

# REQUERIMIENTO SOFTWARE DE RUTAS DE APRENDIZAJE PERSONALIZADAS CON IA GENERATIVA Y TUTORES DIGITALES PARA UNIDIGIHUB LATAM

<b>Información general del proyecto.....</b>	<b>3</b>
➤ Nombre del proyecto/solicitud.....	3
➤ Descripción breve.....	3
➤ Objetivo principal.....	3
➤ Justificación.....	3
Beneficios operativos.....	4
Beneficios legales.....	4
Beneficios financieros.....	4
Beneficios sociales.....	5
<b>Detalles técnicos.....</b>	<b>5</b>
➤ Tipo de solicitud.....	5
➤ Sistemas/plataformas impactados.....	5
➤ Arquitectura de la plataforma de aprendizaje adaptativo.....	8
➤ Requerimientos funcionales.....	9
Generar rutas de aprendizaje personalizadas.....	9
Proveer contenido educativo dinámico.....	10
Facilitar el aprendizaje colaborativo.....	10
Ofrecer retroalimentación personalizada.....	10
Garantizar la accesibilidad.....	11
Sistema de medición y evaluación.....	11
1. Monitoreo del progreso.....	11
2. Evaluación del aprendizaje.....	11
3. Indicadores clave de desempeño (KPIs).....	11
➤ Niveles de aprendizaje por cluster y su relación con las rutas de aprendizaje.....	12
➤ Agentes virtuales.....	15
Objetivo de los agentes virtuales.....	15
Características generales de los agentes virtuales.....	15
Interacción con los agentes virtuales.....	16
Especificaciones para cada cluster sectorial.....	16
Arquitectura del software con IA generativa.....	16
Requerimientos no funcionales.....	17
➤ Rendimiento.....	17
➤ Escalabilidad.....	17
➤ Seguridad.....	18
➤ Disponibilidad.....	18
➤ Mantenibilidad.....	18
➤ Integración.....	19

➤ Cumplimiento legal.....	19
➤ Sostenibilidad.....	19
<b>Recursos y plazos.....</b>	<b>20</b>
Recursos disponibles.....	20
Equipo involucrado.....	20
<b>Cronograma.....</b>	<b>21</b>
1. Recopilación de requisitos.....	21
2. Análisis de factibilidad.....	21
3. Diseño del sistema.....	22
4. Desarrollo del software.....	22
5. Pruebas.....	22
6. Implementación.....	22
7. Mantenimiento.....	22
➤ Entregables clave con aprobación del Director de Operaciones.....	24
<b>Presupuesto.....</b>	<b>26</b>
ROI Esperado.....	26
<b>Aprobaciones.....</b>	<b>26</b>

**Área solicitante:** Dirección general

**Fecha de solicitud:** 3/02/2025

**Responsable de la solicitud:** Estefanía Amaya Rojas, directora general

**Prioridad:** Alta

## **Información general del proyecto**

### **> Nombre del proyecto/solicitud**

Software de rutas de aprendizaje personalizadas con ia generativa y tutores digitales para UniDigiHub LATAM

### **> Descripción breve**

Este documento describe los requerimientos para el desarrollo de un sistema de diagnóstico automatizado dentro del LXP (Learning Experience Platform) de UniDigiHub LATAM. El objetivo principal es personalizar la experiencia educativa de cada estudiante, optimizando su trayectoria de aprendizaje y maximizando su potencial. El Diagnóstico Automatizado tiene como objetivo principal evaluar y clasificar a los estudiantes de UniDigiHub LATAM para identificar su nivel de aprendizaje, asignar un sector de interés, fomentar el aprendizaje colaborativo y personalizar la experiencia educativa.

### **> Objetivo principal**

Implementar un software que, utilizando IA Generativa, genere rutas de aprendizaje personalizadas para los estudiantes de UniDigiHub LATAM, considerando su nivel de aprendizaje ("UniExplorador", "UniCreador" o "UniVisionario"), cluster sectorial (AgriTech, FinTech, HealthTech o energías renovables), intereses, habilidades, problemáticas locales y contexto socioeconómico, con el fin de personalizar la experiencia educativa, optimizar el aprendizaje y alinear las capacidades individuales con las necesidades de desarrollo tecnológico sostenible en América Latina. Además, el software debe integrar agentes virtuales como tutores digitales que brinden apoyo y guía a los estudiantes en cada cluster sectorial, aumentando la efectividad del aprendizaje práctico, reduciendo la dependencia de mentores físicos y facilitando la innovación. Estos agentes virtuales deben ser capaces de interactuar de forma conversacional y personalizada con los estudiantes.

### **> Justificación**

Este software permitirá a UniDigiHub LATAM escalar sus operaciones, optimizar el aprendizaje, personalizar la experiencia educativa, reducir costos, aumentar la retención de estudiantes y generar un impacto social positivo, consolidándose como un referente en la formación de talento tecnológico en América Latina.

### *Beneficios operativos*

1. Adaptación a niveles de conocimiento: Los estudiantes reciben rutas de aprendizaje adaptadas a su nivel (UniExplorador, UniCreador, UniVisionario), optimizando su progreso.
2. Contenido relevante: Las rutas se ajustan a los intereses y problemáticas locales, aumentando la motivación y retención de los estudiantes.
3. Automatización de procesos: La IA Generativa reduce la carga de trabajo de los mentores físicos, permitiéndoles enfocarse en tareas estratégicas.
4. Seguimiento en tiempo real: Herramientas de analítica predictiva permiten identificar y apoyar a estudiantes en riesgo de abandonar.
5. Disponibilidad 24/7: Los agentes virtuales brindan apoyo inmediato, reduciendo la dependencia de mentores físicos y facilitando el aprendizaje autónomo.
6. Interacción personalizada: Los tutores digitales ofrecen respuestas y guías adaptadas a las necesidades individuales de cada estudiante.

### *Beneficios legales*

1. Protección de datos: El software cumple con regulaciones como el GDPR (Europa) y la LGPD (Brasil), garantizando la privacidad y seguridad de los datos de los estudiantes.
2. Accesibilidad: Cumple con los estándares WCAG 2.1 AA, asegurando que el contenido sea accesible para personas con discapacidades.
3. Validez internacional: Las certificaciones emitidas por UniDigiHub LATAM están alineadas con estándares globales, aumentando su reconocimiento en el mercado laboral.

