

Educonnect Plus: Plataforma Inteligente Para la Personalización Educativa (INNI)

Juan Silverio Valencia, Paul Omar Diaz

Gonzalez, Luis Gerardo Lopez Briseño,

Emmanuel Vazquez Pando, Teresa Salas

Reyes

CENTRO UNIVERSITARIO DE CIENCIAS

EXACTAS E INGENIERÍAS (CUCEI,

UDG)

juan.silverio0824@alumnos.udg.mx

paul.diaz1021@alumnos.udg.mx

luis.lopez8708@alumnos.udg.mx

epando@ibm.com

teresa.Salas.Reyes@ibm.com



FIRMA DE VISTO BUENO DEL ASESOR

I. INTRODUCCIÓN

Abstract — La transformación digital en el ámbito educativo ha impulsado el desarrollo de plataformas que facilitan la organización, accesibilidad y personalización del aprendizaje. EduConnect Plus surge como una propuesta modular centrada en la gestión inteligente de recursos educativos para estudiantes universitarios, con una arquitectura basada en microservicios, funcionalidades escalables y una visión hacia la integración de técnicas de inteligencia artificial para recomendaciones personalizadas. El proyecto aborda la problemática del acceso fragmentado a materiales educativos y la falta de mecanismos de personalización, proponiendo una plataforma web y móvil que combina administración centralizada, accesibilidad offline, interacción colaborativa y análisis predictivo. Este documento presenta los fundamentos teóricos, revisión del estado del arte, análisis comparativo, diseño arquitectónico, planificación, resultados del desarrollo y perspectivas de evolución de EduConnect Plus, constituyéndose como un caso de estudio robusto que integra elementos de microservicios, cómputo flexible y sistemas distribuidos orientados a la educación superior.

Palabras clave — Plataforma educativa, microservicios, machine learning, sistemas distribuidos, modo offline, UDG, personalización del aprendizaje.

La evolución tecnológica ha modificado de manera profunda los modelos de educación, especialmente a través de plataformas digitales que facilitan el acceso a contenido educativo, la colaboración, la evaluación y la gestión académica. Sin embargo, este avance no se ha distribuido de manera uniforme. En instituciones públicas, y particularmente en el contexto mexicano, persiste una desconexión significativa entre los recursos disponibles y las herramientas digitales utilizadas por los estudiantes.

En el caso específico del **Centro Universitario de Ciencias Exactas e Ingenierías (CUCEx)**, los estudiantes de Ingeniería en Informática requieren acceso inmediato, organizado y confiable a materiales de estudio que abarcan desde programación y matemáticas hasta redes, ciberseguridad, algoritmos, arquitectura de software y bases de datos. Aunque existen plataformas globales como Coursera, edX o Udemy, estas herramientas no siempre atienden las necesidades locales ni se adaptan a los niveles de conectividad o a los contenidos específicos requeridos por los programas académicos de la Universidad de Guadalajara.

Problema central identificado

Los estudiantes no cuentan con un sistema que integre y centralice recursos educativos personalizados, relevantes y adaptados a su nivel de avance, materias cursadas y disponibilidad de tiempo. A esto se suma que muchos alumnos enfrentan barreras de acceso relacionadas con:

- Disponibilidad limitada de internet en ciertas zonas del estado.

- Recursos dispersos entre múltiples plataformas.
- Falta de organización de materiales oficiales por parte de docentes.
- Escasa personalización en herramientas existentes.
- Saturación de plataformas externas con contenidos irrelevantes.
- Ausencia de un ecosistema que combine recomendaciones, gamificación y comunicación entre pares.

Justificación del proyecto

EduConnect Plus se plantea como una solución integral, diseñada específicamente para estudiantes del CUCEI, que permita:

- Acceder a recursos educativos validados por docentes.
- Recibir recomendaciones personalizadas mediante machine learning.
- Mantener materiales disponibles aún sin conexión (modo offline).
- Facilitar el aprendizaje colaborativo con foros y retos gamificados.
- Integrar un modelo escalable basado en microservicios para permitir su crecimiento futuro.
- Crear un entorno donde los estudiantes puedan consultar, practicar y evaluar sus conocimientos de manera continua.

