoyando tareas como la síntesis de información, la generación de ideas, la orientación académica y el acompañamiento en procesos de investigación. Estudios de casos en universidades latinoamericanas y europeas evidencian que, cuando estas herramientas se integran de forma planificada, contribuyen a mejorar la eficiencia del aprendizaje y la percepción de utilidad por parte de los estudiantes, sin sustituir el criterio académico humano [23].

1.4.6 Interacción Humano–Computador (HCI)

La Interacción Humano–Computador (Human–Computer Interaction, HCI) es un campo interdisciplinario que estudia la forma en que las personas interactúan con los sistemas computacionales, así como los principios y métodos necesarios para diseñar tecnologías

que resulten comprensibles, eficientes y acordes a las capacidades humanas. Su objetivo central no es únicamente optimizar el rendimiento técnico de los sistemas, sino asegurar que estos puedan ser entendidos y utilizados de manera significativa por los usuarios en contextos reales de uso [24].

Desde sus orígenes, la HCI ha integrado aportes de disciplinas como la informática, la psicología cognitiva, la ergonomía, el diseño y las ciencias sociales. Esta naturaleza multidisciplinaria permite analizar la interacción considerando tanto los aspectos tecnológicos como los factores humanos, tales como la percepción, la memoria, la carga cognitiva y el contexto sociocultural en el que se produce la interacción. En este sentido, la HCI se orienta a reducir la brecha entre la lógica de funcionamiento de los sistemas y los modelos mentales de los usuarios [25].

Un concepto clave dentro de la HCI es el de comunicabilidad, propuesto por la Ingeniería Semiótica, el cual plantea que la interfaz actúa como un medio de comunicación entre el diseñador y el usuario. A través de la interfaz, el diseñador transmite sus intenciones, supuestos y decisiones de diseño, mientras que el usuario interpreta estos mensajes durante la interacción. Cuando esta comunicación es clara y coherente, el usuario puede comprender qué ofrece el sistema, cómo utilizarlo y cuáles son sus limitaciones; en caso contrario, se generan rupturas en la interacción y experiencias negativas de uso [26].

La HCI también se relaciona estrechamente con conceptos como usabilidad y experiencia de usuario. Mientras la usabilidad se enfoca en la eficacia, eficiencia y satisfacción con la que un usuario puede alcanzar sus objetivos, la HCI ofrece un marco más amplio para analizar cómo se construye la interacción a lo largo del tiempo y cómo influyen los factores cognitivos, emocionales y contextuales en dicha interacción. De esta manera, la HCI proporciona los fundamentos teóricos y metodológicos que sustentan el diseño centrado en el usuario [2].

En el ámbito educativo, la interacción humano–computador adquiere especial relevancia, ya que las plataformas digitales median gran parte de los procesos de enseñanza y aprendizaje. Un diseño deficiente de la interacción puede generar confusión, sobrecarga cognitiva o desmotivación en los estudiantes. Por el contrario, una interacción bien diseñada facilita la comprensión de los contenidos, promueve la exploración autónoma y favorece un aprendizaje más activo y significativo [27].

En el contexto de sistemas educativos apoyados por inteligencia artificial, la HCI permi-

te analizar cómo los usuarios interpretan las recomendaciones, respuestas y adaptaciones generadas por el sistema. Una interacción transparente y comunicativa contribuye a generar confianza en la tecnología y a evitar una dependencia acrítica de las respuestas automatizadas. Por ello, la HCI se consolida como un eje fundamental para el diseño de soluciones educativas que integren IA de manera responsable y centrada en las personas.

2 METODOLOGÍA

Este sección describe la forma en que se organizará y ejecutará el desarrollo del componente, con el objetivo de que el proceso sea claro y pueda replicarse. La metodología se plantea como una guía para pasar desde la definición de necesidades y requerimientos hasta la construcción de prototipos, la implementación de una interfaz funcional y su validación con usuarios reales.

Dado que el componente implica decisiones de diseño que se refinan conforme se obtiene retroalimentación (por ejemplo, ajustes en navegación, presentación del contenido y flujo de interacción), se trabajará con un enfoque incremental. Por esta razón, el desarrollo se estructura en ciclos cortos que permiten planificar, ejecutar, revisar y mejorar entregables de manera progresiva, manteniendo control del avance y facilitando correcciones oportunas conforme se identifiquen nuevas necesidades.

En las siguientes secciones se detalla el marco de trabajo seleccionado (SCRUM), la organización del trabajo por sprints, las actividades realizadas en cada etapa, así como las técnicas consideradas para recopilar información y evaluar los resultados del componente.

2.1 Metodología SCRUM

Para desarrollar el componente de Diseño de la Interfaz y Experiencia de Usuario (UX/UI) se empleó el marco de trabajo ágil SCRUM, ya que permite avanzar por iteraciones cortas, revisar resultados con frecuencia y ajustar el producto a medida que se obtiene nueva retroalimentación. En otras palabras, en lugar de intentar “acertar” a la primera, se trabaja por incrementos: se planifica, se construye, se revisa lo logrado y se mejora en el siguiente ciclo. Este enfoque encaja bien con UX/UI porque el diseño se fortalece cuando se valida con usuarios y se refina de forma continua [28].

