

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, enero 2026

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)	4
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
2.3.4	Sprint 3: Diseño de prototipos de alta fidelidad	26
2.3.5	Sprint 4: Implementación inicial de la interfaz	35
3	RESULTADOS, CONCLUSIONES Y RECOMENDACIONES	51
3.1	Resultados	51
3.2	Conclusiones	52
3.3	Recomendaciones	52
4	REFERENCIAS BIBLIOGRÁFICAS	53
5	ANEXOS	54
	ANEXO I: Aplicación Móvil	54
	ANEXO II: Aplicación Web	55
	ANEXO III: Página Web	56

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 el inicio del proyecto se entendió que la interfaz no debía limitarse únicamente a un diseño visual atractivo. Su función principal era actuar como puente entre el alumno y el sistema, facilitando la interacción y el acceso a los contenidos sin provocar confusiones ni dificultades innecesarias.

Con este enfoque se inició una fase de análisis en la que se revisaron diversas plataformas y herramientas de formación basadas en inteligencia artificial generativa. El objetivo era identificar prácticas efectivas y elementos que pudieran adaptarse al contexto del simulador.

A partir de este análisis se definieron las principales funcionalidades de la plataforma y se desarrollaron prototipos utilizando un enfoque centrado en el usuario. Estos diseños iniciales priorizaron una navegación sencilla y una estructura organizada que permitiera a los estudiantes acceder rápidamente a las secciones importantes: lecciones personalizadas, comentarios instantáneos y ejercicios adaptativos.

Se prestó especial atención a aspectos clave como la legibilidad del texto, el contraste de color, el uso adecuado de los iconos y la presentación clara de la información, garantizando así una experiencia de usuario coherente y comprensible.

Durante el proceso de desarrollo se realizaron pruebas de usabilidad con estudiantes universitarios, las cuales fueron fundamentales para evaluar la funcionalidad real de la interfaz. Estas pruebas permitieron observar directamente cómo los usuarios interactuaban con la plataforma, identificar dificultades recurrentes y recopilar recomendaciones de mejora.

Con base en estos resultados se ajustaron varios elementos de diseño, como la ubicación de los componentes individuales, la claridad de los botones e instrucciones, y la optimización de los tiempos de carga.

La interfaz final facilita a los alumnos el acceso al contenido producido por inteligencia artificial, la solicitud de explicaciones adicionales cuando sea necesario y la realización de

ejercicios adaptados a su propio ritmo de aprendizaje. Todo esto se ofrece en un ambiente visual práctico y claro, creado para disminuir la carga cognitiva y simplificar el proceso de aprendizaje.

Este módulo resultó ser un puente esencial para que el simulador funcionara correctamente, facilitando la gestión de aspectos técnicos que quedaron transparentes para los usuarios. El hecho de haber documentado cuidadosamente todo el proceso hará más sencillo implementar mejoras futuras o vincular la herramienta 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. Revisar herramientas de IA generativa y analizar cómo se aplican para extraer buenas prácticas en el 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.

Durante el proceso, analizamos varios elementos de la generación de IA que se pueden integrar a la interfaz. Con el objetivo de entender nuestra cultura y las expectativas que tenemos sobre estas plataformas, llevamos a cabo una investigación con alumnos universitarios. Para reunir datos acerca de las personas, nos enfocamos en las capacidades y los trabajos requeridos para crear el componente.

Una vez que tuvimos claros los requisitos, empezamos a crear prototipos básicos y luego otros más detallados para ir probando el diseño paso a paso. Lo que más nos importaba era que fuera fácil de usar, así que trabajamos en que las instrucciones fueran claras y en explicar bien cómo funcionaba la IA. La idea era lograr que cualquier usuario pudiera moverse por el simulador de manera intuitiva, sin complicarse.

Gracias a la utilización de tecnologías web actuales para la implementación, se logró integrar de manera óptima los módulos principales del simulador, incluyendo la traducción automática y las clases creadas. Además, garantizamos que el diseño sea adaptable a todos los dispositivos (computadoras, tablets y celulares) para que los estudiantes puedan utilizarlo desde cualquier sitio.

Hicimos pruebas con los estudiantes para ver qué problemas tenían al usar las herramientas. A partir de lo que nos dijeron, fuimos mejorando el sistema poco a poco: ajustamos los tiempos de carga, cambiamos algunos botones y otros elementos que se veían, e hicimos mejoras en lo visual. Todo esto terminó mejorando bastante la experiencia que tenían los usuarios al final.

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, conside-

rando 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 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. Herramien-

tas 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 permite 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

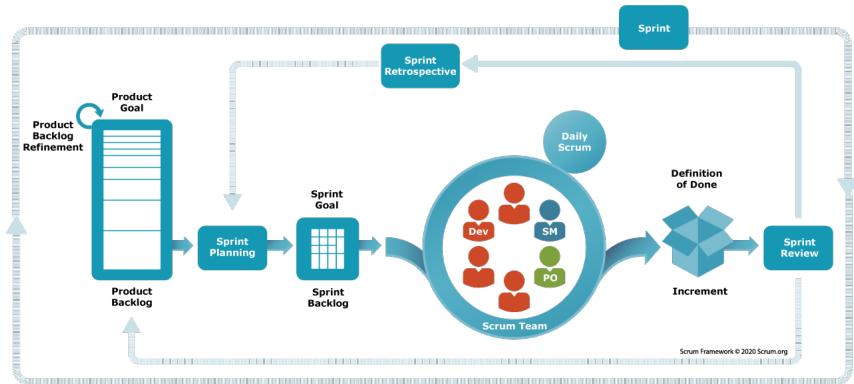


Figura 2.1: Framework SCRUM (basado en Scrum.org) [28]

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.

Es importante señalar que los prototipos desarrollados en este sprint corresponden a representaciones de baja fidelidad, cuyo propósito principal es validar la secuencia de navegación, la disposición general de los elementos y la lógica de interacción entre pantallas, sin profundizar aún en aspectos visuales definitivos como colores, tipografías o estilos gráficos finales. En esta etapa, el énfasis se coloca en comprender cómo el usuario recorre el sistema y cómo se articulan las distintas pantallas dentro del proceso de simulación, más que en la apariencia gráfica del sistema. La Figura 2.2 presenta el diagrama de navegación propuesto, el cual resume el recorrido principal del usuario desde el inicio de sesión hasta la visualización de resultados, así como el acceso transversal a la sección de ayu-

da. Los aspectos visuales y de diseño detallado se abordan posteriormente en el Sprint 3, correspondiente al desarrollo de prototipos de alta fidelidad.