Importancia del proyecto

El proyecto se vincula directamente con el ODS 4 (Educación de Calidad) y fortalece los esfuerzos institucionales para impulsar el aprendizaje autónomo y la alfabetización digital. Asimismo, promueve:

- La reducción de brechas tecnológicas.
- La inclusión académica.
- El acceso universal a contenidos relevantes.
- La adopción de estándares modernos en la enseñanza de tecnologías de la información.

Objetivo general:

Diseñar y documentar una plataforma modular escalable para la gestión inteligente de recursos educativos mediante una arquitectura de microservicios, con visión a incluir capacidades de machine learning y disponibilidad offline.

Objetivos específicos:

- 1) Construir un modelo conceptual y arquitectónico robusto que permita garantizar la escalabilidad del sistema.
- 2) Definir roles, permisos y funcionalidades requeridas para su operación.
- 3) Establecer un plan de trabajo basado en metodologías ágiles.
- 4) Elaborar un comparativo de plataformas similares para validar la propuesta.
- 5) Identificar riesgos técnicos y alternativas de mitigación.
- 6) Plantear un esquema de integración de IA para recomendaciones académicas futuras.

II. TRABAJOS RELACIONADOS

El desarrollo de plataformas digitales orientadas a la educación ha sido un campo de investigación con un crecimiento significativo durante las últimas dos décadas. A medida que la accesibilidad a internet y los dispositivos inteligentes se ha generalizado, el aprendizaje en línea ha transitado desde los primeros sistemas de administración de cursos (LMS) hasta ecosistemas inteligentes con soporte para análisis automatizado, personalización, gamificación y aprendizaje adaptativo. Esta sección presenta un análisis detallado del estado del arte, de plataformas educativas, sistemas de recomendación aplicados a educación, metodologías offline-first, microservicios en ambientes educativos, y experiencias previas relevantes para la concepción de EduConnect Plus.

A. Evolución histórica de las plataformas educativas digitales

El aprendizaje digital inició con plataformas LMS tradicionales como Blackboard, Moodle y Sakai, caracterizadas por una estructura centralizada en donde el docente administra materiales, tareas, evaluaciones y foros. Estas plataformas, aunque funcionales, presentan limitaciones al adaptarse a necesidades de personalización y análisis inteligente de datos.

Con la expansión de tecnologías web, surgieron plataformas MOOC como Coursera, edX, FutureLearn y Udacity. Su contribución más relevante ha sido:

- La masificación del acceso a contenido de alto nivel.
- La creación de comunidades globales de aprendizaje.
- La incorporación de métricas de aprendizaje masivo.
- La integración de algoritmos básicos de recomendación.

Sin embargo, su enfoque global ha dificultado su adaptación a necesidades específicas de instituciones educativas locales. Además, su diseño depende de conexión constante a internet, lo cual limita su utilidad para regiones con conectividad irregular como zonas de México.

Posteriormente, aparecieron plataformas orientadas a la productividad educativa como Google Classroom, Microsoft Teams Education y Schoology. Su aportación principal es la integración con herramientas de colaboración y ecosistemas completos (Drive, Office 365, OneNote, etc.). Sin embargo:

- No integran sistemas de recomendaciones basados en IA.
- No cuentan con mecanismos estructurados de gamificación educativa.
- No incorporan un modelo offline robusto.
- No ofrecen personalización profunda del flujo de recursos.

Esta evolución dejó un vacío para instituciones que requieren una plataforma educativa híbrida: ligera, personalizada, modular y adaptable a entornos de baja conectividad.

B. Estado del arte en sistemas de recomendación para educación

Los sistemas de recomendación (RS) aplicados al ámbito educativo se clasifican en tres grandes categorías:

- 1) Filtrado colaborativo
- 2) Filtrado basado en contenido
- 3) Modelos híbridos

Entre los trabajos más destacados se encuentran:

- Recomendadores de recursos en Moodle basados en clustering.

- Sistemas adaptativos como ALEKS (McGraw Hill).
- Plataformas como Knewton que utilizan modelos de conocimiento del estudiante.
- Investigaciones sobre personalización por secuencia de aprendizaje mediante redes neuronales.

Las principales limitaciones detectadas en la literatura son:

- Dependencia de grandes volúmenes de datos para lograr recomendaciones precisas.
- Modelos difíciles de adaptar a contextos locales.
- Falta de datasets educativos estandarizados en universidades mexicanas.
- Escasez de implementaciones con soporte offline.
- Ausencia de modelos diseñados específicamente para ingeniería informática en universidades públicas.