### *Beneficios financieros*

1. Menor dependencia de tutores físicos: Los tutores digitales reducen la necesidad de contratar y capacitar tutores, disminuyendo costos a largo plazo.
2. Optimización de recursos: La automatización de procesos educativos y la personalización de rutas minimizan el desperdicio de recursos.
3. Ampliación de la cobertura: El soporte multilingüe y offline permite llegar a comunidades rurales y semiurbanas, ampliando la base de usuarios.
4. Retención de estudiantes: La personalización y el apoyo continuo aumentan la retención, generando ingresos recurrentes.
5. Impacto social: El enfoque en inclusión y desarrollo sostenible atrae inversiones de organizaciones internacionales y gobiernos.
6. Innovación tecnológica: El uso de IA Generativa y tutores digitales posiciona a UniDigiHub LATAM como líder en educación tecnológica en la región.

Beneficios sociales

- 1. Acceso para grupos vulnerables: El software prioriza la inclusión de mujeres, jóvenes y comunidades indígenas, reduciendo brechas de género y socioeconómicas.
- 2. Soporte multilingüe: El contenido en español, portugués, inglés y lenguas indígenas garantiza que nadie quede excluido.
- 3. Alineación con el mercado laboral: Las rutas de aprendizaje se enfocan en sectores estratégicos (AgriTech, FinTech, HealthTech, Energías Renovables), preparando a los estudiantes para empleos de alta demanda.
- 4. Fomento de la innovación: Los proyectos prácticos y la guía de tutores digitales impulsan la creación de soluciones tecnológicas locales.
- 5. Soluciones a problemáticas reales: Los estudiantes desarrollan proyectos que abordan desafíos locales, como la falta de acceso a agua potable o energía renovable.
- 6. Empoderamiento comunitario: La formación en tecnologías emergentes permite a las comunidades ser autosuficientes y competitivas en la economía digital.

Detalles técnicos

- Tipo de solicitud
  - ☒ Nuevo desarrollo
  - ☐ Mejora o ampliación de sistema existente
  - ☐ Corrección de error (Bug)
  - ☐ Integración con otros sistemas
  - ☐ Otro:
- Sistemas/plataformas impactados

Sistemas			
Nombre	Tipo	Descripción	Impacto/Integración
Sistema de Diagnóstico Inteligente	Módulo de IA	Clasifica a los estudiantes en niveles (UniExplorador, UniCreador, UniVisionario) y clusters sectoriales (AgriTech, FinTech, etc.).	Provee datos clave para generar rutas de aprendizaje personalizadas.
Plataforma Educativa (LXP)	Sistema principal	Plataforma de aprendizaje de UniDigiHub LATAM.	Integra las rutas de aprendizaje y tutores digitales como módulos funcionales.

Sistema de Rutas de Aprendizaje	Módulo central	Genera y gestiona rutas adaptativas basadas en IA Generativa.	Interactúa con el diagnóstico y los tutores digitales.
Sistema de Tutores Digitales	Módulo de IA Conversacional	Agentes virtuales especializados por sector (Aylen, Kuntur, Nayra, Inti).	Brinda soporte personalizado dentro de las rutas de aprendizaje.
<b>Bases de datos</b>			
<i>Nombre</i>	<i>Tipo</i>	<i>Descripción</i>	<i>Impacto/Integración</i>
PostgreSQL	Base de datos relacional	Almacena perfiles de estudiantes, cursos, progreso y metadatos de rutas.	Fuente principal para datos estructurados del sistema.
MongoDB	Base de datos NoSQL	Guarda logs de interacción con tutores digitales, respuestas de texto libre y feedback cualitativo.	Usada para análisis de NLP y mejora continua de las rutas.
Redis	Base de datos en memoria	Cachea resultados de diagnósticos y recomendaciones para reducir latencia.	Optimiza el rendimiento en consultas recurrentes.
Google Cloud Storage (GCS)	Almacenamiento en la nube	Aloja modelos de IA, datasets de entrenamiento y materiales educativos (videos, PDFs).	Clave para el pipeline de IA y acceso offline.
<b>APIs</b>			
<i>Nombre</i>	<i>Tipo</i>	<i>Descripción</i>	<i>Endpoints Clave</i>
Diagnóstico API	REST (FastAPI)	Obtiene datos del perfil del estudiante (nivel, cluster, habilidades).	/api/v1/profile/{user_id} (GET).
Tutoring API	REST (Node.js)	Gestiona interacciones con los tutores digitales (Aylen, Kuntur, etc.).	/api/v1/tutor/ask(P OST: envía preguntas).
HubSpot CRM API	REST	Sincroniza progreso y métricas para campañas personalizadas.	/crm/v3/objects/co ntacts (POST: actualiza datos).

Google Cloud Translation API	Cloud API	Traduce contenido educativo a múltiples idiomas (español, portugués, quechua, etc.).	/v3/translate (POST).
Google Natural Language API	Cloud API	Analiza texto para identificar intereses y problemáticas locales.	/v1/documents:analyzeEntities (POST).
LXP Integration API	REST	Conecta el sistema de rutas con la plataforma educativa principal.	/api/v1/sync-courses (POST).

### Herramientas y Frameworks

Nombre	Tipo	Descripción	Uso en el Proyecto
React.js + TypeScript	Frontend	Desarrollo de la interfaz de usuario (formularios, dashboard).	Creación de una interfaz responsive y accesible.
Python (FastAPI)	Backend	Construcción de microservicios para IA y gestión de rutas.	Desarrollo del motor de recomendación y analítica.
TensorFlow/PyTorch	Machine Learning	Entrenamiento de modelos de clasificación y predicción.	Modelado de niveles de aprendizaje y riesgo de deserción.
Hugging Face Transformers	NLP	Procesamiento de texto para análisis de intereses y feedback.	Integración con modelos como Sentence-BERT.
Kubernetes + Docker	Orquestación	Gestión de contenedores y escalabilidad en GCP.	Despliegue de microservicios y modelos de IA.
Apache Kafka	Stream Processing	Ingesta de datos en tiempo real desde formularios y tests.	Canalización de interacciones para análisis inmediato.
MLflow	MLOps	Gestión del ciclo de vida de modelos de IA.	Registro de experimentos y despliegue de modelos.