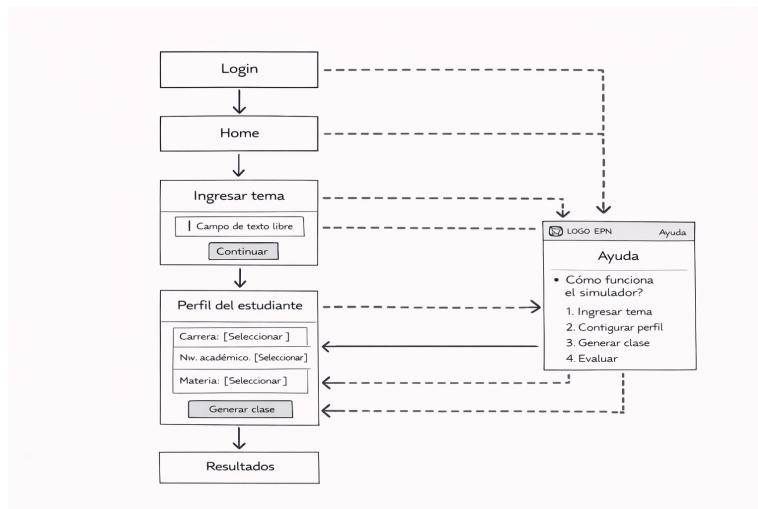


Figura 2.2: Estructura de navegación del Simulador de Clases Personalizadas con IA

2.3.4 Sprint 3: Diseño de prototipos de alta fidelidad

El Sprint 3 se orientó al diseño de prototipos de alta fidelidad del Simulador de Clases Personalizadas con IA, tomando como base los flujos de navegación y las estructuras definidas en el Sprint 2. Mientras que la etapa anterior permitió validar la lógica del recorrido y la organización de la información, este sprint se centró en materializar dichas decisiones a un nivel visual más detallado, acercando el diseño a la experiencia final del usuario.

La necesidad de este sprint surge de la importancia que tiene la presentación visual en sistemas educativos digitales. Una vez definido qué información se muestra y en qué orden, resulta fundamental evaluar cómo los colores, tipografías, iconografía y jerarquías visuales influyen en la comprensión del contenido y en la percepción general del sistema. En este sentido, los prototipos de alta fidelidad permiten analizar aspectos que no pueden evaluarse con wireframes, como la claridad visual, la coherencia gráfica y la consistencia entre pantallas.

En este proyecto, el Sprint 3 permitió refinar la experiencia de usuario al incorporar elementos visuales alineados con el contexto institucional y académico del simulador. El diseño buscó transmitir una sensación de formalidad y confianza, sin sacrificar claridad ni facilidad de uso. De esta manera, los prototipos de alta fidelidad no solo funcionaron como una representación estética del sistema, sino como una herramienta para evaluar si las decisiones de diseño apoyan efectivamente el proceso de aprendizaje planteado.

Para materializar este análisis, los prototipos de alta 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.

Pantalla de inicio (Home)

La pantalla de inicio cumple la función de introducir al usuario al simulador y de marcar, desde el primer momento, el propósito del sistema. En este proyecto se decidió que esta pantalla sea simple y directa, ya que su objetivo no es ofrecer múltiples opciones, sino guiar al usuario hacia el inicio del proceso de simulación de clases personalizadas.

En términos de contenido, la pantalla presenta un mensaje principal que identifica claramente al sistema como un simulador de clases personalizadas con apoyo de inteligencia artificial. Este mensaje se complementa con un texto breve que explica el valor del simulador, resaltando la posibilidad de aprender a un ritmo propio a partir del perfil académico del estudiante. La intención de esta redacción es que el usuario comprenda rápidamente qué ofrece la plataforma sin necesidad de explicaciones extensas.

Desde el punto de vista de la interacción, se priorizó una acción principal claramente visible:

iniciar la simulación. Esta decisión se tomó para evitar que el usuario tenga que evaluar varias alternativas al mismo tiempo, lo cual podría generar confusión en una etapa temprana del uso del sistema. De forma secundaria, se incluye una opción para conocer cómo funciona el simulador, pensada para usuarios que requieren mayor contexto antes de avanzar, pero sin interferir con el flujo principal.

La estructura visual de esta pantalla refuerza el flujo definido en el Sprint 2, ya que orienta al usuario de manera natural hacia la configuración de su perfil académico, que corresponde al siguiente paso del proceso. En la Figura 2.3, se muestra el prototipo de baja fidelidad de la pantalla de inicio desarrollado en Figma, el cual permitió validar la jerarquía de los elementos, la claridad del mensaje principal y la correcta identificación de las acciones disponibles antes de continuar con las siguientes pantallas del simulador.



Figura 2.3: Pantalla de inicio del simulador

Pantalla de ingreso del tema

La pantalla de ingreso del tema corresponde a la etapa en la que el usuario comienza a definir de forma concreta el contenido académico que será utilizado para generar la clase personalizada. En este punto del flujo, el sistema solicita información clave que permite contextualizar el proceso de simulación y orientar la generación del contenido de acuerdo con las necesidades reales del estudiante.

En esta pantalla, el usuario ingresa progresivamente datos relacionados con su formación académica, como la carrera, el nivel académico y la materia, antes de especificar el tema sobre el cual desea aprender. Esta organización no es casual, ya que se decidió solicitar primero información general y luego información específica, con el fin de evitar confusión y

facilitar la toma de decisiones por parte del usuario. De esta manera, el estudiante puede comprender mejor el alcance de lo que está configurando antes de definir el tema concreto. Desde la perspectiva de la experiencia de usuario, se optó por el uso de campos desplegables y un campo de texto claramente identificado para el ingreso del tema. Esta combinación permite reducir errores de ingreso y guía al usuario durante la configuración, manteniendo la interacción simple y controlada. Además, el mensaje introductorio de la pantalla explica de forma breve por qué se solicita esta información, reforzando la idea de que los datos ingresados influyen directamente en la personalización de la clase.

La disposición visual de los elementos también cumple un rol importante dentro del flujo de navegación. El formulario se presenta de manera centrada y ordenada, mientras que el menú lateral permite al usuario identificar en qué etapa del proceso se encuentra, sin distraerlo de la tarea principal. Esta decisión busca mantener el foco en la configuración del tema y preparar al usuario para la siguiente etapa del simulador.

En la Figura 2.4 se presenta el prototipo de baja fidelidad de la pantalla de ingreso del tema, desarrollado en Figma. Este prototipo permitió validar la secuencia de los campos, la claridad de las etiquetas y la comprensión general de la tarea que el usuario debe realizar antes de continuar con la generación de la clase.

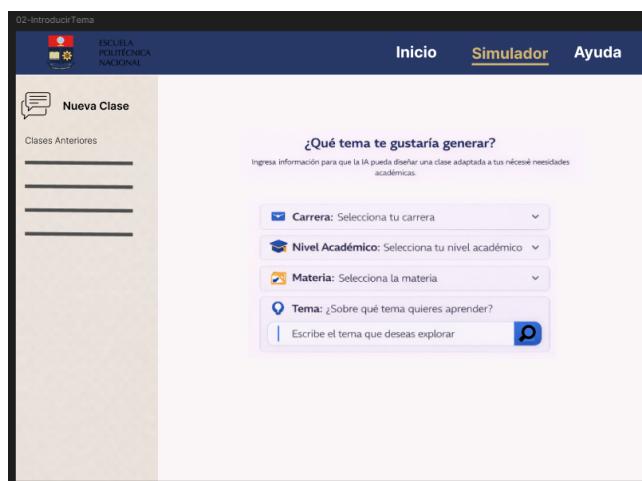


Figura 2.4: Pantalla de ingreso del tema del simulador

Pantalla de generación de la clase

La pantalla de generación de la clase representa una etapa intermedia clave dentro del flujo de navegación del simulador. En este punto, el usuario ya ha definido su perfil académico y el tema de interés, por lo que el sistema inicia el proceso de construcción de la clase

personalizada. Aunque el procesamiento interno no es visible para el usuario, esta pantalla cumple la función de comunicar de manera clara que el sistema se encuentra trabajando con la información proporcionada.