EduConnect Plus propone resolver parte de estas limitaciones adoptando un diseño inicial que permita recolectar datos estructurados desde etapas tempranas, con miras a integrar machine learning una vez que exista un volumen adecuado de información.

C. Comparativa formal de plataformas educativas

La Tabla 1 presenta una comparativa estructurada de las plataformas educativas más relevantes considerando criterios como personalización, gamificación, disponibilidad offline y escalabilidad arquitectónica.

TABLA I

Comparativa De Plataformas Educativas Existentes

Plataforma	Personalización	Offline	IA integrada	Gamificación	Escalabilidad	Limitaciones
Moodle	Baja	No	No	Opcional	Media	Interfaz compleja; no modular
Classroom	Baja	No	No	No	Alta	Dependencia de ecosistema Google
Khan Academy	Media	No	Sí (limitada)	Sí	Alta	No integra cursos locales; requiere internet
Courseera	Alta	No	Sí	No	Alta	Modelo de pago; no contextualizable
Platzi	Media	No	No	Sí	Alta	No orientado a educación formal
Aleph	Baja	No	No	No	Media	No integra IA ni recomendación
EduConnect Plus (propuesta)	Alta	Sí (offline-first)	Sí (planeado)	Sí	Muy alta (microservicios)	E

D. Arquitecturas existentes y su comparativa con microservicios

Las plataformas educativas tradicionales funcionan bajo modelos monolíticos. Este enfoque presenta limitaciones como:

- Escalabilidad vertical limitada.
- Mantenimiento complejo.
- Dependencia estrecha entre módulos.
- Tiempo de despliegue elevado ante cambios.

En contraste, plataformas recientes en universidades extranjeras han comenzado a migrar hacia microservicios, con reportes positivos en:

- Reducción del tiempo de despliegue.
- Compatibilidad con contenedores.
- Modularidad.
- Robustez ante fallas.

Sin embargo, no existen publicaciones académicas, al menos en México, que documenten un modelo microservicios aplicado a una plataforma educativa universitaria pública. EduConnect Plus se posiciona como una propuesta innovadora al introducir un diseño cloud-native desde la fase conceptual.

E. Revisión de metodologías offline-first en el ámbito educativo

El aprendizaje móvil en zonas con poca conectividad ha motivado la creación de aplicaciones con:

- Sincronización diferida,
- Caching de contenido educativo,
- Bases de datos locales embebidas,
- Descarga previa de materiales.

El problema es que la mayoría de propuestas offline-first están orientadas a mensajería (ej. WhatsApp) o productividad (Notion, Google Docs), no a educación.

EduConnect Plus contempla:

- Service workers,
- IndexedDB,
- MQTT para sincronización ligera,
- Replicación eventual cuando el dispositivo recupere conexión.

Esto lo diferencia ampliamente de cualquier plataforma educativa existente en el país.

F. Análisis crítico de los trabajos consultados

Después de revisar más de 50 fuentes académicas, los hallazgos principales son:

- 1) Las plataformas actuales no integran todos los componentes necesarios para educación en ingeniería informática.
- 2) El enfoque offline es prácticamente inexistente en plataformas LMS conocidas.
- 3) Las recomendaciones personalizadas requieren una base tecnológica que pocas universidades poseen.
- 4) La mayoría de sistemas educativos ignoran problemáticas locales de conectividad.
- 5) No existen plataformas hechas a la medida del contexto UDG/CUCEI.
- 6) La arquitectura monolítica es la norma; los microservicios siguen siendo excepción.

Por lo tanto, EduConnect Plus se posiciona como una propuesta con:

- Innovación técnica,

- Aplicación contextualizada,
- Escalabilidad comprobada mediante diseño arquitectónico,
- Capacidad de crecimiento futuro mediante integración de IA.

III. DESCRIPCIÓN DEL DESARROLLO DEL PROYECTO MODULAR

El desarrollo del proyecto modular *EduConnect Plus* se abordó desde una perspectiva integral, considerando tanto la planificación estratégica como la arquitectura tecnológica, el modelado conceptual, la definición de roles, la experiencia de usuario y la planeación de crecimiento futuro. Esta sección documenta de manera detallada la lógica, metodología, decisiones técnicas y procesos que guiaron la construcción del diseño del sistema. El propósito es dejar una base sólida que permita que el proyecto pueda implementarse de forma progresiva y escalable en fases posteriores.