De acuerdo con Scrum.org, SCRUM se organiza mediante un conjunto de roles, eventos y artefactos que buscan asegurar transparencia del trabajo, inspección frecuente del avance y adaptación oportuna ante cambios [28]. En este componente, dichos elementos se aplicaron para ordenar la planificación semanal, mantener claridad sobre lo que se estaba construyendo y asegurar entregables verificables al final de cada sprint.

A continuación, se presenta en la Figura 2.1 el framework general de SCRUM

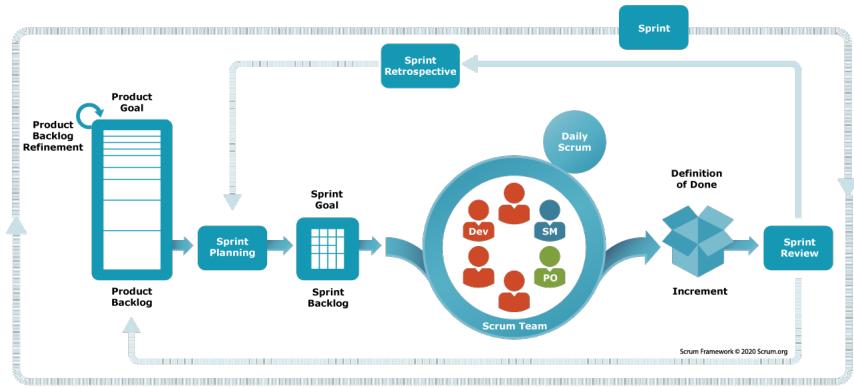


Figura 2.1: Framework SCRUM (basado en Scrum.org) [28]

En las tablas siguientes, se presentan los elementos principales (equipo, eventos y artefactos) con su interpretación aplicada al componente UX/UI. Todas las definiciones y denominaciones empleadas se basan en la descripción de Scrum.org [28].

Tabla 2.1: Equipo SCRUM (basado en Scrum.org) [28]

Rol	Responsabilidad principal	Aplicación en el componente UX/UI
Product Owner	Maximiza el valor del producto y gestiona el Product Backlog.	Prioriza necesidades de la interfaz (funcionalidades, pantallas, flujos) y valida que el incremento responda al objetivo del simulador.
Scrum Master	Asegura que Scrum se entienda y se aplique; elimina impedimentos.	Facilita la dinámica de sprints, cuida que se cumplan eventos y ayuda a resolver bloqueos (dependencias técnicas, acceso a usuarios para pruebas, etc.).
Developers	Construyen el incremento y son responsables de convertir backlog en resultados.	Diseñan prototipos, implementan la UI y realizan ajustes; documentan decisiones y preparan el incremento para revisión y pruebas.

Tabla 2.2: Eventos SCRUM (basado en Scrum.org) [28]

Evento	Propósito	Resultado esperado	Cómo se aplicó al componente UX/UI
Sprint	Iteración fija donde se construye un incremento.	Incremento potencialmente utilizable.	Sprints semanales para entregar avances: prototipos, pantallas implementadas, integración y mejoras.
Sprint Planning	Definir objetivo del sprint y seleccionar trabajo del backlog.	Sprint Goal + Sprint Backlog.	Selección de historias/tareas UX/UI (pantallas, flujos, criterios de usabilidad) y definición del objetivo semanal.
Daily Scrum	Inspección diaria del avance y ajuste del plan.	Plan actualizado para alcanzar el Sprint Goal.	Seguimiento breve de tareas (por ejemplo: integración con módulos, cambios de flujo, ajustes de accesibilidad).
Sprint Review	Revisión del incremento con interesados y ajuste del backlog.	Retroalimentación y backlog refinado.	Demostración de la interfaz o prototipo; registro de mejoras de navegación, claridad visual y necesidades detectadas.
Sprint Retrospective	Mejorar el proceso del equipo.	Acciones de mejora para el siguiente sprint.	Lecciones aprendidas (tiempos, comunicación, criterios de diseño) y ajustes en la forma de trabajo.

Tabla 2.3: Artefactos SCRUM (basado en Scrum.org) [28]

Artefacto	Qué representa	Contenido típico	Uso en el componente UX/UI
Product Backlog	Lista ordenada de todo lo necesario para el producto.	Historias, requisitos, mejoras, correcciones.	Pantallas del simulador, flujos, criterios de usabilidad, componentes UI, ajustes por pruebas.
Sprint Backlog	Trabajo seleccionado para el sprint + plan para completarlo.	Tareas del sprint.	Wireframes, prototipos Hi-Fi, implementación de vistas, integración con módulos y correcciones priorizadas.
Incremento	Resultado integrado del trabajo del sprint.	Funcionalidad terminada y verificable.	Versión de la interfaz lista para revisión y, cuando aplique, para pruebas con estudiantes.

Tabla 2.4: Pilares de SCRUM aplicados al componente (basado en Scrum.org) [28]

Principio	Sentido en SCRUM	Aplicación práctica en UX/UI
Transparencia	El trabajo y su estado deben ser visibles para el equipo y partes interesadas.	Backlogs claros y entregables revisables (prototipos, pantallas, criterios de diseño).
Inspección	Revisión frecuente para detectar desviaciones.	Revisiones por sprint y pruebas de usabilidad para identificar fricción en la interfaz.
Adaptación	Ajuste del plan cuando la inspección revela oportunidades o problemas.	Iteraciones de diseño: cambios en navegación, jerarquía visual y mensajes según evidencia.