En el diseño de esta pantalla se consideró importante evitar la percepción de inactividad o error del sistema. Por esta razón, se incorporó un mensaje explícito que informa al usuario que la clase está siendo generada en función de su perfil académico, acompañado de un indicador visual de progreso. Este recurso permite reducir la incertidumbre durante el tiempo de espera y mantiene al usuario informado sobre el estado del proceso.

Desde la perspectiva de la experiencia de usuario, esta pantalla actúa como un punto de transición entre la configuración del tema y la visualización del contenido generado. Su diseño es intencionalmente simple, con pocos elementos visuales, de modo que la atención del usuario se centre en el mensaje principal y no se generen distracciones innecesarias. Esta decisión busca reforzar la continuidad del flujo y preparar al usuario para la siguiente etapa del simulador.

En la Figura 2.5 se muestra el prototipo de baja fidelidad de la pantalla de generación de la clase, desarrollado en Figma. Este prototipo permitió validar la claridad del mensaje, la ubicación del indicador de progreso y la comprensión general del estado del sistema antes de presentar la clase personalizada al usuario.

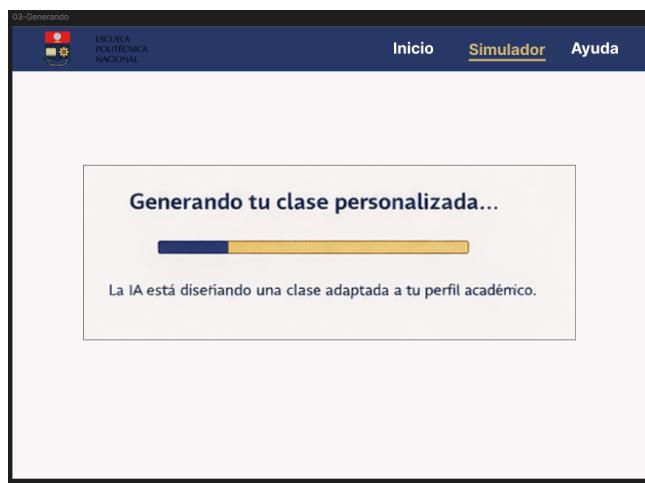


Figura 2.5: Pantalla de generación de la clase personalizada

Pantalla de clase generada

La pantalla de clase generada constituye el núcleo del simulador, ya que es el espacio donde el estudiante interactúa directamente con el contenido educativo personalizado. En

esta etapa, el sistema presenta el resultado del proceso de generación realizado a partir del perfil académico y del tema previamente definido, materializando el objetivo principal del simulador.

En este proyecto se decidió organizar el contenido de la clase de forma estructurada y progresiva, comenzando con una explicación general del tema y avanzando hacia elementos de apoyo como esquemas visuales, ejemplos y recursos complementarios. Esta organización busca facilitar la comprensión del contenido, permitiendo que el estudiante construya una idea global antes de profundizar en los detalles específicos.

La interfaz incorpora un menú lateral que permite identificar distintas secciones, como la creación de una nueva clase o el acceso a clases anteriores. Esta decisión responde a la necesidad de ofrecer continuidad en el aprendizaje, sin obligar al usuario a reiniciar el proceso cada vez que desea revisar o generar nuevo contenido. Al mismo tiempo, el diseño mantiene el foco en la clase actual, evitando que la navegación secundaria interfiera con la lectura y análisis del tema.

Un elemento relevante de esta pantalla es la inclusión de acciones orientadas al apoyo del aprendizaje, como las opciones para solicitar más ejemplos, indicar que el tema no fue comprendido o acceder a referencias bibliográficas. Estas opciones permiten al estudiante ajustar la profundidad del contenido según su nivel de comprensión, reforzando un aprendizaje más activo y personalizado en lugar de una lectura pasiva.

En la Figura 2.6 se presenta el prototipo de baja fidelidad de la pantalla de clase generada, desarrollado en Figma. Este prototipo permitió evaluar la disposición del contenido, la claridad de las secciones y la utilidad de las acciones de apoyo, asegurando que el estudiante pueda interactuar con la clase de manera flexible y comprensible.



Figura 2.6: Pantalla de visualización de la clase generada

Pantalla de evaluación

La pantalla de evaluación corresponde a la etapa en la que el estudiante tiene la oportunidad de comprobar su nivel de comprensión sobre el contenido presentado en la clase generada. En este punto del flujo, el objetivo principal no es calificar formalmente al usuario, sino ofrecer un espacio de autoevaluación que permita reforzar el aprendizaje y promover una reflexión inmediata sobre lo estudiado.

En este proyecto se optó por una evaluación breve y directa, compuesta por preguntas de selección múltiple, con el fin de no interrumpir el ritmo del aprendizaje ni generar una carga cognitiva excesiva. La formulación de una pregunta a la vez permite que el estudiante se concentre en el contenido evaluado, sin distracciones adicionales ni elementos innecesarios en la interfaz.

Desde la perspectiva de la experiencia de usuario, esta pantalla mantiene una estructura visual sencilla, donde la pregunta y las opciones de respuesta ocupan el centro de la atención. El botón para avanzar hacia los resultados se presenta de forma clara y diferenciada, de modo que el usuario comprenda cuándo ha finalizado la evaluación y cuál es el siguiente paso dentro del simulador.

Además, la ubicación de esta pantalla dentro del flujo refuerza la lógica del proceso de simulación, ya que conecta directamente la visualización de la clase con la retroalimentación posterior. De esta manera, el estudiante puede cerrar el ciclo de aprendizaje de forma inmediata, sin necesidad de abandonar el contexto del simulador.

En la Figura 2.7 se presenta el prototipo de baja fidelidad de la pantalla de evaluación, desarrollado en Figma. Este prototipo permitió validar la claridad de las preguntas, la disposición de las opciones de respuesta y la comprensión del flujo hacia la visualización de resultados, antes de avanzar a etapas de mayor fidelidad del diseño.

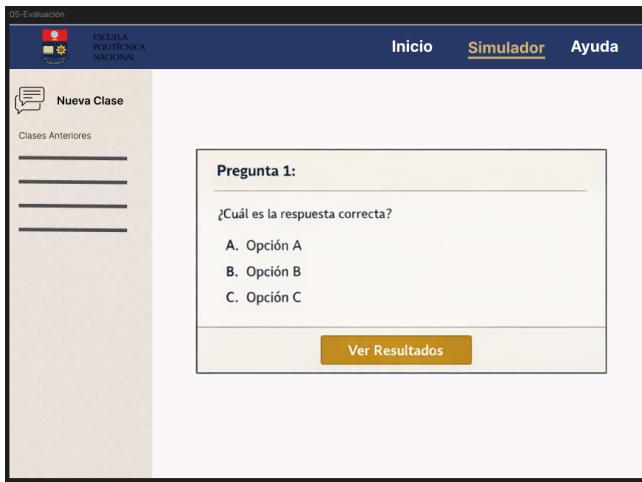


Figura 2.7: Pantalla de evaluación del simulador

Pantalla de resultados y retroalimentación

La pantalla de resultados y retroalimentación corresponde a la etapa final del flujo de navegación del simulador y cumple la función de cerrar el proceso de aprendizaje iniciado con la generación de la clase. En esta pantalla, el estudiante recibe una síntesis clara de su desempeño, junto con orientaciones que le permiten decidir cómo continuar su proceso formativo dentro del sistema.