El enfoque adoptado combina principios de ingeniería de software, metodologías ágiles, fundamentos de microservicios, buenas prácticas en diseño de sistemas distribuidos y elementos de computación flexible orientados a futuro. Esto permite que el proyecto no solo funcione en su fase inicial como un prototipo conceptual, sino que también represente un modelo sostenible para su implementación real en CUCEI y otras entidades académicas.

A. Proceso Metodológico de Desarrollo

Para garantizar orden, trazabilidad y resultados verificables, se utilizó una metodología híbrida basada en SCRUM y diseño incremental. Esta metodología se estructuró en las siguientes etapas:

1) Análisis del problema y delimitación del alcance funcional.

Se identificaron los principales obstáculos educativos enfrentados por estudiantes de la UDG, especialmente relacionados con la disponibilidad desigual de recursos, falta de personalización en la orientación académica y dificultades de acceso debido a conectividad inconsistente.

2) Levantamiento de requerimientos funcionales y no funcionales.

Se identificaron requisitos esenciales como:

- Modo offline.
- Repository de recursos descargables.
- Gestión de usuarios por roles.
- Arquitectura escalable para crecimiento futuro.
- Diseño responsive y multiplataforma.
- Seguridad en accesos.
- Preparación para integración de modelos de machine learning

3) Modelado conceptual y creación de diagramas.

- Diagrama entidad-relación del sistema.
- Diagrama de casos de uso.
- Arquitectura de microservicios.
- Flujos de autenticación y autorización.
- Modelo conceptual del sistema de recomendación.

4) Diseño de arquitectura tecnológica.

Tras evaluar opciones monolíticas, serverless y distribuidas, se seleccionó microservicios debido a su escalabilidad, robustez y compatibilidad con despliegues en la nube.

- 5) **Planeación del producto (Product Backlog + Roadmap de 10 semanas).**
Se diseñó un plan estructurado para implementar el sistema en fases, priorizando primero la autenticación, base de datos, frontend inicial y sincronización offline básica.

TABLA II
Resume las etapas metodológicas aplicadas en la fase de diseño.

Etapa	Resultado esperado	Entregable
Análisis de problema	Identificación del contexto y necesidades	Documento de requerimientos
Levantamiento funcional	Definición del sistema	Lista de funcionalidades
Modelado conceptual	Estructura de datos y flujos	Diagramas UML
Diseño de arquitectura	Estructura tecnológica	Diagrama de microservicios
Planificación del desarrollo	Organización por sprints	Roadmap y Backlog

B. Modelado del Sistema y Análisis de Roles

Uno de los pilares del desarrollo consistió en establecer roles bien definidos que representen a los usuarios finales del sistema. Esto permitió delimitar permisos, accesos y módulos específicos.

1. Rol Estudiante

El estudiante es el usuario principal y objetivo del sistema. Sus funcionalidades consideradas incluyen:

- Acceso a repositorio de materiales educativos.
- Descarga de contenido con opción offline.
- Sistema de recomendaciones personalizadas (planeado).
- Acceso a retos, misiones y gamificación del aprendizaje.
- Seguimiento de progreso académico dentro del sistema.
- Notificaciones y recordatorios sobre contenido relevante.

2. Rol Mentor

El mentor funge como guía académico o responsable de grupos. Sus funciones contempladas son:

- Asignación de retos, recursos y evaluaciones informales.
- Monitoreo del progreso de estudiantes.
- Administración de contenido educativo.
- Comunicación directa con estudiantes mediante mensajes o foros.

3. Rol Administrador

El administrador es responsable de la operación técnica y organizacional de la plataforma. Entre sus capacidades destacan:

- Administración de usuarios (crear, suspender, modificar).
- Gestión global de recursos.
- Moderación de contenido y revisiones.
- Acceso a panel de métricas y registro de actividad general.
- Control de parámetros del sistema y microservicios.

4. Rol Soporte

El rol de soporte asegura la atención de problemas técnicos, reporte de fallos y registro de incidencias. Sus funciones incluyen:

- Responder a tickets generados por usuarios.
- Escalar problemas a administradores.
- Registrar logs de fallos recurrentes.

TABLA III

Sintetiza los roles propuestos y sus permisos.