### Herramientas de Monitoreo y Seguridad

Nombre	Tipo	Descripción	Uso en el Proyecto
Prometheus +	Monitoreo	Supervisión de métricas de	Alertas sobre

Grafana		rendimiento y salud del sistema.	latencia o fallos en tiempo real.
ELK Stack (Elasticsearch, Logstash, Kibana)	Logging	Centralización y análisis de logs.	Diagnóstico de errores en interacciones con tutores.
Google Cloud KMS	Seguridad	Gestión de claves de encriptación para datos sensibles.	Protección de información de estudiantes y modelos.
Auth0	Autenticación	Gestión de identidad y acceso (SSO, 2FA).	Seguridad en el acceso a la plataforma.
<b>Impacto en Herramientas Externas</b>			
Herramienta	Impacto		
HubSpot CRM	Sincronización de datos de estudiantes para segmentación y marketing educativo.		
Google Cloud Platform (GCP)	Uso intensivo de servicios como BigQuery (análisis), Cloud Storage (almacenamiento) y Kubernetes (escalabilidad).		
Plataformas de Comunicación(ej: WhatsApp Business API y MSN)	Envío de notificaciones y recordatorios a estudiantes.		

➤ Arquitectura de la plataforma de aprendizaje adaptativo

La plataforma se basa en una arquitectura modular que integra diferentes sistemas y tecnologías para ofrecer una experiencia de aprendizaje personalizada y adaptativa. Los componentes principales son:

1. Sistema de diagnóstico inteligente: Clasifica a los estudiantes en niveles de aprendizaje ("UniExplorador", "UniCreador", "UniVisionario") y clusters sectoriales



(AgriTech, FinTech, HealthTech, Energías Renovables). Provee datos clave para la personalización.

2. Plataforma educativa (LXP): Plataforma principal de aprendizaje de UniDigiHub LATAM. Integra las rutas de aprendizaje y los tutores digitales.
3. Sistema de rutas de aprendizaje: Genera y gestiona rutas adaptativas basadas en IA Generativa. Interactúa con el diagnóstico y los tutores digitales.
4. Sistema de tutores digitales: Agentes virtuales especializados por sector (Aylen, Kuntur, Nayra, Inti). Brindan soporte personalizado dentro de las rutas de aprendizaje.

#### ➤ Requerimientos funcionales

Para cumplir con el objetivo principal, el software debe ser capaz de realizar las siguientes funciones:

##### *Generar rutas de aprendizaje personalizadas*

1. Integrarse con el Sistema de Diagnóstico Inteligente: El software debe acceder a la información del perfil del estudiante generada por el Sistema de Diagnóstico Inteligente. Esto incluye el nivel de aprendizaje ("UniExplorador", "UniCreador" o "UniVisionario"), el cluster sectorial (AgriTech, FinTech, HealthTech o energías renovables), los intereses, las habilidades, las problemáticas locales que les interese resolver y el contexto socioeconómico.
2. Utilizar algoritmos de IA Generativa: El software debe utilizar algoritmos de IA Generativa para crear secuencias de aprendizaje óptimas para cada estudiante. Estos algoritmos deben considerar la información del perfil del estudiante, sus objetivos de aprendizaje, sus preferencias de aprendizaje (ej: formato de contenido preferido, ritmo de aprendizaje) y la disponibilidad de recursos educativos.
3. Adaptar el contenido y la dificultad: El software debe adaptar el contenido y la dificultad de las rutas de aprendizaje al nivel de aprendizaje del estudiante. Por ejemplo, un "UniExplorador" en AgriTech recibirá contenido introductorio sobre agricultura y tecnología, mientras que un "UniVisionario" en el mismo cluster recibirá contenido más avanzado y especializado.
4. Considerar los intereses y las problemáticas locales: El software debe considerar los intereses del estudiante y las problemáticas locales relevantes al cluster sectorial al generar las rutas de aprendizaje. Por ejemplo, si un estudiante está interesado en la agricultura sostenible y pertenece al cluster AgriTech, el software debe generar una ruta de aprendizaje que incluya cursos y proyectos relacionados con este tema.
5. Ofrecer rutas de aprendizaje en múltiples idiomas: El software debe ofrecer rutas de aprendizaje en español, portugués, inglés y lenguas indígenas, para asegurar la accesibilidad a estudiantes de diferentes regiones y con diversas necesidades lingüísticas.

### *Proveer contenido educativo dinámico*

1. Generar contenido personalizado: El software debe utilizar IA Generativa para crear contenido educativo personalizado, como textos, ejercicios, ejemplos y evaluaciones. Este contenido debe adaptarse al perfil del estudiante, su nivel de aprendizaje y sus preferencias.
2. Adaptar el formato del contenido: El software debe ofrecer el contenido en diferentes formatos (texto, video, audio, simulaciones interactivas, etc.) según las preferencias de aprendizaje del estudiante.
3. Actualizar el contenido de forma dinámica: El software debe actualizar el contenido de forma dinámica, incorporando nueva información y recursos relevantes. Esto asegura que las rutas de aprendizaje se mantengan actualizadas con las últimas tendencias y avances en cada sector.
4. Conexión con microlearning y gamificación: Las rutas de aprendizaje personalizadas se benefician de la integración con el microlearning y la gamificación. El microlearning, al dividir el contenido en segmentos más pequeños, facilita la asimilación de la información y se adapta al ritmo individual de cada estudiante. La gamificación, por su parte, introduce elementos de juego que motivan a los estudiantes y hacen que el aprendizaje sea más atractivo.

### *Facilitar el aprendizaje colaborativo*

1. Integrar herramientas de comunicación: El software debe integrar herramientas de comunicación y colaboración, como foros de discusión, chats, videoconferencias y plataformas de trabajo en equipo, para facilitar la interacción entre estudiantes.
2. Formar grupos de estudio: El software debe utilizar la información del perfil del estudiante para formar grupos de estudio con estudiantes de perfiles complementarios. Esto fomenta la diversidad y el aprendizaje entre pares.
3. Proponer proyectos colaborativos: El software debe proponer proyectos colaborativos que permitan a los estudiantes aplicar sus conocimientos a la resolución de problemas reales. Esto fomenta el trabajo en equipo y la aplicación práctica del aprendizaje.