2.1.1 Definición y asignación de roles

El desarrollo del componente UX/UI se organizó bajo el marco de trabajo SCRUM, el cual establece roles claramente definidos para asegurar una gestión ordenada del proyecto y una

adecuada coordinación de las actividades. La asignación de estos roles permitió estructurar el trabajo desde la planificación hasta la entrega del componente, garantizando coherencia entre los objetivos planteados y los resultados obtenidos [28].

Cada rol asumió responsabilidades específicas dentro del proceso, facilitando la toma de decisiones, el seguimiento del avance y la generación de incrementos funcionales del componente. La Tabla 2.5 presenta los roles definidos y los responsables asignados para el desarrollo del componente UX/UI.

Tabla 2.5: Roles y responsables del equipo SCRUM

Rol	Responsable
Product Owner	Larco Ampudia Enrique Andrés, PhD.
Scrum Master	Larco Ampudia Enrique Andrés, PhD.
Equipo de desarrollo	Hernán Darío Sánchez Tenelanda Carlos Andrés Córdova Acaro Michael Andrés Pillaga Sosa Carla Annai Ruiz Acevedo

2.1.2 Planificación y organización de los sprints

La planificación del desarrollo del componente UX/UI se realizó mediante *sprints*, entendidos como períodos de tiempo definidos en los que se ejecuta un conjunto de tareas orientadas a la obtención de un incremento del producto. Esta organización permitió dividir el trabajo en etapas manejables y facilitar el control del avance del proyecto [28].

Cada sprint contó con tareas previamente definidas y una duración establecida, lo que permitió revisar los resultados obtenidos al final de cada ciclo y realizar ajustes oportunos en función de los avances y necesidades identificadas. La Tabla 2.6 presenta la planificación de los sprints definidos para el desarrollo del componente UX/UI.

Tabla 2.6: Planificación de sprints del componente UX/UI

Sprint	Tareas	Duración
Sprint 0	Revisión del documento TIC y definición de objetivos de diseño UX/UI.	20 horas
Sprint 1	Análisis de herramientas de IA generativa y levantamiento de requerimientos.	20 horas
Sprint 2	Diseño de prototipos de baja fidelidad y estructura de navegación.	20 horas
Sprint 3	Desarrollo de prototipos de alta fidelidad y ajustes iniciales.	20 horas
Sprint 4	Implementación inicial de la interfaz web.	20 horas
Sprint 5	Integración de módulos funcionales del simulador.	20 horas
Sprint 6	Pruebas de usabilidad con estudiantes y mejoras iterativas.	20 horas
Sprint 7	Documentación técnica del componente y entrega final.	20 horas

2.2 Tecnologías Utilizadas

El desarrollo del componente de Diseño de la Interfaz y Experiencia de Usuario (UX/UI) del simulador educativo se apoyó en tecnologías web ampliamente utilizadas en el desarrollo de interfaces interactivas. Estas tecnologías fueron seleccionadas por su estabilidad, documentación disponible y compatibilidad con arquitecturas modernas basadas en servicios. En esta sección se presentan los lenguajes, frameworks, plataformas y herramientas empleadas, incluyendo una breve descripción de su función dentro del desarrollo del componente. La información se organiza en tablas con el fin de facilitar su comprensión y replicabilidad.

Tabla 2.7: Lenguajes y frameworks utilizados

Tecnología	Tipo	Descripción
HTML5	Lenguaje de marcado	Lenguaje estándar para estructurar el contenido de aplicaciones web [29].
CSS3	Lenguaje de estilos	Lenguaje utilizado para definir la presentación visual y el diseño responsivo de interfaces web [30].
JavaScript	Lenguaje de programación	Lenguaje de programación orientado a la interacción y lógica del lado del cliente [31].
React	Biblioteca / framework	Biblioteca de JavaScript para la construcción de interfaces basadas en componentes reutilizables [32].

Tabla 2.8: Plataformas y APIs consideradas

Plataforma	Tipo	Descripción
APIs REST	Interfaz de programación	Estilo de arquitectura que permite la comunicación entre sistemas mediante servicios web [17].
Node.js	Plataforma de ejecución	Entorno de ejecución que permite ejecutar JavaScript del lado del servidor [33].

2.2.1 Herramientas Utilizadas

Para apoyar las actividades de diseño y desarrollo del componente UX/UI se utilizaron herramientas que facilitaron el trabajo colaborativo, la implementación del código y la gestión del proyecto.

Tabla 2.9: Herramientas utilizadas durante el desarrollo

Herramienta	Tipo	Descripción
Figma	Diseño y prototipado	Herramienta colaborativa para el diseño de interfaces y prototipos interactivos [34].
Visual Studio Code	Editor de código	Editor de código fuente multiplataforma ampliamente utilizado en desarrollo web [35].
NPM	Gestor de paquetes	Administrador de dependencias para proyectos basados en JavaScript [36].
Git	Control de versiones	Sistema de control de versiones distribuido para el seguimiento de cambios en el código [37].