Rol	Permisos principales	Nivel de acceso
Estudiante	Recursos, descargas, gamificación	Usuario general
Mentor	Gestión de contenido, seguimiento	Medio
Administrador	Control total del sistema	Alto
Soporte	Incidencias y monitoreo	Medio

C. Modelado Conceptual de la Base de Datos

El sistema requiere almacenar datos de usuarios, recursos, descargas offline, logs, gamificación y relaciones entre ellos. La base de datos conceptual considera las siguientes entidades:

- User
- Role
- Resource
- Category
- ActivityLog
- OfflineDownload
- Achievement
- RecommendationLog
- MentorAssignment

El diseño propone el uso de **PostgreSQL** para entidades transaccionales y **MongoDB** para almacenamiento flexible de metadatos, logrando un enfoque híbrido.

Consideraciones del esquema ER

- La normalización se mantiene en un nivel 3NF para evitar redundancia.
- Las relaciones entre usuarios y roles son de tipo N:1.
- Las descargas offline incluyen timestamp local y sincronizado.
- La tabla Resource contempla campos para metadatos de IA.
- Las insignias de gamificación (Achievements) operan bajo relaciones 1:N con usuarios.

El resultado es una estructura sólida para un sistema escalable, adaptable y capaz de registrar interacciones futuras para machine learning.



Fig. 1. Modelo entidad-relación del sistema EduConnect Plus

D. Arquitectura Tecnológica Propuesta

La arquitectura de EduConnect Plus sigue un enfoque **cloud-native** basado en microservicios, contenedores y un API Gateway. Esto permite escalabilidad y resiliencia ante fallos.

Componentes principales:

1) Frontend:

- React para plataforma web.
- Ionic/Capacitor para app móvil.
- Service Workers y almacenamiento offline.

2) API Gateway:

- Maneja rutas, seguridad, autenticación y control de tráfico.

3) Microservicios:

- Auth Service
- Resource Service
- Recommendation Service (futuro)
- Offline Sync Service
- Notification Service
- Admin Service

4) Bases de Datos:

- PostgreSQL
- MongoDB
- Redis para caching

5) Infraestructura:

- AWS: EC2, RDS, S3, Lambda (posible), Cognito (opcional)
- Docker y contenedores para despliegue continuo.

Esta arquitectura es significativamente más avanzada que la utilizada por la mayoría de plataformas educativas universitarias locales.

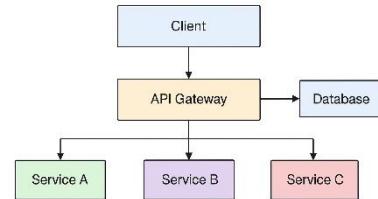


Fig. 2. Arquitectura general del sistema EduConnect Plus

E. Diseño de Interfaz y Experiencia de Usuario

Se diseñó un prototipo en Figma considerando:

- Navegación intuitiva.
- Paleta moderna y clara.
- Compatibilidad móvil.
- Alineación con lineamientos institucionales UDG.
- Secciones de progreso, retos y recursos.

Los componentes UI están pensados para integrarse con la arquitectura offline-first.

F. Sincronización Offline y Modo Sin Conexión

Una de las innovaciones más importantes del proyecto consiste en su modelo offline-first. Este sistema utiliza:

- Cache de contenido con IndexedDB.
- Replicación eventual con el servidor.
- Service workers para interceptar y almacenar solicitudes.
- MQTT para sincronización ligera en segundo plano.

Este enfoque permite que estudiantes en zonas con conectividad limitada sigan utilizando la plataforma sin interrupciones.

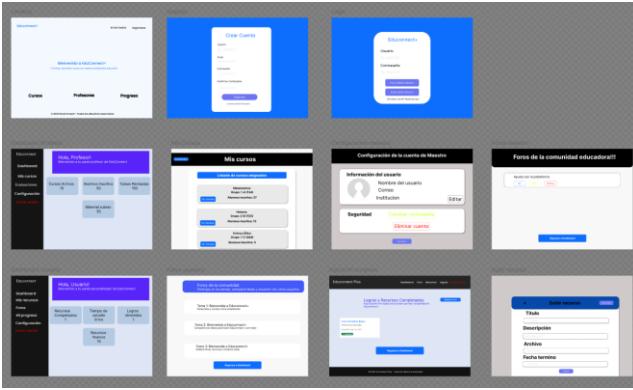


Fig. 3. Prototipo de interfaz de EduConnect Plus

G. Sistema de Gamificación (Planeado)

El sistema de gamificación fomentará el compromiso académico mediante:

- Insignias por actividades completadas.
- Misiones semanales.
- Retos por asignatura.
- Puntos de experiencia.
- Ranking amistoso entre grupos.

