**Transformación Digital de EduTech Innovators SPA**

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Experiencia 1 – Arquitectura de Microservicios

Benjamín Torrejón Soto – Alejandra Reyes Duque

Desarrollo Fullstack I 002D

Duoc UC – Ingeniería Informática

Abril 2025

**Índice**

**1. Requisitos del Sistema**

1.1 Requisitos Funcionales

1.1.1 Administrador del Sistema

1.1.2 Gerente de Cursos I

1.1.3 Gerente de Cursos II

1.1.4 Logística de Soporte

1.1.5 Clientes vía Web

1.2 Requisitos No Funcionales

1.2.1 Escalabilidad

1.2.2 Disponibilidad

1.2.3 Rendimiento

1.2.4 Seguridad

1.2.5 Sostenibilidad

1.2.6 Ética y Privacidad

**2. Análisis del Sistema Actual**

**3. Diseño de la Nueva Arquitectura**

3.1 Microservicios Propuestos

3.2 Infraestructura y Base de Datos

3.3 Diagrama de Casos de Uso

3.4 Diagrama de Clases

3.5 Diagrama de Despliegue

**4. Plan Detallado de Migración**

4.1 Preparación

4.2 Migración Progresiva

4.3 Validación y Optimización

4.4 Transición Final

**5. Identificación de Riesgos y Plan de Mitigación**

**6. Evaluación Ética en la Implementación de Microservicios**

**7. Uso de Herramientas Colaborativas (Miro y Trello)**

**1. Requisitos del Sistema**

Requisitos Funcionales

Administrador del Sistema:

* El sistema debe permitir gestionar usuarios (crear, actualizar, desactivar y eliminar cuentas).
* Debe permitir configurar y modificar permisos de acceso para diferentes módulos y funciones.
* Debe mostrar información sobre el estado actual del sistema, alertas sobre fallos y rendimiento.
* Debe permitir respaldar y restaurar datos periódicamente.

Gerente de Cursos I:

* El sistema debe permitir crear, actualizar y eliminar cursos del catálogo.
* Debe generar reportes sobre inscripción y rendimiento de cursos.
* Debe gestionar instructores, asignándolos a cursos específicos.
* Debe permitir evaluar y aprobar contenidos antes de su publicación.
* Debe realizar feedback al cliente vía web.

Gerente de Cursos II:

* El sistema debe permitir desarrollar y subir material didáctico actualizado.
* Debe permitir crear, actualizar y corregir evaluaciones.
* Debe proporcionar herramientas para responder preguntas e interactuar con estudiantes mediante foros y chats.
* Debe mostrar información sobre el progreso de los estudiantes inscritos.

Logística de Soporte:

* El sistema debe recibir, asignar y permitir la resolución de incidencias técnicas reportadas por usuarios.
* Debe optimizar y asegurar la disponibilidad de recursos del sistema.
* Debe permitir actualizar el estado de las incidencias reportadas.
* Debe mantener actualizada la información sobre proveedores tecnológicos y de servicios.

Clientes vía Web:

* El sistema debe permitir la creación de cuentas por parte de los clientes.
* Debe facilitar el inicio de sesión con credenciales.
* Debe proporcionar opciones para navegar, buscar e inscribirse en cursos.
* Debe mostrar claramente el progreso en los cursos inscritos.
* Debe permitir gestionar perfiles personales, detalles de pago y preferencias.
* Debe proveer medios para solicitar soporte mediante formularios o chat en línea.
* Debe permitir a los clientes dejar reseñas y calificaciones.
* Debe ofrecer la opción de aplicar cupones y descuentos.

Requisitos No Funcionales

Escalabilidad:

* El sistema debe soportar incrementos considerables en la cantidad de usuarios simultáneos sin degradación del rendimiento.

Disponibilidad:

* Debe garantizar una disponibilidad del servicio superior al 99.9%.

Rendimiento:

* Las respuestas del sistema no deben exceder los 3 segundos ante solicitudes comunes.

Seguridad:

* Debe contar con mecanismos robustos para proteger datos personales y transacciones financieras, utilizando cifrado de datos y acceso controlado por roles.

Sostenibilidad:

* El diseño e implementación deben considerar el impacto ambiental mediante el uso eficiente de recursos tecnológicos.