2.3 Etapa de Desarrollo

2.3.1 Sprint 0: Análisis inicial y definición del enfoque UX/UI

El Sprint 0 se planteó como una etapa estratégica previa al desarrollo del componente UX/UI, orientada a reducir la incertidumbre inicial y establecer criterios claros antes de iniciar actividades de diseño o implementación. Aunque el proyecto contaba con objetivos generales definidos en el documento TIC, fue necesario analizar estos lineamientos desde una perspectiva de experiencia de usuario, con el fin de traducirlos en decisiones concretas de diseño y alcance funcional. Este enfoque permitió identificar tempranamente qué aspectos del simulador requerían mayor énfasis en la interacción y cómo la interfaz podía apoyar de forma efectiva el proceso de aprendizaje del usuario.

Durante este sprint se realizó un análisis crítico del documento TIC general, evaluando la coherencia entre los objetivos específicos del componente y las funcionalidades esperadas del simulador. A partir de este análisis se definió el enfoque UX/UI, delimitando el alcance del componente y estableciendo prioridades iniciales. Esta actividad permitió evitar supuestos implícitos sobre la interfaz y generar una comprensión compartida de lo que debía resolverse desde el diseño, especialmente en términos de navegación, organización del contenido y presentación de la información generada por la inteligencia artificial.

Como resultado del Sprint 0, se formularon objetivos de diseño UX/UI alineados con las necesidades del usuario final **estudiantes universitarios** y con el contexto académico del proyecto. Asimismo, se establecieron criterios iniciales de usabilidad relacionados con la claridad de los flujos, la consistencia visual y la retroalimentación durante la interacción. Estos criterios funcionaron como marco de referencia para evaluar las decisiones tomadas en los sprints posteriores.

2.3.2 Sprint 1: Análisis de herramientas de IA generativa y levantamiento de requerimientos

2.3.2.1 Análisis de herramientas de IA generativa en contextos educativos

En el marco del Sprint 1 se realizó un análisis conceptual de distintas herramientas y enfoques de inteligencia artificial generativa descritos en la literatura académica, con el objetivo de identificar criterios relevantes para el diseño de la interfaz del simulador educativo.

Este análisis no se enfocó en aspectos técnicos de implementación, sino en la forma en que dichas herramientas presentan información, estructuran los resultados y acompañan al usuario durante el proceso de aprendizaje, elementos clave desde la perspectiva de la experiencia de usuario.

Diversos autores coinciden en que la inteligencia artificial generativa se caracteriza por su capacidad para producir contenidos nuevos como textos, explicaciones o ejemplos a partir de patrones aprendidos de grandes volúmenes de datos [8], [9]. En contextos educativos, esta capacidad ha sido utilizada principalmente como apoyo al aprendizaje autónomo, la generación de explicaciones personalizadas y la simulación de escenarios de práctica, especialmente en educación superior [12], [19]. Sin embargo, la literatura enfatiza que el valor educativo de estas herramientas no depende únicamente de la calidad del contenido generado, sino de cómo este es presentado y contextualizado para el estudiante.

Estudios recientes señalan que muchas herramientas de IA generativa orientadas a la educación destacan por integrar interfaces que priorizan la claridad visual, la segmentación de la información y la retroalimentación inmediata [10], [13]. Estas características permiten que el usuario comprenda no solo el resultado generado, sino también el razonamiento asociado, lo cual resulta fundamental para fomentar el pensamiento crítico y evitar una interacción pasiva con la tecnología. En este sentido, la explicabilidad y la transparencia de los resultados se convierten en factores determinantes para su aceptación y uso efectivo en entornos universitarios.

Desde la perspectiva del diseño centrado en el usuario, la literatura sobre experiencia de usuario resalta que los sistemas basados en inteligencia artificial deben minimizar la carga cognitiva y facilitar la interpretación de la información presentada [1], [2]. Esto implica interfaces que organicen los resultados de manera jerárquica, utilicen un lenguaje comprensible y ofrezcan apoyos visuales o textuales que guíen al estudiante en la lectura y análisis de los contenidos generados. Estas recomendaciones coinciden con los principios establecidos por la norma ISO 9241-210, que subraya la importancia de adaptar los sistemas interactivos a las capacidades y necesidades del usuario final.

A partir de este análisis, se identificó que las herramientas de IA generativa utilizadas en educación comparten patrones comunes que pueden servir como referencia para el diseño del componente UX/UI del simulador. Entre estos patrones se destacan la presentación progresiva de la información, la disponibilidad de explicaciones complementarias, la retroalimentación clara ante las acciones del usuario y la consistencia visual a lo largo de la interacción. Estos elementos no implican la adopción directa de herramientas existentes,

sino que funcionan como insumos conceptuales para definir cómo los resultados generados por otros componentes del sistema deben ser mostrados e interpretados dentro de la interfaz.

En consecuencia, el análisis de herramientas de IA generativa permitió establecer una base conceptual sólida para el levantamiento de requerimientos del componente UX/UI, asegurando que el diseño de la interfaz esté alineado tanto con las buenas prácticas identificadas en la literatura como con las necesidades específicas del contexto educativo universitario [21], [22].

2.3.2.2 Requerimientos funcionales

A partir del análisis de herramientas de inteligencia artificial generativa y de su aplicación en contextos educativos universitarios, se identificaron los requerimientos funcionales del componente UX/UI del simulador. Estos requerimientos describen las funcionalidades que la interfaz debe ofrecer para permitir una interacción efectiva entre el estudiante y los contenidos generados por el sistema, garantizando claridad, comprensión y apoyo al proceso de aprendizaje.