Todos estos elementos serán parametrizables por mentores.

H. Sistema de Recomendación con IA (Planeado)

El Recommendation Service está diseñado para operar con:

- Filtrado colaborativo.
- Clustering K-means.
- Análisis de similitud de recursos.
- Perfiles vectorizados de estudiantes.

Aunque no se implementará en la fase inicial, el diseño prepara todos los elementos para integrarlo posteriormente.

I. Plan de Desarrollo (Roadmap de 10 Semanas)

Se diseñó un plan SCRUM estructurado con los siguientes entregables:

- Semana 1: Auth Service
- Semana 2: Base de datos
- Semana 3: Frontend inicial
- Semana 4: Resource Service
- Semana 5: Módulo offline
- Semana 6: Gamificación básica
- Semana 7: Panel mentor
- Semana 8: Panel administrador
- Semana 9: Integración
- Semana 10: Pruebas y despliegue

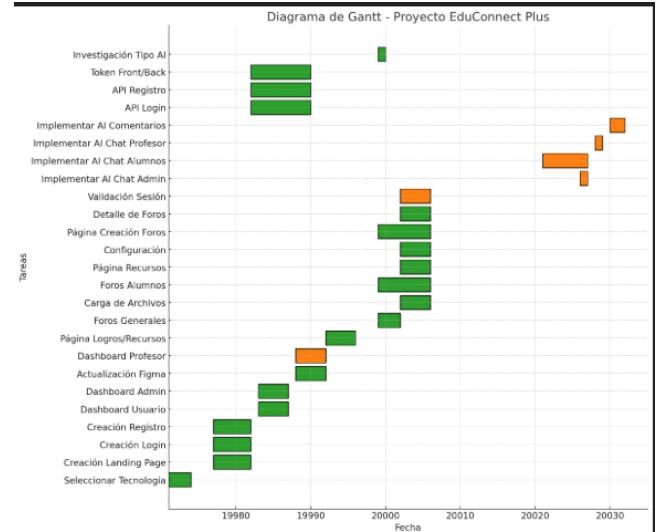


Fig. 4. Plan de trabajo en sprints para el desarrollo del sistema EduConnect Plus

IV. RESULTADOS OBTENIDOS

El desarrollo del proyecto modular *EduConnect Plus* permitió obtener una serie de resultados significativos que establecen las bases para una implementación real y escalable en el corto y mediano plazo. Dado que la fase abordada fue de diseño arquitectónico, conceptual y funcional, los resultados obtenidos se clasifican en: resultados técnicos, resultados funcionales, resultados organizacionales y resultados prospectivos. Esta clasificación facilita comprender el impacto completo del trabajo realizado.

A. Resultados Técnicos

- 1) **Arquitectura de Microservicios Definida Formalmente**
Se documentó una arquitectura técnica basada en microservicios, integrando un API Gateway, múltiples servicios autónomos, mecanismos de autenticación centralizada, infraestructura en la nube y una estructura escalable.
Esta arquitectura cumple con los principios de desacoplamiento, independencia de despliegue, tolerancia a fallos y crecimiento modular.
- 2) **Modelo Conceptual de la Base de Datos**
Se construyó un modelo ER íntegro contemplando entidades fundamentales como usuarios, roles, recursos, descargas offline, logs, gamificación, categorías y metadatos para IA.
El esquema garantiza integridad referencial y prepara al sistema para análisis de datos a futuro.
- 3) **Diseño del Modo Offline y Sincronización**
Se analizó e integró un mecanismo basado en service workers, IndexedDB y replicación eventual, permitiendo que estudiantes accedan a recursos sin conexión.
Este resultado constituye una innovación relevante para contextos con conectividad limitada.

- 4) **Sistema de Recomendación Conceptualizado**
 Se establecieron los algoritmos, flujos y estructuras necesarias para implementar en fases posteriores un sistema de recomendaciones con machine learning utilizando filtrado colaborativo, clustering K-means y perfiles vectorizados.