Ética y Privacidad:

* El manejo de datos personales deberá cumplir estrictamente con normativas vigentes sobre privacidad de datos personales y ética digital, garantizando transparencia en la gestión y uso responsable de información sensible.

**2. Análisis del Sistema Actual**

Actualmente, EduTech Innovators SPA utiliza un sistema de software monolítico que inicialmente permitió gestionar de forma adecuada sus operaciones. Sin embargo, debido al rápido crecimiento y expansión de la empresa, el sistema monolítico ha presentado problemas significativos que afectan negativamente el rendimiento y disponibilidad de los servicios. Entre los principales problemas detectados se encuentran:

* Rendimiento deficiente ante altos volúmenes de usuarios simultáneos, causando lentitud en la respuesta y disminuyendo la satisfacción del usuario.
* Caídas frecuentes del sistema que interrumpen las operaciones diarias y generan una percepción negativa por parte de los usuarios finales.
* Escasa flexibilidad y dificultad para escalar eficientemente el sistema, limitando la capacidad de expansión y adaptación a nuevas necesidades del mercado.
* Problemas en la gestión efectiva de incidencias técnicas, que derivan en demoras en la resolución y una insatisfacción creciente entre los clientes y personal interno.

Estos desafíos reflejan la necesidad urgente de adoptar una arquitectura más robusta y modular, específicamente orientada a microservicios, que permita mejorar significativamente la capacidad de gestión, rendimiento y escalabilidad del sistema.

**3. Diseño de la Nueva Arquitectura**

La solución propuesta se basa en una arquitectura de microservicios distribuida, diseñada para responder a las limitaciones actuales, asegurando escalabilidad, rendimiento óptimo y alta disponibilidad. Los microservicios específicos propuestos incluyen:

* Microservicio de Usuarios y Autenticación: Gestiona la creación y administración de cuentas, permisos y seguridad de acceso.
* Microservicio de Gestión de Cursos: Maneja la creación, actualización, eliminación y asignación de instructores para cursos.
* Microservicio de Gestión de Contenido Educativo: Permite la carga, actualización y gestión del material didáctico disponible.
* Microservicio de Evaluaciones y Reportes: Administra evaluaciones, calificaciones y genera reportes relacionados con el progreso académico.
* Microservicio de Gestión de Soporte Técnico: Facilita la gestión rápida y eficiente de incidencias técnicas reportadas.
* Microservicio de Pagos y Promociones: Gestiona pagos, transacciones financieras y aplicación de descuentos y cupones promocionales.

Cada microservicio contará con su propia base de datos MySQL independiente, asegurando desacoplamiento, flexibilidad y simplicidad en su gestión y mantenimiento. La infraestructura tecnológica propuesta utilizará servicios de la nube AWS, aprovechando la confiabilidad, escalabilidad y amplias capacidades técnicas disponibles en este proveedor.

Diagrama de Casos de Uso

Diagrama

El contenido generado por IA puede ser incorrecto.