A continuación, en la Tabla 2.10, se presentan los requerimientos funcionales definidos para este componente.

Tabla 2.10: Requerimientos funcionales del componente UX/UI

ID	Descripción del requisito
RF-01	El sistema debe permitir la visualización de los resultados generados por el simulador de forma clara y estructurada.
RF-02	El sistema debe mostrar explicaciones asociadas a los resultados presentados para facilitar su comprensión.
RF-03	El sistema debe permitir la presentación de ejemplos complementarios que refuerzen los conceptos evaluados.
RF-04	El sistema debe permitir la navegación fluida entre los distintos módulos del simulador sin pérdida de contexto.
RF-05	El sistema debe proporcionar retroalimentación visual inmediata ante las acciones realizadas por el usuario.
RF-06	El sistema debe permitir la organización progresiva de la información cuando los resultados sean extensos.
RF-07	El sistema debe mostrar mensajes claros y orientativos cuando se detecten errores o acciones incorrectas del usuario.
RF-08	El sistema debe permitir al usuario identificar claramente las secciones y estados de la interfaz durante la interacción.

2.3.2.3 Requerimientos no funcionales

Además de las funcionalidades que debe ofrecer la interfaz, es necesario definir un conjunto de requerimientos no funcionales que establezcan las condiciones de calidad bajo las cuales el componente UX/UI debe operar. Estos requerimientos no describen acciones específicas del sistema, sino atributos relacionados con la usabilidad, claridad, consistencia y accesibilidad de la interfaz, los cuales son fundamentales para garantizar una experiencia de usuario adecuada en contextos educativos universitarios.

A continuación, en la Tabla 2.11, se presentan los requerimientos no funcionales definidos para este componente.

Tabla 2.11: Requerimientos no funcionales del componente UX/UI

ID	Descripción del requisito
RNF-01	La interfaz debe presentar la información con una jerarquía visual clara que facilite su comprensión.
RNF-02	El diseño visual de la interfaz debe ser consistente en todas las pantallas del simulador.
RNF-03	La interfaz debe minimizar la carga cognitiva del usuario mediante una presentación ordenada de la información.
RNF-04	Los textos y mensajes mostrados en la interfaz deben utilizar un lenguaje claro y comprensible para estudiantes universitarios.
RNF-05	La interfaz debe ofrecer retroalimentación comprensible y oportuna ante las acciones realizadas por el usuario.
RNF-06	La interfaz debe permitir una navegación intuitiva, reduciendo la necesidad de aprendizaje previo.
RNF-07	La interfaz debe ser accesible para usuarios con distintos niveles de experiencia tecnológica.
RNF-08	La interfaz debe mantener coherencia visual y funcional al presentar los resultados generados por el simulador.

2.3.2.4 Historias de usuario

Con base en los requerimientos funcionales y no funcionales definidos previamente, se elaboraron las historias de usuario del componente UX/UI del simulador. Las historias de usuario permiten describir las necesidades del sistema desde la perspectiva del usuario final, facilitando la planificación del desarrollo y asegurando la trazabilidad entre los objetivos del diseño y las funcionalidades a implementar.

Para mantener uniformidad en su redacción y evaluación, se adoptó el formato que se presenta a continuación en la Tabla 2.12.

Tabla 2.12: Formato de historias de usuario

Código	Título	Descripción	Criterio de aceptación	Prioridad
HU-XX	Nombre descriptivo	Como [usuario], quiero [objetivo] para [beneficio].	Dado..., cuando..., entonces...	Alta / Media / Baja

A partir de este formato, se definieron las historias de usuario del componente UX/UI, las cuales se presentan a continuación en la Tabla 2.13.

Estas historias reflejan los principales escenarios de interacción del estudiante con la interfaz y sirven como base para la planificación de los sprints posteriores, en coherencia con la metodología SCRUM.