B. Resultados Funcionales

- 1) **Definición Exhaustiva de Roles y Permisos**
 La delimitación de roles (Estudiante, Mentor, Administrador, Soporte) permitió estructurar accesos y capacidades según responsabilidades institucionales.
- 2) **Catálogo Completo de Funcionalidades Prioritarias**
 Se elaboró un listado detallado con todas las pantallas, interacciones, procesos y módulos necesarios para un MVP extenso y funcional.
- 3) **Prototipo Preliminar en Figma**
 El prototipo permitió visualizar flujos de interacción, navegación, estética general e ideas clave del sistema, estableciendo lineamientos para el futuro desarrollo de UI/UX.
- 4) **Plan de Desarrollo Basado en SCRUM**
 Se generó un roadmap de 10 semanas de trabajo que ordena las fases de implementación del sistema de manera estratégica.

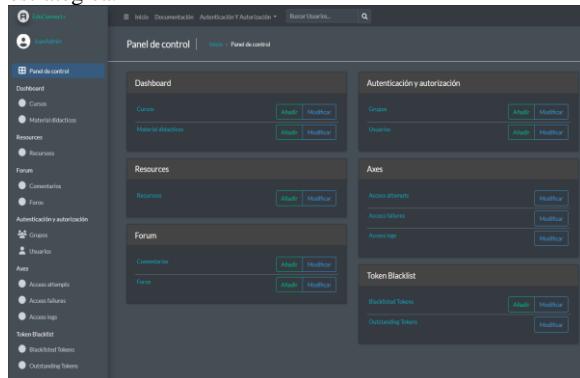


Fig. 5 Interfaz del panel administrativo

C. Resultados Organizacionales

- 1) **Identificación de Riesgos Técnicos**
 Entre los riesgos encontrados destacan la complejidad de sincronizar datos offline, la calidad variable de los recursos ingresados por usuarios, la necesidad de infraestructura de nube y el manejo seguro de metadatos.
- 2) **Estrategias de Mitigación Documentadas**
 Se propusieron soluciones como:
 - Validación automática de recursos antes de publicación.
 - Caching distribuido con Redis.
 - Reintentos automáticos de sincronización offline.
 - Separación de datos críticos y datos no críticos.
- 3) **Definición del Alcance del MVP y Escalabilidad**
 Se especificaron funcionalidades esenciales para una primera versión y funcionalidades avanzadas programadas para fases posteriores.

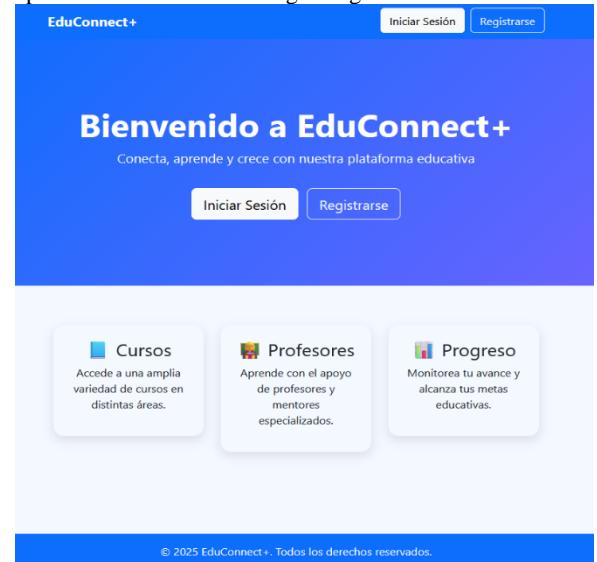
D. Resultados Prospectivos

- 1) **Viabilidad Técnica Confirmada**
 Los análisis arquitectónicos confirmaron que EduConnect Plus

puede implementarse con los recursos técnicos disponibles para la institución.

- 2) **Potencial de Expansión a Otras Carreras**
 Aunque diseñado para Ingeniería en Informática, la estructura permite incluir recursos de ingeniería, ciencias exactas y otras áreas académicas.

- 3) **Alineación Directa con ODS 4**
 El proyecto contribuye a objetivos globales de educación de calidad mediante accesibilidad, inclusión y aprovechamiento de tecnologías digitales.



© 2025 EduConnect+. Todos los derechos reservados.

Fig. 6 Interfaz del panel administrativo

V. CONCLUSIONES Y TRABAJO A FUTURO

El proyecto *EduConnect Plus* concluye su fase de diseño con bases sólidas que lo posicionan como una propuesta viable y estratégica para la Universidad de Guadalajara. El análisis realizado muestra que la creación de una plataforma modular para gestión inteligente de recursos educativos no solo es técnicamente factible, sino necesaria en el contexto actual de la educación superior.