En este proyecto se decidió presentar el resultado de la evaluación mediante un nivel de desempeño cualitativo, en lugar de una calificación numérica estricta. Esta decisión busca reducir la presión asociada a la evaluación tradicional y orientar la atención del estudiante hacia la mejora continua, más que hacia la obtención de una nota. El mensaje de retroalimentación inicial refuerza esta idea, utilizando un lenguaje cercano que reconoce el esfuerzo realizado.

La interfaz ofrece recomendaciones concretas, como repasar el tema, avanzar al siguiente nivel o consultar material de refuerzo. Estas opciones permiten que el estudiante tenga control sobre su siguiente paso, adaptando el recorrido a su nivel de comprensión y necesidades personales. De este modo, la retroalimentación no se limita a informar un resultado, sino que se convierte en una guía para la toma de decisiones dentro del simulador.

Además, la pantalla incluye acciones claras para continuar interactuando con el sistema, como generar una nueva clase, cambiar de tema o regresar al inicio. Estas opciones refuerzan la continuidad del flujo y evitan que el usuario quede sin una dirección clara al finalizar la evaluación, manteniendo la experiencia de uso fluida y coherente.

En la Figura 2.8 se presenta el prototipo de baja fidelidad de la pantalla de resultados y retro-

alimentación, desarrollado en Figma. Este prototipo permitió validar la claridad del mensaje de desempeño, la comprensión de las recomendaciones ofrecidas y la identificación de las acciones disponibles para continuar el proceso de aprendizaje.



Figura 2.8: Pantalla de resultados y retroalimentación del simulador

Pantalla de ayuda

La pantalla de ayuda fue diseñada como un espacio de apoyo permanente para el usuario, pensado especialmente para quienes interactúan por primera vez con el simulador o requieren aclaraciones durante su uso. Su objetivo principal es explicar de forma sencilla y accesible cómo funciona el sistema, sin interrumpir el flujo principal de aprendizaje.

En este proyecto se consideró necesario incluir esta pantalla debido a que el simulador integra varios pasos consecutivos —configuración del perfil, generación de la clase, interacción con el contenido y evaluación— que pueden no resultar evidentes para todos los usuarios desde el inicio. Por esta razón, la ayuda presenta el funcionamiento del sistema de manera secuencial, describiendo cada etapa del proceso y relacionándola con la experiencia del estudiante.

La información se organiza en pasos numerados que explican, de forma resumida, qué ocurre en cada fase del simulador. Este enfoque permite que el usuario comprenda rápidamente el flujo general sin necesidad de leer instrucciones extensas. Además, el uso de elementos visuales de apoyo refuerza la comprensión del proceso y facilita la asociación entre la explicación y las pantallas del sistema.

Asimismo, la pantalla de ayuda incorpora un índice de navegación que permite al usuario ubicarse dentro del sistema y acceder fácilmente a las secciones principales. Esta deci-

sión busca que la ayuda no sea un elemento aislado, sino una referencia constante que acompañe al usuario durante su interacción con el simulador.

En la Figura 2.9 se presenta el prototipo de baja fidelidad de la pantalla de ayuda, desarrollado en Figma. Este prototipo permitió validar la claridad de las explicaciones, la organización de la información y la utilidad de los elementos visuales, asegurando que la ayuda cumpla su función de orientación sin generar sobrecarga cognitiva.

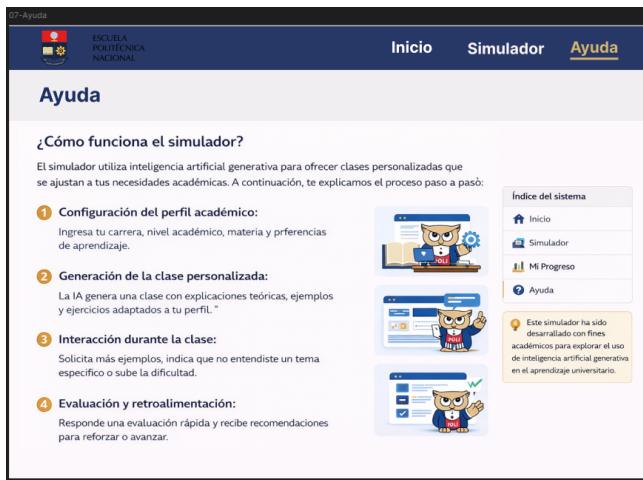


Figura 2.9: Pantalla de ayuda del simulador

2.3.5 Sprint 4: Implementación inicial de la interfaz

El Sprint 4 se orientó a la implementación inicial de la interfaz del Simulador de Clases Personalizadas con IA, a partir de los prototipos de alta fidelidad definidos en el Sprint 3. En esta etapa, el diseño conceptual y visual previamente validado se traduce en una estructura funcional básica, permitiendo comprobar la viabilidad técnica de las decisiones de diseño y evaluar el comportamiento general de la interfaz en un entorno de prueba.

La incorporación de este sprint resulta necesaria debido a que los prototipos, aunque útiles para validar la experiencia de usuario y la coherencia visual del sistema, no permiten analizar aspectos propios de una interfaz en funcionamiento, como la respuesta ante interacciones, la navegación real entre pantallas o la organización del código que soporta la presentación del contenido. Por ello, el Sprint 4 marca el primer acercamiento del proyecto a un producto operativo, aunque aún en una fase preliminar.

Durante este sprint, la implementación se centró en construir una versión básica de la interfaz que refleje fielmente los flujos definidos, sin incorporar todavía la totalidad de las funcionalidades del simulador. El objetivo principal no fue desarrollar un sistema completo, sino establecer una base estructural sólida que permita visualizar las pantallas, navegar

entre ellas y simular la experiencia de uso desde una perspectiva más cercana a la realidad. De esta manera, el Sprint 4 actúa como un puente entre el diseño y la integración funcional del sistema, permitiendo identificar ajustes necesarios antes de avanzar hacia la conexión con los módulos internos del simulador y las pruebas de usabilidad. Esta aproximación progresiva reduce riesgos en etapas posteriores y facilita una implementación más controlada y coherente con los objetivos del proyecto.

2.3.5.1 4.1 Entorno de desarrollo y tecnologías empleadas

Para la implementación inicial de la interfaz del simulador se optó por un entorno de desarrollo sencillo, basado en tecnologías web conocidas, que permitiera concentrarse en validar la estructura general de la aplicación y su flujo de navegación, más que en la complejidad técnica. En esta etapa, la prioridad fue comprobar que las pantallas diseñadas en los sprints anteriores podían trasladarse a un entorno real de ejecución y que el recorrido del usuario resultara coherente y funcional.

La elección de estas tecnologías respondió a su facilidad de uso y a su capacidad para representar de forma clara la estructura y el comportamiento básico de la interfaz. No se buscó construir una aplicación completamente terminada, sino una versión inicial que hiciera posible simular la experiencia de uso del sistema, identificar ajustes necesarios y verificar la correspondencia entre el diseño y su implementación.

De manera concreta, el entorno de desarrollo estuvo conformado por los siguientes elementos:

- **HTML:** se utilizó para definir la estructura base de cada pantalla, organizando los elementos principales y estableciendo la jerarquía del contenido. Su uso permitió trasladar los prototipos a una forma tangible dentro de la aplicación web.
- **CSS:** se empleó para aplicar estilos iniciales a la interfaz, respetando la distribución visual planteada en los prototipos. En esta fase, el objetivo no fue alcanzar un diseño final, sino aproximarse a la organización visual necesaria para evaluar la legibilidad y el orden de la información.
- **JavaScript:** se usó para simular la interacción entre pantallas y permitir una navegación básica dentro del simulador. Gracias a esto, fue posible comprobar el funcionamiento del flujo definido en el Sprint 2 y observar cómo el usuario avanzaría a través de las distintas etapas del sistema.