Tabla 2.13: Historias de usuario del componente UX/UI

Código	Título	Descripción	Criterio de aceptación	Prioridad
HU-01	Visualización de resultados	Como estudiante, quiero visualizar los resultados del simulador de forma clara para comprender mi desempeño.	Dado que completo una actividad, cuando accedo a los resultados, entonces estos se muestran de forma estructurada.	Alta
HU-02	Consulta de explicaciones	Como estudiante, quiero acceder a explicaciones asociadas a los resultados para entenderlos mejor.	Dado que visualizo un resultado, cuando solicito una explicación, entonces el sistema la presenta de forma clara.	Alta
HU-03	Acceso a ejemplos	Como estudiante, quiero ver ejemplos complementarios para reforzar mi comprensión.	Dado que reviso una explicación, cuando solicito ejemplos, entonces se muestran casos ilustrativos.	Media
HU-04	Navegación entre módulos	Como estudiante, quiero navegar entre módulos sin perder el contexto de aprendizaje.	Dado que cambio de módulo, cuando navego por la interfaz, entonces mantengo el contexto previo.	Alta
HU-05	Retroalimentación visual	Como estudiante, quiero recibir retroalimentación visual sobre mis acciones para saber si fueron correctas.	Dado que realizo una acción, cuando el sistema la procesa, entonces recibo una respuesta visual comprensible.	Media
HU-06	Organización progresiva	Como estudiante, quiero que la información extensa se muestre de forma progresiva para no saturarme.	Dado que reviso resultados extensos, cuando avanco en la lectura, entonces la información se presenta por secciones.	Media
HU-07	Identificación de secciones	Como estudiante, quiero identificar fácilmente las secciones de la interfaz para orientarme mejor.	Dado que ingreso a una pantalla, cuando observo la interfaz, entonces reconozco claramente sus secciones.	Alta
HU-08	Mensajes de error	Como estudiante, quiero recibir mensajes claros cuando cometo errores para saber cómo corregirlos.	Dado que ocurre un error, cuando el sistema lo detecta, entonces muestra un mensaje orientativo.	Media
HU-09	Consistencia visual	Como estudiante, quiero que la interfaz mantenga coherencia visual en todas las pantallas.	Dado que navego entre pantallas, cuando cambio de vista, entonces se mantiene el mismo estilo visual.	Media
HU-10	Comprensión del flujo	Como estudiante, quiero entender el flujo general del simulador desde el primer uso.	Dado que utilizo el sistema por primera vez, cuando completo una tarea, entonces comprendo el flujo general.	Alta
HU-11	Claridad del lenguaje	Como estudiante, quiero que los textos sean claros para interpretar correctamente la información.	Dado que leo un mensaje, cuando este se presenta, entonces utiliza un lenguaje comprensible.	Media
HU-12	Facilidad de uso	Como estudiante, quiero que la interfaz sea fácil de usar sin necesidad de capacitación previa.	Dado que interactúo con la interfaz, cuando realizo acciones básicas, entonces puedo hacerlo sin dificultad.	Alta

2.3.3 Sprint 2: Diseño de prototipos de baja fidelidad y estructura de navegación

El Sprint 2 se enfocó en el diseño de prototipos de baja fidelidad y en la definición detallada de la estructura de navegación del Simulador de Clases Personalizadas con IA. Esta etapa resulta fundamental dentro del proceso de desarrollo, ya que permite transformar los requerimientos identificados en el Sprint 1 en un flujo de interacción concreto, comprensible y validable desde el punto de vista del usuario, antes de abordar decisiones visuales o de implementación más complejas.

La necesidad de describir y modelar el flujo de navegación en esta fase responde a que la interacción con sistemas educativos basados en inteligencia artificial puede generar incertidumbre en los usuarios si el proceso no es claro desde el inicio. En particular, cuando el sistema realiza procesos internos que no son visibles como la generación de contenidos personalizados, la interfaz se convierte en el principal medio para comunicar al usuario qué está ocurriendo, en qué etapa se encuentra y qué se espera de él en cada momento. Por ello, definir explícitamente el recorrido entre pantallas no solo facilita el uso del sistema, sino que contribuye a generar confianza y comprensión del funcionamiento general del simulador.

Desde una perspectiva centrada en el estudiante, el diseño del flujo de navegación se planteó como un acompañamiento progresivo a lo largo del proceso de simulación. En lugar de presentar múltiples opciones de forma simultánea, se optó por una secuencia controlada de pantallas que guía al usuario paso a paso, reduciendo la carga cognitiva y evitando decisiones innecesarias. Esta estrategia resulta especialmente relevante en contextos educativos, donde el objetivo principal no es la exploración libre del sistema, sino el apoyo efectivo al proceso de aprendizaje.

Para materializar este análisis, los prototipos de baja fidelidad fueron desarrollados utilizando la herramienta Figma, la cual permitió representar de forma visual y estructurada la disposición de los elementos,

Pantalla de inicio de sesión:

La pantalla de inicio de sesión constituye el primer punto de interacción del usuario con el simulador y cumple un rol clave en la contextualización del uso académico de la plataforma. Su objetivo principal es permitir el acceso al sistema mediante credenciales institucionales

simuladas, estableciendo desde el inicio un marco de uso orientado al ámbito universitario. Desde el punto de vista del diseño, esta pantalla fue concebida para ser clara y directa, priorizando la comprensión inmediata de su función. Se limitaron los elementos visuales a los estrictamente necesarios, como los campos de ingreso y el botón de acceso, con el fin de evitar distracciones y reducir la carga cognitiva en esta etapa inicial. Esta decisión busca que el usuario complete el acceso al sistema de forma rápida y sin ambigüedades.

Asimismo, la pantalla de inicio de sesión permite identificar el rol del usuario dentro del simulador, lo cual resulta relevante para contextualizar el recorrido posterior. Aunque el enfoque principal del componente es el estudiante, esta identificación inicial refuerza la coherencia del flujo de navegación y prepara al sistema para adaptar la experiencia de acuerdo con el perfil definido.

En la Figura 2.2, se presenta el prototipo de baja fidelidad de la pantalla de inicio de sesión, desarrollado en Figma. Esta representación permitió validar la disposición de los campos, la jerarquía visual de los elementos y la claridad del mensaje transmitido al usuario, antes de avanzar hacia prototipos de mayor fidelidad.