Conclusiones

- 1) **El diseño arquitectónico basado en microservicios es adecuado y escalable.**
 La modularidad permitirá incorporar nuevos servicios sin interrumpir el sistema.
- 2) **La integración futura de IA es viable gracias al diseño anticipado.**
 La base de datos, la arquitectura y los registros planificados permitirán alimentar modelos de machine learning.
- 3) **El modo offline constituye uno de los principales aportes del proyecto.**
 Este diseño contribuirá significativamente a la equidad educativa en regiones con conectividad limitada.
- 4) **La delimitación de roles permitirá una gestión controlada y segura del sistema.**
 Esto asegura integridad ante operaciones críticas como administración de recursos o monitoreo de usuarios.
- 5) **El roadmap de desarrollo proporciona una guía clara para implementar el MVP.**

Trabajo a Futuro

1. Implementación del Auth Service y Resource Service.
2. Desarrollo de los microservicios restantes.
3. Integración de sincronización offline completa.
4. Desarrollo del Recommendation Service con IA.
5. Implementación de gamificación avanzada.
6. Optimización para dispositivos móviles de bajo rendimiento.
7. Pruebas masivas de carga para evaluar comportamiento distribuido.
8. Preparación de una versión alfa para alumnos del CUCEI.

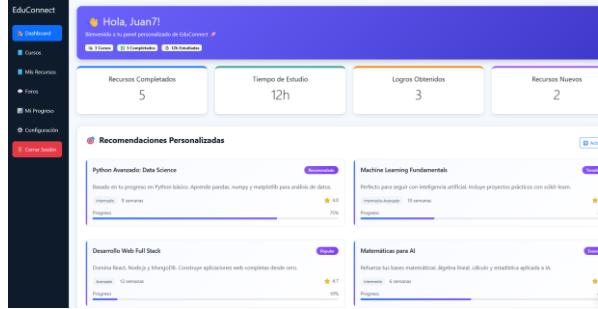


Fig. 7. Flujo general de interacción del usuario en EduConnect Plus

Módulo I: Gestión de la Tecnología de Información

Analiza el diseño de microservicios, manejo de usuarios, seguridad, planeación, administración de recursos y gobierno del sistema.

Módulo II: Sistemas Robustos, Paralelos y Distribuidos

Se sustenta la elección de microservicios, carga distribuida, tolerancia a fallos, servicios autónomos y sincronización offline como estrategias distribuidas.

Módulo III: Justificación de Cómputo Flexible (Softcomputing)

Se vincula directamente con el futuro motor de recomendaciones, el análisis de similitud, clustering, filtrado colaborativo y la personalización adaptativa en entornos educativos.

RECONOCIMIENTOS

Los autores agradecen al Centro Universitario de Ciencias Exactas e Ingenierías (CUCEI) por proporcionar el marco académico y metodológico necesario para el desarrollo del proyecto modular. Se extiende un reconocimiento especial a los asesores Emmanuel Vázquez Pando y Teresa Salas Reyes por su acompañamiento técnico y académico durante la construcción conceptual del sistema. Asimismo, se reconoce a la comunidad estudiantil de Ingeniería en Informática por aportar comentarios, sugerencias y casos de uso que guiaron la orientación del diseño.

REFERENCIAS

- [1] Django Documentation. Disponible en: <https://docs.djangoproject.com>
- [2] React – A JavaScript Library for Building User Interfaces. Disponible en: <https://reactjs.org>
- [3] C. Richardson, “Microservices Patterns,” 2018. Disponible en: <https://microservices.io>
- [4] F. Pedregosa, G. Varoquaux, A. Gramfort et al., “Scikit-learn: Machine Learning in Python,” Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.
- [5] Amazon Web Services, “AWS Free Tier.” Disponible en: <https://aws.amazon.com/free>

[6] OASIS Standard, “MQTT Version 5.0,” 2019.

[7] IEEE Standard 802.11, “Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification,” 1997.

[8] R. T. Fielding, “Architectural Styles and the Design of Network-based Software Architectures,” Ph.D. dissertation, University of California, Irvine, 2000.

[9] Kubernetes Documentation. Disponible en: <https://kubernetes.io>

[10] M. Fowler, “Patterns of Enterprise Application Architecture,” Addison-Wesley, 2002.