- **Visual Studio Code:** se eligió como entorno de desarrollo por su facilidad de uso y por permitir una organización clara de los archivos del proyecto. Esta herramienta facilitó el trabajo progresivo sobre la interfaz y la revisión constante de los cambios realizados.
- **Figma:** continuó utilizándose como referencia visual durante la implementación. Los prototipos desarrollados en el Sprint 3 sirvieron como guía para mantener coherencia entre el diseño propuesto y lo que se iba construyendo en la aplicación web.
- **Google Chrome:** se empleó como navegador principal para la ejecución y validación de la interfaz. Esto permitió observar el comportamiento real de la aplicación, comprobar la correcta visualización de los elementos y evaluar la navegación entre pantallas.

2.3.5.2 4.2 Estructura general del flujo y de la aplicación

En esta etapa se buscó que la estructura de la aplicación web refleje de manera directa el flujo de uso definido para el simulador. No se trató únicamente de “conectar pantallas”, sino de asegurar que el recorrido del usuario tenga sentido, sea predecible y mantenga una lógica clara desde el primer acceso hasta la obtención de resultados. La organización general se diseñó pensando en que cada paso prepare al siguiente, evitando saltos innecesarios o situaciones en las que el usuario no tenga claro qué debe hacer.

El flujo general de uso del sistema se concibió como un proceso secuencial y guiado. El usuario inicia su interacción desde una pantalla de bienvenida, que funciona como punto de entrada al simulador. A partir de ahí, el sistema lo conduce hacia la definición del contexto académico, lo cual constituye la base para la generación posterior de los contenidos. Este planteamiento responde a la necesidad de que el simulador siempre trabaje sobre información previamente establecida, evitando resultados ambiguos o descontextualizados. Una vez configurado el contexto, el usuario es dirigido al espacio donde puede interactuar con los contenidos generados. En esta parte del flujo, el sistema deja de ser únicamente un generador de información y pasa a convertirse en un entorno de trabajo, en el que el estudiante puede consultar resultados, explorar explicaciones adicionales y avanzar hacia la evaluación de su aprendizaje. Esta secuencia permite validar el funcionamiento integral del sistema, desde la entrada de datos hasta la obtención de resultados personalizados.

La aplicación se organizó en distintas pantallas o vistas, cada una con un propósito claro dentro de este flujo. Esta separación permitió trabajar de manera ordenada durante la implementación y facilitó la validación individual de cada etapa del proceso. De forma general,

las pantallas principales consideradas fueron:

- **Pantalla de inicio (Home):** introduce el propósito del simulador y permite al usuario comenzar el flujo principal de uso.
- **Pantalla de ingreso del tema:** permite definir el contexto académico que será utilizado para la generación de la clase.
- **Pantalla de generación:** representa el estado intermedio mientras el sistema procesa la información ingresada.
- **Pantalla de clase generada:** muestra el contenido educativo personalizado.
- **Pantalla de evaluación:** permite al usuario comprobar su nivel de comprensión.
- **Pantalla de resultados:** presenta la retroalimentación y orienta los siguientes pasos.
- **Pantalla de ayuda:** ofrece soporte informativo durante todo el proceso.

Esta separación por vistas permitió mantener una estructura clara y comprensible, tanto desde la perspectiva del usuario como desde el punto de vista del desarrollo.

A nivel técnico, la aplicación fue organizada bajo una lógica modular, inspirada en el paradigma de componentes. Esta decisión permitió dividir la interfaz en partes más pequeñas y manejables, facilitando el desarrollo progresivo y la corrección de errores. Más que buscar una arquitectura compleja, el objetivo fue mantener el código legible y adaptable a futuras ampliaciones.

Dentro de esta organización, se identificaron algunos elementos transversales que estructuran la experiencia de uso:

- **Navbar:** encargado de mantener una referencia constante de navegación y orientación dentro del sistema.
- **Componentes de visualización de contenido:** responsables de mostrar la información generada por el simulador de forma ordenada.
- **Componentes de interacción:** orientados a permitir la participación activa del usuario durante el proceso.

2.3.5.3 4.3 Implementación de la navegación entre pantallas

En este sprint la navegación entre pantallas se trabajó como un elemento central, ya que permitió comprobar si el flujo definido en las etapas de diseño realmente funciona cuando se lo traslada a una aplicación web. Más allá de la parte técnica, el interés principal fue observar si el recorrido del usuario es lógico, continuo y fácil de seguir, y si cada pantalla aparece en el momento adecuado dentro del proceso de uso del simulador.

Se decidió que la navegación sea progresiva y controlada, de modo que el usuario no pueda acceder a pantallas que dependen de información que todavía no ha sido definida. Esto evita situaciones confusas, como intentar visualizar una clase sin haber configurado previamente el tema, y asegura que el sistema siempre trabaje con un contexto válido. De esta manera, la propia navegación se convierte en una forma de guiar y ordenar la interacción del usuario.

De forma general, el flujo de navegación implementado sigue la siguiente secuencia:

- **Pantalla de inicio (Home):** punto de entrada al sistema, desde donde el usuario puede comenzar la simulación o acceder a la sección de ayuda.
- **Pantalla de ingreso del tema:** espacio donde el usuario define el contexto académico que servirá como base para la generación de la clase.
- **Pantalla de generación de la clase:** vista intermedia que indica que el sistema está procesando la información ingresada.
- **Pantalla de clase generada:** muestra el contenido educativo personalizado.
- **Pantalla de evaluación:** permite al usuario comprobar su nivel de comprensión.
- **Pantalla de resultados y retroalimentación:** presenta el desempeño obtenido y orienta los siguientes pasos.

Este orden responde directamente al ciclo de uso que se busca simular: configuración, generación, estudio, evaluación y reflexión. Al implementarlo de esta manera, fue posible verificar que el usuario puede recorrer todo el proceso sin perder el contexto de lo que está haciendo.

Durante el desarrollo del Sprint 4, varias de las transiciones entre pantallas se implementaron de forma simulada, utilizando datos estáticos o comportamientos predefinidos. Esta decisión fue intencional, ya que el objetivo principal no era contar todavía con un sistema

completamente funcional, sino validar la experiencia de navegación y la coherencia visual del flujo completo.

Desde el punto de vista técnico, la navegación se organizó mediante un sistema de rutas que define de manera explícita cómo se pasa de una vista a otra. Sin entrar en detalles de implementación, esta estructura permitió asegurar que cada pantalla tenga un lugar claro dentro de la aplicación y que el recorrido del usuario sea siempre predecible.

En conjunto, esta implementación de la navegación permitió confirmar que el flujo conceptual definido en el Sprint 2 puede trasladarse a un entorno real de ejecución. Además, facilitó la identificación temprana de ajustes necesarios en el orden de las pantallas o en la forma en que el usuario avanza entre ellas, lo cual resulta fundamental antes de integrar la lógica completa del sistema y los módulos de inteligencia artificial.