### *Ofrecer retroalimentación personalizada*

1. Utilizar IA Generativa: El software debe utilizar IA Generativa para proporcionar retroalimentación específica y relevante sobre el desempeño del estudiante en las rutas de aprendizaje. Esta retroalimentación debe identificar áreas de mejora y ofrecer sugerencias.
2. Adaptar el tono y el estilo: El software debe adaptar el tono y el estilo de la retroalimentación al nivel de aprendizaje del estudiante. Por ejemplo, la retroalimentación para un "UniExplorador" debe ser más alentadora y motivadora, mientras que la retroalimentación para un "UniVisionario" puede ser más desafiante y crítica.

3. Ofrecer sugerencias de mejora: El software debe ofrecer sugerencias de mejora y recursos adicionales, como tutoriales, documentación o enlaces a otros materiales, para apoyar el aprendizaje del estudiante.

#### *Garantizar la accesibilidad*

1. Permitir el acceso offline: El software debe permitir el acceso offline a las rutas de aprendizaje y al contenido educativo. Esto es especialmente importante para estudiantes en zonas rurales con acceso limitado a internet.
2. Cumplir con los estándares de accesibilidad: El software debe cumplir con los estándares de accesibilidad WCAG 2.1 para asegurar la inclusión de estudiantes con discapacidades. Esto incluye funciones como la compatibilidad con lectores de pantalla, el alto contraste visual y las transcripciones interactivas en videos.

#### *Sistema de medición y evaluación*

1. Monitoreo del progreso
  - a. Registrar y analizar el progreso del estudiante en cada ruta de aprendizaje.
  - b. Monitorear la participación en actividades clave (bootcamps, prácticas, validaciones).
  - c. Identificar patrones de aprendizaje y prever riesgos de deserción.
  - d. Plataforma analítica integrada: Registrar datos en tiempo real sobre el progreso, la participación en actividades y el uso de la plataforma.
  - e. Supervisión en los CTH: Registrar la asistencia, la interacción y los resultados en actividades prácticas.
2. Evaluación del aprendizaje
  - a. Evaluaciones formativas: Aplicar evaluaciones continuas en cada etapa de las rutas de aprendizaje para medir el progreso y ofrecer retroalimentación.
  - b. Validación comunitaria: Involucrar a las comunidades en la validación de prototipos y MVPs para evaluar el impacto de las soluciones.
  - c. Encuestas de satisfacción: Aplicar encuestas al finalizar cada etapa para medir la experiencia del estudiante y la percepción de impacto.
3. Indicadores clave de desempeño (KPIs)
  - a. Educativos: Tasa de finalización de rutas de aprendizaje, progreso en competencias y certificación en tecnologías emergentes.
  - b. Impacto comunitario: Número de problemas locales identificados y soluciones implementadas.
  - c. Conexión con el entorno laboral: Número de estudiantes que obtienen empleo o inician emprendimientos.

➤ Niveles de aprendizaje por cluster y su relación con las rutas de aprendizaje

<b>Cluster Sectorial</b>	<b>Nivel de Aprendizaje</b>	<b>Ruta de Aprendizaje 1: Comprensión del problema (1 mes - 76 horas)</b>	<b>Ruta de Aprendizaje 2: Ideación y prototipado de soluciones (4 meses - 76 horas)</b>	<b>Ruta de Aprendizaje 3: Construcción y prueba del MVP (4 meses - 76 horas)</b>
AgriTech	UniExplorador	Identificar un problema básico relacionado con la agricultura. Formular preguntas de investigación sencillas. Utilizar herramientas de análisis básicas.	Participar en la ideación de soluciones. Utilizar herramientas de prototipado básicas. Presentar la solución de forma clara.	Participar en la construcción del MVP. Realizar pruebas básicas. Comprender el proceso de desarrollo.
	UniCreador	Analizar el problema en profundidad. Proponer soluciones creativas. Utilizar herramientas de análisis de datos.	Liderar la ideación de soluciones. Diseñar prototipos funcionales. Aplicar conocimientos técnicos del cluster.	Contribuir al desarrollo del MVP. Realizar pruebas de usuario. Proponer mejoras.
	UniVisionario	Identificar un problema complejo. Analizarlo desde diferentes perspectivas. Formular preguntas de investigación innovadoras.	Proponer soluciones innovadoras. Diseñar prototipos complejos. Liderar la validación comunitaria.	Liderar el desarrollo del MVP. Integrar diferentes tecnologías. Presentar el proyecto de forma convincente.
FinTech	UniExplorador	Identificar un problema básico relacionado con las finanzas. Comprender los conceptos básicos	Participar en la ideación de soluciones Fintech. Crear prototipos de aplicaciones Fintech sencillas.	Participar en la construcción de un MVP Fintech. Realizar pruebas básicas de la aplicación.

		de Fintech. Utilizar herramientas de análisis financiero básicas.	Presentar la solución de forma clara.	Comprender el proceso de desarrollo.
	UniCreador	Analizar el problema financiero en profundidad. Proponer soluciones Fintech creativas. Utilizar herramientas de análisis financiero avanzadas.	Liderar la ideación de soluciones Fintech. Diseñar prototipos de aplicaciones Fintech funcionales. Aplicar conocimientos técnicos del cluster.	Contribuir al desarrollo del MVP Fintech. Realizar pruebas de usuario de la aplicación. Proponer mejoras.
	UniVisionario	Identificar un problema financiero complejo. Analizarlo desde diferentes perspectivas. Formular preguntas de investigación innovadoras en Fintech.	Proponer soluciones Fintech innovadoras. Diseñar prototipos de aplicaciones Fintech complejas. Liderar la validación comunitaria.	Liderar el desarrollo del MVP Fintech. Integrar diferentes tecnologías Fintech. Presentar el proyecto de forma convincente.
HealthTech	UniExplorador	Identificar un problema básico relacionado con la salud. Comprender los conceptos básicos de HealthTech. Utilizar herramientas de análisis de datos de salud básicas.	Participar en la ideación de soluciones HealthTech. Crear prototipos de aplicaciones HealthTech sencillas. Presentar la solución de forma clara.	Participar en la construcción de un MVP HealthTech. Realizar pruebas básicas de la aplicación. Comprender el proceso de desarrollo.
	UniCreador	Analizar el problema de salud en profundidad. Proponer soluciones HealthTech creativas.	Liderar la ideación de soluciones HealthTech. Diseñar prototipos de aplicaciones HealthTech	Contribuir al desarrollo del MVP HealthTech. Realizar pruebas de usuario de la aplicación.