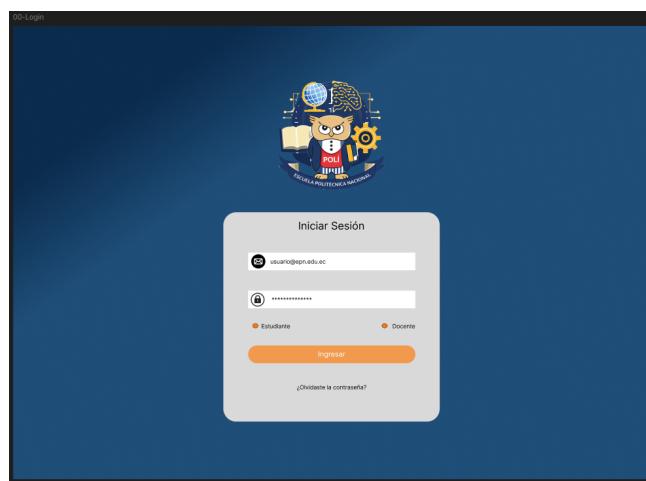


Figura 2.2: Pantalla de inicio de sesión del simulador

3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

3.1 Resultados

Exponer los resultados obtenidos, utilizando para esto el apoyo de tablas, figuras, entre otros.

Un ejemplo de una tabla se presenta en Tabla 3.1.

Tabla 3.1: Ejemplo tabla y alineación de texto.

Texto Centrado	Texto Centrado	Texto Centrado
Los comandos en esta celda también permiten centrar el texto.	Este texto aparece justificado.	Este texto aparece centrado.
Texto centrado para 3 filas	1 2 3	0,00 0,00 0,00
Texto centrado para 3 columnas		
Ejemplo de nota para una tabla ^[1]	Ejemplo Imagen 	PRUEBA

Un ejemplo de una figura se presenta en Figura 3.1

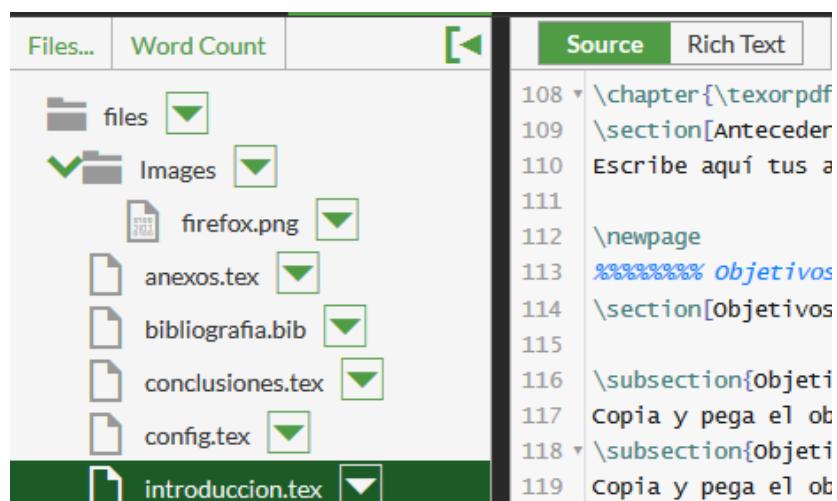


Figura 3.1: Ejemplo imagen

3.2 Conclusiones

Presenta lo novedoso del trabajo de integración curricular, así como evaluación del cumplimiento o no de lo propuesto en los objetivos. En el caso en que no se cumpla uno o varios objetivos, y no se logren los resultados esperados, se propone una posible respuesta que explique por qué sucedió esto o las falencias de la planteado.

3.3 Recomendaciones

Indicar las recomendaciones formuladas a partir del desarrollo de este trabajo de integración curricular.

4 REFERENCIAS BIBLIOGRÁFICAS

- [1] Y. Hassan Montero, *Experiencia de Usuario: Principios y Métodos*. 2015.
- [2] *ISO 9241-210:2019 Ergonomics of Human-System Interaction – Human-centred design for interactive systems*, International Organization for Standardization, 2019.
- [3] P. B. Vidal y A. Martin, «Experiencia de Usuario y Diseño Web Responsivo: Un enfoque integrado,» *Revista ICT-UNPA*, vol. 12, n.º 1, 2019.
- [4] L. S. M. Gómez, *Diseño de Interfaces de Usuario: Principios, Prototipos y Heurísticas para Evaluación*. 2012.
- [5] M. C. Albornoz, «Diseño de Interfaz Gráfica de Usuario,» en *WICC*, 2014.
- [6] S. Morejón Labrada, «Principios del proceso de Diseño de Interfaz de Usuario,» *Revista Cubana de Transformación Digital*, vol. 1, n.º 3, 2020.
- [7] M. C. Albornoz, M. Berón y G. Montejano, «Interfaz Gráfica de Usuario: el usuario como protagonista del diseño,» Universidad Nacional de San Luis, inf. téc., 2000.
- [8] J. R. Casar Corredera, «Inteligencia artificial generativa,» *Anales de la Real Academia de Doctores de España*, vol. 8, n.º 3, págs. 475-489, 2023.
- [9] K. M. Gutiérrez López, «Inteligencia artificial generativa: irrupción y desafíos,» *Revista Enfoques*, 2023.
- [10] L. Finkel et al., «ChatGPT como fuente de información en la Educación Superior,» *Profesional de la información*, vol. 33, n.º 6, 2024.
- [11] J. Franganillo, «La inteligencia artificial generativa y su impacto en la creación de contenidos mediáticos,» *methaodos.revista de ciencias sociales*, vol. 11, n.º 2, 2023.
- [12] J. C. Sánchez-Prieto et al., «Inteligencia artificial generativa para autoaprendizaje en educación superior,» *RIED*, vol. 28, n.º 1, págs. 59-81, 2025.
- [13] L. Finkel et al., «Valoración crítica de la IA generativa en contextos educativos,» *Profesional de la información*, 2024.
- [14] B. Cooksey, *An Introduction to APIs*. Zapier Inc., 2014.
- [15] D. Jacobson, G. Brail y D. Woods, *APIs: A Strategy Guide*. O'Reilly Media, 2012.
- [16] E. Amodeo, *Principios de diseño de APIs REST*. Leanpub, 2013.
- [17] R. T. Fielding, «Architectural Styles and the Design of Network-based Software Architectures,» Tesis doct., University of California, Irvine, 2000.