2.3.5.4 4.4 Implementación inicial de las pantallas principales

En esta etapa del desarrollo, el trabajo se centró en llevar a la práctica las pantallas más importantes del simulador, tomando como base los mockups definidos en el Sprint 3. No se buscó que todas las vistas estén completamente funcionales, sino que el sistema ya permita recorrer el flujo principal, interactuar de forma básica y comprobar que la estructura general es viable desde el punto de vista técnico y de experiencia de usuario.

Esta implementación inicial permitió pasar de un diseño visual a una aplicación web tangible, donde es posible observar cómo se comportan las pantallas en un entorno real de ejecución. Además, sirvió para identificar qué partes del sistema podían conectarse directamente con el backend y cuáles debían mantenerse aún en un nivel de simulación.

Pantalla de inicio (Home)

La pantalla de inicio fue desarrollada como el punto de entrada principal al simulador. Su implementación se enfocó en representar fielmente la estructura definida en los prototipos y en permitir la navegación hacia las secciones clave del sistema. En esta fase, la funcionalidad está orientada principalmente a la presentación de información y a la activación de la navegación, sin incluir todavía procesos internos complejos.

En concreto, esta pantalla permitió validar:

- La correcta visualización del mensaje introductorio del simulador.
- La identificación clara de la acción principal para iniciar la simulación.

- El acceso directo a la sección de ayuda.

La Figura 2.10 muestra la implementación inicial de la pantalla de inicio en el entorno web.



Figura 2.10: Implementación inicial de la pantalla Home

Pantalla de carga de documento (*IntroducirTema*)

La pantalla de carga de documento fue una de las primeras en contar con una funcionalidad real, ya que permite al usuario seleccionar y enviar un archivo académico al sistema. A diferencia de otras vistas, aquí sí se estableció una comunicación directa con el backend, lo cual permitió validar el inicio del flujo funcional completo.

Esta implementación hizo posible:

- La selección de un archivo desde el dispositivo del usuario.
- El envío del archivo al backend para su procesamiento.
- La recepción de un identificador de documento que sirve como contexto activo.

Gracias a esta pantalla, fue posible comprobar que el sistema puede mantener la persistencia del documento y que el contexto de trabajo se conserva al avanzar hacia las siguientes vistas.

La Figura 2.11 presenta la implementación inicial de la pantalla de carga de documento.



Figura 2.11: Implementación inicial de la pantalla de carga de documento

Espacio de trabajo (Workspace)

El espacio de trabajo fue implementado como la pantalla central del simulador, ya que en ella se concentran las principales interacciones del usuario. Esta vista representa el punto donde el sistema deja de ser únicamente un generador de contenido y se convierte en un entorno activo de aprendizaje.

En esta etapa, el Workspace permitió validar:

- La integración inicial con el backend para ejecutar acciones básicas.
- La organización visual de los resultados generados.
- La activación de funcionalidades como resúmenes, preguntas de práctica y planes de estudio.
- La interacción mediante un chat con el asistente virtual, aunque con respuestas aún limitadas o parcialmente simuladas.

Aunque no todas las respuestas del backend están completamente habilitadas, esta pantalla permitió comprobar que la lógica de interacción general funciona y que el flujo de información entre frontend y backend es técnicamente viable.

La Figura 2.12 muestra la implementación inicial del espacio de trabajo.

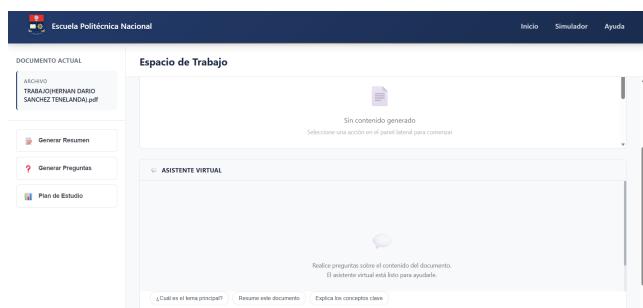


Figura 2.12: Implementación inicial del espacio de trabajo

Pantallas complementarias

Las pantallas Generando, ClaseGenerada, Evaluación y Resultados fueron implementadas de manera parcial o simulada. En esta etapa su propósito no fue ofrecer funcionalidades completas, sino validar la continuidad visual del sistema y asegurar que el usuario puede recorrer todo el flujo sin interrupciones.

Estas vistas permitieron:

- Representar estados intermedios del sistema, como el procesamiento de información.
- Simular la visualización de contenidos generados.
- Comprobar la transición hacia la evaluación y la retroalimentación.
- Mantener coherencia gráfica entre todas las pantallas.

La implementación de estas pantallas resultó clave para evaluar la experiencia completa del usuario, incluso cuando parte de la lógica aún se encuentra en desarrollo.

La Figura 2.13 ilustra la implementación inicial de algunas de estas vistas complementarias.

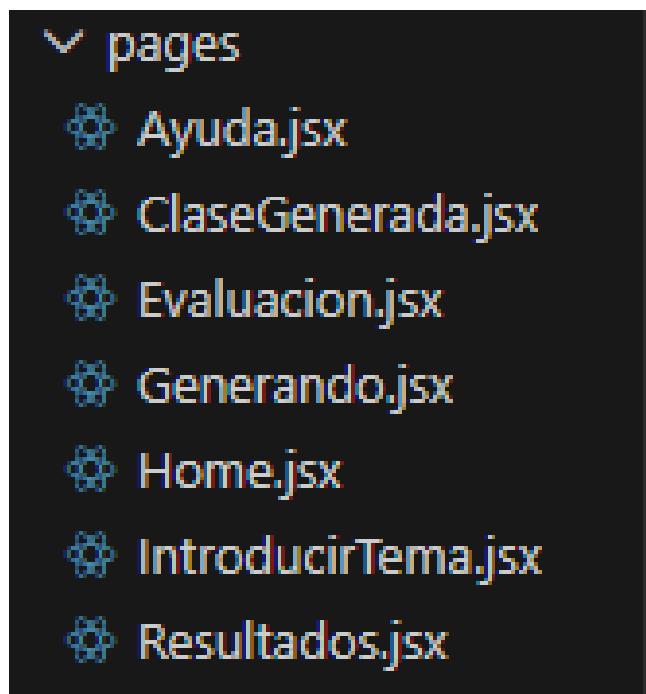


Figura 2.13: Implementación inicial de pantallas complementarias del simulador

2.3.5.5 4.5 Pruebas funcionales y limitaciones de la implementación

2.3.5.6 4.6 Resultados del Sprint 4 y mejoras iterativas

El Sprint 4 permitió consolidar una primera versión funcional del frontend del simulador, en la cual fue posible recorrer el flujo principal de uso y validar la coherencia entre el diseño propuesto y su implementación real. Más allá de contar con pantallas visualmente completas, el principal resultado de este sprint fue comprobar que la estructura general de la aplicación es viable y que la navegación entre vistas responde de manera adecuada a las acciones del usuario.