		Utilizar herramientas de análisis de datos de salud avanzadas.	funcionales. Aplicar conocimientos técnicos del cluster.	Proponer mejoras.
	UniVisionario	Identificar un problema de salud complejo. Analizarlo desde diferentes perspectivas. Formular preguntas de investigación innovadoras en HealthTech.	Proponer soluciones HealthTech innovadoras. Diseñar prototipos de aplicaciones HealthTech complejas. Liderar la validación comunitaria.	Liderar el desarrollo del MVP HealthTech. Integrar diferentes tecnologías HealthTech. Presentar el proyecto de forma convincente.
Energías Renovables	UniExplorador	Identificar un problema básico relacionado con las energías renovables. Comprender los conceptos básicos de las energías renovables. Utilizar herramientas de análisis de datos energéticos básicas.	Participar en la ideación de soluciones con energías renovables. Crear prototipos de sistemas de energía renovable sencillos. Presentar la solución de forma clara.	Participar en la construcción de un MVP con energías renovables. Realizar pruebas básicas del sistema. Comprender el proceso de desarrollo.
	UniCreador	Analizar el problema energético en profundidad. Proponer soluciones con energías renovables creativas. Utilizar herramientas de análisis de datos energéticos avanzadas.	Liderar la ideación de soluciones con energías renovables. Diseñar prototipos de sistemas de energía renovable funcionales. Aplicar conocimientos técnicos del cluster.	Contribuir al desarrollo del MVP con energías renovables. Realizar pruebas de usuario del sistema. Proponer mejoras.

	UniVisionario	Identificar un problema energético complejo. Analizarlo desde diferentes perspectivas. Formular preguntas de investigación innovadoras en energías renovables.	Proponer soluciones con energías renovables innovadoras. Diseñar prototipos de sistemas de energía renovable complejos. Liderar la validación comunitaria.	Liderar el desarrollo del MVP con energías renovables. Integrar diferentes tecnologías de energías renovables. Presentar el proyecto de forma convincente.
--	---------------	--	--	--

### ➤ Agentes virtuales

Los agentes virtuales serán un componente central del software, funcionando como tutores digitales que brindan apoyo y guía a los estudiantes en cada cluster sectorial.

#### *Objetivo de los agentes virtuales*

1. Aumentar la efectividad del aprendizaje práctico en cada cluster.
2. Reducir la dependencia exclusiva de tutores o especialistas físicos, permitiendo una mayor escalabilidad del modelo.
3. Facilitar la innovación y la resolución de problemas locales mediante tecnologías avanzadas.

#### *Características generales de los agentes virtuales*

1. Interactividad y personalización: Se adaptarán al nivel de habilidad, ritmo de aprendizaje y sector de interés de cada participante. Podrán interactuar mediante chatbots, asistentes de voz o interfaces gráficas en tiempo real.
2. Conocimiento especializado: Tendrán un conocimiento profundo del sector al que están asignados (AgriTech, FinTech, HealthTech o energías renovables).
3. Multiplataforma: Accesibles desde cualquier dispositivo (tablets, laptops, smartphones) dentro y fuera del CTH. Conectados a la nube para sincronizar avances y recursos educativos.
4. Soporte Multilingüe: Disponibles en idiomas locales, español, inglés, portugués y lenguas indígenas si es necesario.
5. Inteligencia artificial: Capacidad para resolver preguntas frecuentes, guiar en procesos específicos y ofrecer sugerencias personalizadas para optimizar el aprendizaje. Basados en modelos preentrenados en tecnologías emergentes y diseñados para actualizarse constantemente con nuevas tendencias.
6. Disponibilidad 24/7: Disponibles en todo momento para responder preguntas, brindar apoyo y guiar a los estudiantes.

7. Análisis del progreso: Los asistentes virtuales analizan el progreso del estudiante en las rutas de aprendizaje, identificando áreas de fortaleza y debilidad.
8. Recomendación de recursos: Sugieren recursos adicionales, como videos, artículos o ejercicios interactivos, que complementan el aprendizaje y permiten profundizar en los temas de interés.
9. Respuestas a consultas: Responden a las preguntas de los estudiantes de forma inmediata, brindando aclaraciones y resolviendo dudas sobre el contenido del curso.
10. Retroalimentación personalizada: Proporcionarán retroalimentación específica sobre el progreso de los estudiantes, identificando áreas de mejora y ofreciendo sugerencias.
11. Motivación constante: Animarán a los estudiantes a continuar aprendiendo, a superar los desafíos y a alcanzar sus objetivos.

### *Interacción con los agentes virtuales*

Los estudiantes interactuarán con los agentes virtuales a través de

1. Pantallas interactivas: En cada cluster sectorial en los Hub Comunitario de Tecnología y Habilidades (CTH).
2. Plataforma de aprendizaje online: Integrados en UniDigiHub LATAM.
3. Dispositivos móviles: A través de una aplicación móvil.

### *Especificaciones para cada cluster sectorial*

Se definen las características específicas de cada agente virtual, incluyendo su nombre, diseño y funciones principales:

1. AgriTech: "Aylen"
2. FinTech: "Kuntur"
3. HealthTech: "Nayra"
4. Energías renovables: "Inti"

### *Arquitectura del software con IA generativa*

Para lograr la interacción conversacional y la personalización de las respuestas de los agentes virtuales, se requiere una arquitectura robusta basada en IA, que incluya:

1. Modelos de Machine Learning (ML) y Deep Learning (DL): Se utilizarán modelos avanzados de ML y DL, como Procesamiento de Lenguaje Natural (NLP) para la interacción conversacional y Redes Neuronales Recurrentes (RNN) o Transformers para la personalización de las respuestas.
2. Infraestructura de alto rendimiento: Se necesitarán servidores optimizados con GPUs de alto rendimiento, como NVIDIA A100 o RTX 4090, para garantizar un procesamiento eficiente de datos en tiempo real.
3. Frameworks de IA: Se recomienda el uso de frameworks como TensorFlow, PyTorch o OpenAI GPT para el desarrollo de modelos de IA generativa y sistemas de recomendación adaptativos.