- [18] D. Jacobson, G. Brail y D. Woods, «The API Opportunity,» en *APIs: A Strategy Guide*, O'Reilly Media, 2012.
- [19] F. J. García-Peña, F. Llorens-Largo y J. Vidal, «La nueva realidad de la educación ante los avances de la inteligencia artificial generativa,» *RIED-Revista Iberoamericana de Educación a Distancia*, vol. 27, n.º 1, 2024.
- [20] A. Cortés Hernández et al., «La inteligencia artificial generativa como un asistente estratégico en la era del aprendizaje digital,» *Ciencia Latina Revista Científica Multidisciplinar*, vol. 8, n.º 4, 2024.
- [21] C. Muñoz Martínez, V. Roger-Monzo y F. Castelló-Sirvent, «IA generativa y pensamiento crítico en la educación universitaria,» *RIED-Revista Iberoamericana de Educación a Distancia*, vol. 28, n.º 2, 2025.
- [22] F. J. García-Peña et al., «Artificial intelligence in education: challenges and opportunities,» *RIED*, 2024.
- [23] M. Zapata Ros, «IA generativa y ChatGPT en educación: un reto para la evaluación y una nueva pedagogía,» *Revista Paraguaya de Educación a Distancia*, vol. 5, n.º 1, 2024.
- [24] G. M. Martínez de la Teja, «Ergonomía e interfases de interacción humano-computadora,» en *IX Congreso Internacional de Ergonomía*, México, 2007.
- [25] Y. Rogers, «New theoretical approaches for HCI,» *Annual Review of Information Science and Technology*, vol. 38, 2004.
- [26] L. Correa Alfaro, «Comunicabilidad, paradigma de la Interacción Humano-Computador,» *No Solo Usabilidad*, n.º 9, 2010, ISSN: 1886-8592.
- [27] A. K. Alcívar-Cedeño et al., «Interacción humano-computador en el metaverso educativo,» *Revista PENTACIENCIAS*, vol. 5, n.º 2, 2023.
- [28] Scrum.org. «What is Scrum?» Visitado 14 de dic. de 2025. dirección: <https://www.scrum.org/learning-series/what-is-scrum>
- [29] W3C. «HTML5 Specification,» visitado 14 de dic. de 2025. dirección: <https://www.w3.org/TR/html52/>
- [30] W3C. «CSS Snapshot,» visitado 14 de dic. de 2025. dirección: <https://www.w3.org/TR/CSS/>

- [31] Mozilla Developer Network. «JavaScript Guide,» visitado 14 de dic. de 2025. dirección: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- [32] Meta Platforms Inc. «React Documentation,» visitado 14 de dic. de 2025. dirección: <https://react.dev/>
- [33] OpenJS Foundation. «Node.js Documentation,» visitado 14 de dic. de 2025. dirección: <https://nodejs.org/en/docs/>
- [34] Figma Inc. «Figma Help Center,» visitado 14 de dic. de 2025. dirección: <https://help.figma.com/>
- [35] Microsoft. «Visual Studio Code Documentation,» visitado 14 de dic. de 2025. dirección: <https://code.visualstudio.com/docs>
- [36] npm Inc. «npm Documentation,» visitado 14 de dic. de 2025. dirección: <https://docs.npmjs.com/>
- [37] Git SCM. «Git Documentation,» visitado 14 de dic. de 2025. dirección: <https://git-scm.com/doc>

5 ANEXOS

En caso necesario, el documento escrito deberá incluir los anexos y secciones que incorporan información que sea relevante, pero que, por su extensión, no pueden ser incorporadas directamente en ninguna de las secciones anteriores. Normalmente, en la sección de Anexos se incluyen conjuntos de datos extensos, formatos de encuestas, entrevistas, enlaces hacia videos o programas que sean producto o formen parte del Trabajo de Integración Curricular, entre otros.

Ejemplo de Anexos se muestran a continuación:

ANEXO I. Conjunto de Datos Extensos

ANEXO II. Formato de Entrevista

ANEXO III. Enlaces

La numeración de los Anexos debe realizarse con números en formato romano.

ANEXO I: APLICACIÓN MOVIL

ANEXO II: APLICACIÓN WEB

ANEXO III: APLICACIÓN PÁGINA WEB