Tabla 2.14: Resumen de pruebas funcionales realizadas en el frontend

ID	Área evaluada	Descripción de la prueba	Resultado
PF01	Navegación entre pantallas	Verificación del recorrido completo desde Home hasta el Workspace respetando el flujo definido.	Correcto
PF02	Carga de documentos	Envío de archivo académico al backend y recepción del identificador del documento procesado.	Correcto
PF03	Persistencia de contexto	Comprobación de que el documento cargado se mantiene como contexto activo durante la sesión.	Correcto
PF04	Acciones principales	Ejecución de botones para la generación de resúmenes, preguntas de práctica y planes de estudio.	Parcial
PF05	Visualización de resultados	Presentación adecuada de las respuestas recibidas desde el backend en la interfaz.	Parcial
PF06	Chat del asistente virtual	Envío y recepción de mensajes asociados al documento cargado dentro del espacio de trabajo.	Parcial
PF07	Manejo de estados vacíos	Visualización de mensajes cuando no existe documento cargado o contenido generado.	Correcto
PF08	Coherencia visual	Validación de la consistencia gráfica entre las pantallas principales del sistema.	Correcto

Tabla 2.15: Limitaciones identificadas en la implementación actual

ID	Descripción de la limitación	Impacto
L01	Restricciones en el uso de servicios de inteligencia artificial que limitan la cantidad de pruebas continuas.	Medio
L02	Respuestas del backend parcialmente simuladas en algunas funcionalidades del sistema.	Medio
L03	Pantallas de generación completa de clases, evaluación y resultados implementadas de forma parcial.	Alto
L04	Ausencia de pruebas automatizadas durante esta etapa inicial del desarrollo.	Medio
L05	Dependencia de componentes externos aún en desarrollo para completar la funcionalidad total del simulador.	Alto

A partir de las pruebas realizadas se confirmó que el sistema permite iniciar el proceso desde la pantalla principal, cargar un documento académico, mantener dicho documento como contexto activo y acceder al espacio de trabajo sin inconsistencias. Este resultado valida que la lógica básica de interacción está correctamente establecida y que la integración inicial entre frontend y backend funciona dentro de los límites definidos para esta etapa.

Asimismo, el Sprint 4 permitió identificar aspectos de la interfaz que requerían ajustes menores, principalmente relacionados con la claridad de los textos, la disposición de algunos botones y la retroalimentación visual ofrecida al usuario. Estos cambios no respondieron a fallos críticos del sistema, sino a la necesidad de hacer la interacción más intuitiva y coherente con el flujo de aprendizaje que se busca promover.

Otro resultado importante fue reconocer qué partes del sistema aún deben mantenerse en un nivel de simulación. Las pantallas asociadas a la generación completa de clases, la evaluación y los resultados finales demostraron ser útiles para validar la experiencia de usuario, aun cuando su lógica funcional todavía no esté completamente desarrollada. Esto permitió separar con claridad lo que corresponde a validación de interfaz de lo que será trabajo de integración avanzada en sprints posteriores.

2.3.5.7 5.1 Integración entre la interfaz y los módulos del simulador

La integración entre la interfaz de usuario y los módulos funcionales del simulador representa un cambio importante en el comportamiento del sistema, ya que marca el paso desde una aplicación principalmente visual hacia una plataforma capaz de responder dinámicamente a las acciones del usuario. En esta fase, la interfaz deja de cumplir un rol únicamente representativo y pasa a convertirse en el medio a través del cual se activa la lógica central del simulador.

Desde el punto de vista estructural, esta integración se apoya en una arquitectura cliente–servidor, donde la aplicación desarrollada en React actúa como cliente y el backend del simulador concentra los módulos encargados del procesamiento de información. La comunicación entre ambos se realiza a través de servicios REST, los cuales permiten enviar datos estructurados y recibir respuestas procesadas de manera ordenada. Este enfoque facilita la separación de responsabilidades entre la capa de presentación y la capa lógica del sistema, haciendo que cada una pueda evolucionar de forma independiente.

La interfaz comienza a adquirir un comportamiento dinámico cuando empieza a consumir activamente los servicios del backend. Cada acción del usuario deja de ser un simple cam-

bio visual y pasa a convertirse en una solicitud real al sistema. Este intercambio se implementa mediante peticiones HTTP asíncronas, lo que permite que la aplicación siga siendo fluida mientras los módulos internos procesan la información.

De manera general, la integración se puede entender a partir de los siguientes puntos clave:

- Envío de información estructurada:** la interfaz transmite archivos, identificadores de documento y consultas textuales hacia el backend, estableciendo el contexto sobre el cual trabajan los módulos del simulador.
- Recepción de resultados procesados:** los módulos devuelven información generada a partir del análisis del documento, la cual es interpretada y mostrada dinámicamente en la interfaz.
- Gestión de estados intermedios:** la aplicación controla estados de carga, espera y error, permitiendo que el usuario comprenda qué está ocurriendo en cada momento.

El primer punto concreto de integración se da en el módulo de carga y procesamiento de documentos. En este caso, la interfaz envía el archivo académico seleccionado por el usuario al backend y recibe como respuesta un identificador único de documento. Este identificador se almacena en el estado global de la aplicación y se convierte en la referencia principal para todas las interacciones posteriores. De esta forma, se garantiza que cada acción realizada por el usuario esté asociada siempre al mismo contexto académico.

Una vez establecido este contexto, el espacio de trabajo se integra con los módulos de generación de contenido. Las acciones disponibles en la interfaz, como la generación de resúmenes, preguntas de práctica o planes de estudio, activan servicios específicos del backend utilizando el identificador del documento como parámetro principal. Los resultados obtenidos son renderizados directamente en la interfaz, lo que confirma la correcta conexión entre la capa de presentación y la lógica interna del simulador.

De manera complementaria, el módulo de interacción conversacional también se integra dentro de este esquema. El chat del asistente virtual permite enviar consultas relacionadas con el documento activo, mientras que la interfaz se encarga de mantener el historial de la conversación. El backend, por su parte, utiliza el contexto proporcionado para generar respuestas coherentes, lo que refuerza la sensación de continuidad y personalización en la interacción.

Desde el punto de vista del control del sistema, la integración incluye mecanismos básicos para manejar situaciones como la ausencia de un documento cargado, la falta de resulta-

dos generados o los errores de comunicación. Estos estados son comunicados al usuario mediante mensajes claros e indicadores visuales, evitando comportamientos ambiguos o confusos.

2.3.5.8 5.2 Integración del módulo de generación de clases personalizadas

La integración del módulo de generación de clases personalizadas constituye el núcleo funcional del simulador, ya que es el punto en el que la intención pedagógica del usuario se transforma en un resultado concreto. En esta fase, el ingreso del tema y del material académico deja de ser una acción meramente representativa para convertirse en el disparador real de un proceso de generación de contenido educativo.

Desde la perspectiva de la interfaz, el usuario define el tema de estudio y proporciona el documento que servirá como base contextual. Esta información es enviada al backend de forma estructurada, permitiendo que el módulo de generación acceda tanto al contenido académico como a los parámetros que orientan la construcción de la clase. De esta manera, la interacción inicial del usuario se convierte en una entrada directa para el sistema, estableciendo una relación clara entre lo que se solicita y lo que se genera.

El proceso de integración se apoya en tres elementos fundamentales:

- **Definición del contexto académico:** el documento cargado actúa como fuente principal de información, asegurando que la clase generada mantenga coherencia con el material original.
- **Interpretación de la intención del usuario:** el tema ingresado y los parámetros definidos orientan la forma en que el contenido es organizado y presentado.
- **Construcción de la respuesta educativa:** el módulo combina ambos elementos para producir una clase estructurada y alineada con los objetivos de aprendizaje.

El módulo de generación procesa el contenido previamente indexado y lo articula con la intención expresada por el usuario, dando como resultado una clase que no solo resume información, sino que organiza el conocimiento de forma didáctica. La respuesta generada es devuelta a la interfaz en formato textual, donde es presentada de manera dinámica dentro de la pantalla correspondiente. Este paso permite comprobar que la comunicación entre la interfaz y el módulo funcional se realiza de forma efectiva.