4. Backend robusto: La integración con laboratorios y plataformas educativas del CTH requerirá un backend robusto basado en microservicios con tecnologías como Docker, Kubernetes y bases de datos escalables tipo MongoDB o PostgreSQL.
5. Interfaz interactiva: La implementación de una interfaz interactiva en entornos WebRTC, realidad aumentada (AR) o realidad virtual (VR) mejorará la experiencia de aprendizaje autónomo y el desarrollo de proyectos tecnológicos.

## **Requerimientos no funcionales**

### **➤ Rendimiento**

1. Tiempo de respuesta: El sistema debe ser capaz de generar rutas de aprendizaje personalizadas en un tiempo máximo de 5 minutos. El tiempo de respuesta de la plataforma debe ser inferior a 3 segundos para la mayoría de las interacciones, como la navegación entre pantallas, la carga de contenido y la interacción con los agentes virtuales.
2. Capacidad de carga: El sistema debe ser capaz de manejar un alto volumen de usuarios concurrentes (por ejemplo, 1000 estudiantes accediendo simultáneamente) sin degradación del rendimiento. Esto implica que la plataforma debe ser capaz de procesar un gran número de solicitudes, interacciones y operaciones de forma eficiente, sin afectar la velocidad de respuesta ni la estabilidad del sistema.
3. Eficiencia de los algoritmos de IA: Los algoritmos de IA Generativa deben ser optimizados para garantizar un procesamiento eficiente de los datos y la generación de respuestas en tiempo real. Esto puede implicar el uso de técnicas de optimización de modelos, la selección de algoritmos eficientes y la implementación de estrategias de caching para acelerar el procesamiento.

### **➤ Escalabilidad**

1. Escalabilidad horizontal: El sistema debe ser escalable horizontalmente, lo que significa que se puede aumentar la capacidad del sistema añadiendo más servidores o recursos de hardware. Esto permite que la plataforma se adapte al crecimiento de UniDigiHub LATAM y al aumento en el número de estudiantes y recursos.
2. Escalabilidad de la base de datos: Las bases de datos deben ser escalables para manejar un volumen creciente de datos. Esto puede implicar el uso de bases de datos distribuidas, la implementación de técnicas de sharding o la optimización de las consultas para mejorar el rendimiento.
3. Escalabilidad de los modelos de IA: Los modelos de IA deben ser escalables para manejar un mayor volumen de datos y un mayor número de usuarios. Esto puede implicar el uso de técnicas de aprendizaje distribuido, la optimización de los modelos para la inferencia en tiempo real y la implementación de estrategias de escalado automático.

## ➤ Seguridad

1. Autenticación y autorización: Se deben implementar mecanismos de autenticación robustos para verificar la identidad de los usuarios y controlar el acceso a la plataforma. Esto puede incluir el uso de contraseñas seguras, la autenticación multifactor y la integración con sistemas de gestión de identidad.
2. Encriptación de datos: Los datos sensibles, como la información personal de los estudiantes y los datos de las evaluaciones, deben ser encriptados tanto en tránsito como en reposo. Esto protege la información de accesos no autorizados y garantiza la confidencialidad de los datos.
3. Protección contra ataques: El sistema debe estar protegido contra ataques cibernéticos, como la inyección de SQL, el cross-site scripting (XSS) y la denegación de servicio (DoS). Esto implica la implementación de medidas de seguridad como firewalls, sistemas de detección de intrusos y mecanismos de protección contra malware.
4. Cumplimiento de normas de seguridad: El sistema debe cumplir con las normas de seguridad relevantes, como ISO 27001 y SOC 2, para garantizar la seguridad de la información y la protección de los datos.

## ➤ Disponibilidad

1. Interfaz intuitiva: La interfaz de usuario debe ser intuitiva y fácil de usar para estudiantes de diferentes niveles de habilidad tecnológica. Esto implica el uso de un diseño claro y sencillo, la organización lógica de la información y la implementación de funciones de ayuda y tutoriales.
2. Navegación sencilla: La navegación por la plataforma debe ser sencilla e intuitiva, permitiendo a los estudiantes encontrar fácilmente la información y los recursos que necesitan. Esto puede implicar el uso de menús claros, un sistema de búsqueda eficiente y la implementación de breadcrumbs para facilitar la navegación.
3. Compatibilidad con dispositivos: El software debe ser compatible con diferentes dispositivos, como ordenadores de escritorio, portátiles, tablets y smartphones. La interfaz de usuario debe adaptarse a diferentes tamaños de pantalla y resoluciones para garantizar una experiencia óptima en todos los dispositivos.
4. Accesibilidad para personas con discapacidades: El sistema debe ser accesible para estudiantes con discapacidades, cumpliendo con los estándares WCAG 2.1. Esto incluye funciones como la compatibilidad con lectores de pantalla, el alto contraste visual, la posibilidad de ajustar el tamaño del texto y la implementación de subtítulos en videos.

## ➤ Mantenibilidad

1. Código modular: El código del sistema debe ser modular y bien documentado para facilitar el mantenimiento y las actualizaciones. Esto implica la división del código en componentes independientes, la utilización de convenciones de codificación claras y la creación de documentación técnica completa.

2. Sistema de registro y monitoreo: Se debe implementar un sistema de registro y monitoreo para facilitar la detección y resolución de errores. Esto permite a los desarrolladores identificar y solucionar problemas de forma rápida y eficiente, mejorando la estabilidad y el rendimiento del sistema.
3. Facilidad de actualización: El sistema debe ser diseñado para facilitar la implementación de actualizaciones y nuevas funcionalidades. Esto puede implicar el uso de arquitecturas modulares, la implementación de sistemas de control de versiones y la automatización de procesos de despliegue.

➤ Integración

1. Integración con sistemas existentes: El sistema debe integrarse con el Sistema de Diagnóstico Inteligente, el LXP de UniDigiHub LATAM y otras plataformas relevantes. Esto implica la implementación de interfaces de programación de aplicaciones (APIs) que permitan la comunicación entre los diferentes sistemas.
2. Interoperabilidad: La integración debe ser fluida y eficiente, garantizando la interoperabilidad entre los diferentes sistemas. Esto implica el uso de estándares de comunicación y protocolos que permitan el intercambio de datos entre las plataformas.