Desde el punto de vista de la experiencia de usuario, la interfaz cumple un rol activo durante este proceso. Se gestionan estados intermedios, como la indicación visual de que la clase está siendo generada, así como la transición hacia la visualización del contenido final. Esto resulta clave para que el usuario perciba el sistema como un entorno en funcionamiento y no como una aplicación que responde de forma abrupta o desarticulada.

Además, esta integración refuerza el flujo pedagógico definido en las etapas previas del proyecto. El ingreso del tema, la selección del material académico y la generación de la clase ya no aparecen como acciones aisladas, sino como partes de un mismo proceso continuo. Esto permite que el simulador empiece a cumplir su propósito central: ofrecer clases personalizadas a partir de un contexto real y de una intención educativa concreta.

2.3.5.9 5.3 Integración del módulo de evaluación

La integración del módulo de evaluación permite cerrar el ciclo de aprendizaje propuesto por el simulador, ya que conecta de forma directa la clase generada con un proceso de verificación del conocimiento adquirido. En este punto, la evaluación deja de entenderse como una pantalla aislada y pasa a formar parte de una secuencia pedagógica continua, en la que cada etapa se apoya en la anterior.

Una vez que el usuario ha interactuado con la clase personalizada, la interfaz habilita el acceso a la evaluación como un paso natural dentro del flujo. Esta decisión responde a la necesidad de que el proceso evaluativo no sea percibido como algo externo, sino como una extensión lógica del contenido estudiado. De esta manera, el simulador refuerza la idea de que aprender y evaluar forman parte de un mismo proceso.

Desde la perspectiva técnica, la interfaz envía al backend la referencia del documento activo y la información de la sesión, permitiendo que el módulo de evaluación recupere el contexto académico utilizado en la generación de la clase. Esto garantiza que los reactivos propuestos estén directamente relacionados con los conceptos, definiciones y ejemplos que el estudiante acaba de revisar, manteniendo coherencia entre contenido y evaluación.

La integración del módulo se apoya principalmente en los siguientes aspectos:

- Uso del mismo contexto académico:** la evaluación se construye a partir del documento y del tema que originaron la clase generada, evitando preguntas genéricas o descontextualizadas.
- Generación de reactivos alineados al contenido:** los elementos evaluativos responden directamente a los conceptos tratados en la clase, reforzando la continuidad

pedagógica.

- **Renderización dinámica en la interfaz:** la interfaz recibe los reactivos estructurados y los presenta de forma ordenada al usuario, permitiendo su interacción inmediata.

Durante la interacción con la evaluación, la interfaz gestiona los estados asociados al proceso, como el avance entre preguntas, el registro de respuestas y la finalización de la actividad. Este manejo de estados resulta clave para que el usuario perciba la evaluación como un proceso fluido y controlado, sin rupturas en la experiencia de uso.

Además, el módulo de evaluación actúa como un puente hacia la etapa de resultados y retroalimentación. Las respuestas proporcionadas por el usuario constituyen la base para el análisis de su desempeño, permitiendo que el sistema genere observaciones y recomendaciones personalizadas en la siguiente fase del simulador. Aunque en esta etapa algunas funcionalidades de análisis avanzado se mantienen de forma parcial o simulada, la estructura de integración entre evaluación y retroalimentación ya se encuentra definida.

2.3.5.10 5.4 Integración del módulo de resultados y retroalimentación

La integración del módulo de resultados y retroalimentación representa el cierre del flujo funcional del simulador, ya que convierte la información obtenida durante la evaluación en elementos comprensibles y útiles para el estudiante. En esta etapa, el sistema deja de enfocarse únicamente en la generación de contenidos y pasa a desempeñar un rol orientador dentro del proceso de aprendizaje, ayudando al usuario a interpretar su desempeño y a reflexionar sobre su progreso.

Una vez finalizada la evaluación, las respuestas del usuario son enviadas al backend para su procesamiento. El sistema consolida esta información considerando el contexto académico previamente definido y el contenido de la clase generada, construyendo así un conjunto de resultados que reflejan el nivel de comprensión alcanzado. Esta consolidación permite que los resultados no sean presentados como valores aislados, sino como una síntesis coherente del proceso de aprendizaje desarrollado por el estudiante.

Desde la interfaz, el módulo de resultados recibe esta información mediante una comunicación directa con el backend y la presenta de forma estructurada en la pantalla correspondiente. La organización visual de los resultados busca facilitar su interpretación, evitando lecturas ambiguas y permitiendo que el usuario identifique rápidamente su desempeño general. En este punto, la interfaz cumple un papel clave como mediadora entre la información técnica generada por el sistema y la comprensión del estudiante.

La integración de este módulo se apoya principalmente en los siguientes aspectos:

- Consolidación de la información evaluativa:** las respuestas del usuario son procesadas en conjunto, permitiendo construir una visión global de su desempeño.
- Presentación estructurada de resultados:** la interfaz organiza la información de forma clara, destacando los elementos más relevantes para la interpretación del estudiante.
- Generación de retroalimentación contextualizada:** los mensajes de retroalimentación se basan en el mismo contenido académico utilizado en la clase y la evaluación.

La retroalimentación generada cumple una función orientadora más que calificativa. En lugar de limitarse a mostrar un resultado estático, el sistema ofrece indicaciones que permiten al estudiante reconocer sus fortalezas y las áreas que requieren mayor atención. Esta orientación se construye a partir del contexto del documento académico y del tema estudiado, manteniendo coherencia con todas las etapas previas del simulador.

Además, este módulo establece un punto de conexión hacia posibles iteraciones futuras dentro del sistema. A partir de los resultados obtenidos, el usuario puede decidir reforzar determinados contenidos, volver a generar material de estudio o iniciar una nueva simulación con parámetros diferentes. Aunque algunas de estas acciones se plantean como extensiones posteriores, la integración actual valida la viabilidad de un ciclo de aprendizaje continuo, donde cada interacción alimenta la siguiente.

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

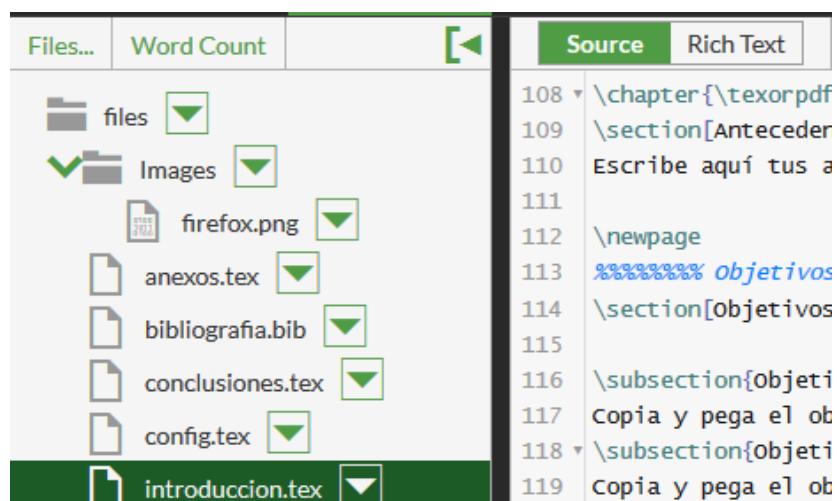


Figura 3.1: Ejemplo imagen

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

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