➤ Cumplimiento legal

1. Protección de datos: El sistema debe cumplir con todas las leyes y regulaciones relevantes, incluyendo las leyes de protección de datos como GDPR y LGPD. Esto implica la implementación de medidas de seguridad para proteger la información personal de los estudiantes y garantizar el cumplimiento de los derechos de los usuarios.
2. Accesibilidad: El sistema debe cumplir con las normas de accesibilidad, como WCAG 2.1, para garantizar que sea accesible para estudiantes con discapacidades.
3. Propiedad intelectual: El software debe cumplir con las leyes de propiedad intelectual, asegurando que no se infrinjan los derechos de autor ni patentes de terceros.

➤ Sostenibilidad

1. Eficiencia energética: El sistema debe ser diseñado para minimizar el impacto ambiental, utilizando tecnologías y prácticas sostenibles. Esto puede implicar la optimización del consumo energético de los servidores, la utilización de energías renovables y la implementación de prácticas de reciclaje.
2. Reducción de residuos: El sistema debe ser diseñado para minimizar la generación de residuos, promoviendo el uso de materiales reciclados y la reutilización de componentes.
3. Responsabilidad social: El desarrollo del software debe considerar aspectos de responsabilidad social, como la promoción de la inclusión, la diversidad y la equidad.

Recursos y plazos

Fecha límite de entrega: 30/04/2025

Recursos disponibles

- ☒ Documentación técnica
- ☒ Acceso a sistemas
- ☐ Equipo de soporte
- ☐ Presupuesto aprobado: \$ USD

Equipo involucrado

Roles clave y cantidad de personas			
Rol	Cantidad	Responsabilidades	Dependencia
Director de Operaciones	1	Supervisa el proyecto, asegura la alineación con los objetivos estratégicos de UniDigiHub LATAM.	Reporta a la Directora General
Gerente de Tecnología y Educación	1	Responsable del proyecto, toma de decisiones, gestión de recursos, comunicación con stakeholders.	Reporta al Director de Operaciones.
Líder Técnico	1	Define la arquitectura del sistema, supervisa el desarrollo, asegura la calidad del código.	Reporta al Gerente de Tecnología y Educación.
Ingeniero de Machine Learning	2	Desarrollo e implementación de los modelos de Machine Learning.	Trabaja bajo el Líder Técnico.
Desarrollador Full-Stack	3	Desarrollo del frontend y backend del software, integración con las APIs.	Coordinación con Líder Técnico.
Especialista en Datos	1	Gestión de las bases de datos, análisis de datos, extracción de información.	Apoya a Ingenieros de ML.
Expertos en Educación y Diseño Instruccional	2	Diseño de las rutas de aprendizaje, selección de contenido, definición de criterios de evaluación.	Trabajan con Gerente de Tecnología y Educación.

Diseñador UX/UI	1	Diseño de la interfaz de usuario, asegurando la usabilidad y la accesibilidad.	Colabora con Desarrolladores.
QA/Testing	1	Pruebas del software, identificación y reporte de errores.	Trabaja con Desarrolladores.
Scrum Master	1	Facilita la implementación de metodologías ágiles, gestiona el trabajo del equipo de desarrollo.	Reporta al Gerente de Tecnología y Educación.

Estructura del equipo (Total: 14 personas)

- Equipo Técnico: 7 personas (Líder Técnico + 2 ML + 3 Full-Stack + 1 Datos).
- Equipo Pedagógico: 2 personas (Expertos en Educación y Diseño Instruccional).
- Diseño y Calidad: 2 personas (Diseñador UX/UI + QA).
- Gestión: 3 personas (Director + Gerente + Scrum Master).

## Cronograma

El desarrollo de este software requiere una planificación meticulosa que abarque las diferentes etapas del proceso, desde la conceptualización hasta la implementación y el mantenimiento. Para construir un sistema de IA robusto, es fundamental considerar la calidad de los datos, el almacenamiento, la gobernanza, la arquitectura y la privacidad. A continuación, se describe el proceso de planificación y diseño:

### 1. Recopilación de requisitos

En esta etapa inicial, se definen las necesidades y expectativas de los usuarios finales, que en este caso son los estudiantes y profesores de UniDigiHub LATAM. Se busca comprender cómo se utilizará el software en el mundo real, qué características son esenciales para su éxito y documentar las expectativas y preocupaciones de las partes interesadas. Algunas preguntas clave a considerar son:

- ¿Cuáles son los objetivos de aprendizaje de los estudiantes?
- ¿Cómo se evaluará el progreso de los estudiantes?
- ¿Qué tipo de contenido educativo se integrará en el software?
- ¿Cómo se adaptará el software a las necesidades individuales de cada estudiante?
- ¿Cómo se garantizará la seguridad y privacidad de los datos de los estudiantes?

### 2. Análisis de factibilidad

Una vez recopilados los requisitos, se realiza un análisis de factibilidad para evaluar si el proyecto es viable desde una perspectiva técnica, económica y operativa. Se consideran los

recursos disponibles, el presupuesto, el tiempo de desarrollo y los posibles riesgos. Este análisis ayuda a determinar si el software puede construirse dentro de las limitaciones existentes y si cumplirá con las necesidades de los usuarios.

3. Diseño del sistema

En esta etapa, se define la arquitectura técnica del software, incluyendo la estructura general, las tecnologías a utilizar, el diseño de la interfaz de usuario, los modelos de bases de datos y el flujo de datos dentro del sistema. Un diseño detallado del sistema garantiza que todos los componentes funcionen de manera eficiente y armoniosa. Además, se identifican y gestionan las dependencias entre las diferentes etapas del desarrollo del software para asegurar una ejecución fluida del proyecto.

4. Desarrollo del software

La fase de desarrollo implica la codificación del software según las especificaciones de diseño. Los desarrolladores escriben el código, implementan las funciones requeridas y se aseguran de que el software funcione como se espera. Esta fase suele ser la más intensiva en términos de tiempo y recursos.

5. Pruebas

Una vez escrito el código, el software se somete a rigurosas pruebas para identificar y corregir cualquier error o problema. Las pruebas pueden incluir pruebas unitarias, pruebas de integración, pruebas de sistema y pruebas de aceptación del usuario, entre otras. Esta fase es crucial para garantizar la calidad, confiabilidad y rendimiento del software.

6. Implementación

Después de un desarrollo y pruebas exitosos, el software se implementa en un entorno real. Esto implica configurar servidores, establecer bases de datos y garantizar que el software funcione correctamente en el entorno de destino.

7. Mantenimiento

Una vez implementado, el software requiere mantenimiento continuo para abordar problemas, aplicar actualizaciones y agregar nuevas funciones según sea necesario. El mantenimiento regular garantiza que el software siga funcionando de manera efectiva y satisfaga las necesidades cambiantes de los usuarios.

Fase 1: Planificación y Diseño (3 de febrero - 28 de febrero 2025)					
Hito	Tareas	Responsable	Fecha Inicio	Fecha Fin	Entregable

Definición de Requerimientos	Definición de requerimientos	Gerente de Tecnología y Educación	3 de febrero de 2025	10 de febrero de 2025	Documento de requerimientos
Diseño del Formulario de Solicitud	Formulario de Solicitud de Desarrollo de Software	Gerente de Tecnología y Educación	3 de febrero de 2025	10 de febrero de 2025	Formulario de solicitud
Arquitectura del Sistema	Diseño de la arquitectura del sistema	Líder Técnico	11 de febrero de 2025	17 de febrero de 2025	Documento de arquitectura del sistema
Diseño de la Interfaz	Diseño de la interfaz de usuario	Diseñador UX/UI	18 de febrero de 2025	25 de febrero de 2025	Prototipo de la interfaz de usuario
Plan de Pruebas	Elaboración del plan de pruebas	QA/Testing	26 de febrero de 2025	28 de febrero de 2025	Plan de pruebas
<b>Fase 2: Desarrollo del MVP (3 de marzo - 30 de mayo 2025)</b>					
APIs del Backend	Desarrollo del backend: APIs	Desarrollador Full-Stack	3 de marzo de 2025	20 de marzo de 2025	APIs documentadas
Modelos de IA	Desarrollo del backend: Modelos de IA	Ingeniero de Machine Learning	3 de marzo de 2025	10 de abril de 2025	Modelos de IA entrenados y validados
Integraciones del Backend	Desarrollo del backend: Integraciones	Desarrollador Full-Stack	11 de abril de 2025	30 de abril de 2025	Integraciones con sistemas externos
Interfaz de Usuario	Desarrollo del frontend: Interfaz de usuario	Desarrollador Full-Stack	1 de mayo de 2025	20 de mayo de 2025	Interfaz de usuario funcional
Componentes Interactivos	Desarrollo del frontend: Componentes interactivos	Desarrollador Full-Stack	21 de mayo de 2025	30 de mayo de 2025	Componentes interactivos implementados
Chatbots	Desarrollo de los agentes virtuales: Chatbots	Ingeniero de Machine Learning	3 de marzo de 2025	20 de abril de 2025	Chatbots funcionales
Personalización de Agentes	Desarrollo de los agentes virtuales:	Ingeniero de Machine	21 de abril de 2025	30 de mayo de	Agentes virtuales personalizados

	Personalización	Learning		2025	
<b>Fase 3: Pruebas Piloto y Ajustes (26 de mayo - 27 de junio 2025)</b>					
Hito	Tareas	Responsable	Fecha Inicio	Fecha Fin	Entregable
Pruebas con Estudiantes	Pruebas del MVP con estudiantes	QA/Testing	26 de mayo de 2025	6 de junio de 2025	Reporte de pruebas de usuario
Diagnóstico Automatizado	Diagnóstico Automatizado	Especialista en Datos	26 de mayo de 2025	6 de junio de 2025	Reporte de diagnóstico
Feedback del MVP	Recopilación de feedback	Expertos en Educación	7 de junio de 2025	13 de junio de 2025	Reporte de feedback
Ajustes de la Interfaz	Ajustes del software: Interfaz	Desarrollador Full-Stack	14 de junio de 2025	20 de junio de 2025	Interfaz de usuario ajustada
Ajustes de Funcionalidades	Ajustes del software: Funcionalidades	Desarrollador Full-Stack	14 de junio de 2025	20 de junio de 2025	Funcionalidades ajustadas
Ajustes del Contenido	Ajustes del software: Contenido	Expertos en Educación	14 de junio de 2025	27 de junio de 2025	Contenido ajustado
<b>Fase 4: Implementación Final (30 de junio - 1 de agosto 2025)</b>					
Hito	Tareas	Responsable	Fecha Inicio	Fecha Fin	Entregable
Implementación en Producción	Implementación en producción	Especialista en Datos	30 de junio de 2025	15 de julio de 2025	Software en producción
Capacitación de Usuarios	Capacitación de usuarios	Expertos en Educación	16 de julio de 2025	25 de julio de 2025	Material de capacitación
Monitoreo del Sistema	Monitoreo del sistema	Especialista en Datos	26 de julio de 2025	1 de agosto de 2025	Reporte de monitoreo

➤ Entregables clave con aprobación del Director de Operaciones



Entregables clave con aprobación del Director de Operaciones		
Entregable	Fecha de Entrega	Rol del Director de Operaciones
Documento de diseño del software	10-Feb-2025	Aprobación formal de objetivos y alcance.
Plan de pruebas	14-Feb-2025	Validación de escalabilidad y seguridad.
Versión MVP del software	23-May-2025	Revisión estratégica antes de pruebas.
Versión del software con ajustes	6-jun.-2025	Aprobación de ajustes basados en feedback.
Documentación de usuario	11-jul.-2025	Autorización para despliegue final.

➤ Reuniones clave con el Director de Operaciones

Reunión	Fecha	Objetivo
Kickoff Inicial	5-Feb-2025	Alinear expectativas y aprobar cronograma.
Revisión de Arquitectura	14-Feb-2025	Validar tecnologías y costos iniciales.
Checkpoint MVP	23-May-2025	Aprobar avances antes de pruebas piloto.
Revisión de Resultados	27-jun.-2025	Aprobar ajustes para implementación final.
Lanzamiento Oficial	1-ago.-2025	Confirmar éxito del proyecto y próximos pasos.

# Presupuesto

## ROI Esperado

1. Reducción del 40% en tiempo de capacitación de nuevos estudiantes
2. Incremento del 35% en retención de usuarios gracias a la personalización.

## Aprobaciones

Aprobado por área solicitante: Dirección general

Nombre: \_\_\_\_\_ Firma: \_\_\_\_\_

Fecha: [DD/MM/AAAA]

Revisado por Área de Tecnología:

Nombre: \_\_\_\_\_ Firma: \_\_\_\_\_

Fecha: [DD/MM/AAAA]