Diagrama de Clases

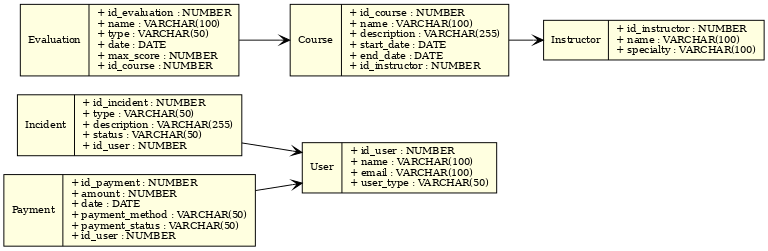
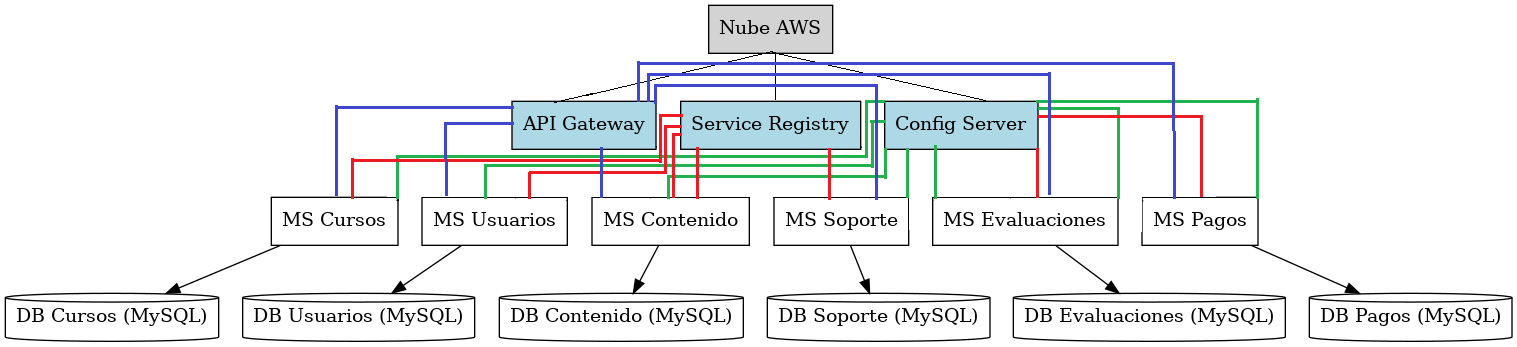


Diagrama de Despliegue (Microservicios con MySQL en AWS)



**4. Plan detallado de Migración**

La migración del sistema monolítico actual a la arquitectura de microservicios propuesta se realizará siguiendo una estrategia progresiva por módulos, con etapas claramente definidas para minimizar riesgos y asegurar continuidad operativa:

1. Preparación:

* Inventario detallado de datos y funcionalidades existentes.
* Configuración de infraestructura inicial en AWS.

1. Migración Progresiva:

* Usuarios y Autenticación.
* Gestión de Cursos y Contenido Educativo.
* Evaluaciones y Reportes.
* Gestión de Soporte Técnico.
* Pagos y Promociones.

1. Validación y Optimización:

* Pruebas de rendimiento y validación.
* Ajustes según resultados.

1. Transición Final:

* Monitoreo constante.
* Apagado progresivo del sistema monolítico.

**5. Identificación de Riesgos y Plan de Mitigación**

| **Riesgo identificado** | **Plan de mitigación** |
| --- | --- |
| Pérdida de datos durante migración | Realizar respaldos frecuentes y comprobación periódica. |
| Retrasos en la entrega del proyecto | Definir hitos y revisiones frecuentes para mantener el cronograma. |
| Falta de conocimiento técnico del equipo | Capacitación rápida y específica en tecnologías seleccionadas. |
| Problemas de integración entre microservicios | Pruebas tempranas de integración continua durante el desarrollo. |
| Sobrecarga del sistema durante la transición | Realizar migración por fases y monitoreo continuo para controlar cargas. |

**6. Evaluación Ética en la Implementación de Microservicios**

Durante el desarrollo de la nueva arquitectura basada en microservicios, se ha considerado un enfoque ético que aborde aspectos clave de responsabilidad digital, privacidad de los usuarios y sostenibilidad tecnológica. Las acciones específicas tomadas incluyen:

* **Privacidad de Datos:** Se garantizará el cumplimiento estricto de la Ley de Protección de Datos Personales mediante el cifrado de datos sensibles, autenticación segura y control de accesos segmentado por roles.
* **Responsabilidad en el Despliegue:** Se aplicarán mecanismos de auditoría para el monitoreo del sistema y se asegurarán prácticas responsables en la gestión de datos e información sensible, con trazabilidad clara en todos los procesos críticos.
* **Transparencia con los Usuarios:** Los usuarios serán informados sobre el uso de sus datos, condiciones del servicio y políticas de privacidad en formatos claros y accesibles desde la plataforma.
* **Impacto en el Empleo:** Se priorizará la capacitación de los empleados actuales sobre las nuevas tecnologías utilizadas, promoviendo la inclusión del equipo en la transición hacia una cultura digital colaborativa.
* **Sostenibilidad Digital:** Se buscará minimizar el impacto ambiental mediante el uso eficiente de recursos en la nube (AWS) y la implementación de prácticas de desarrollo que reduzcan el consumo de energía y recursos computacionales innecesarios.

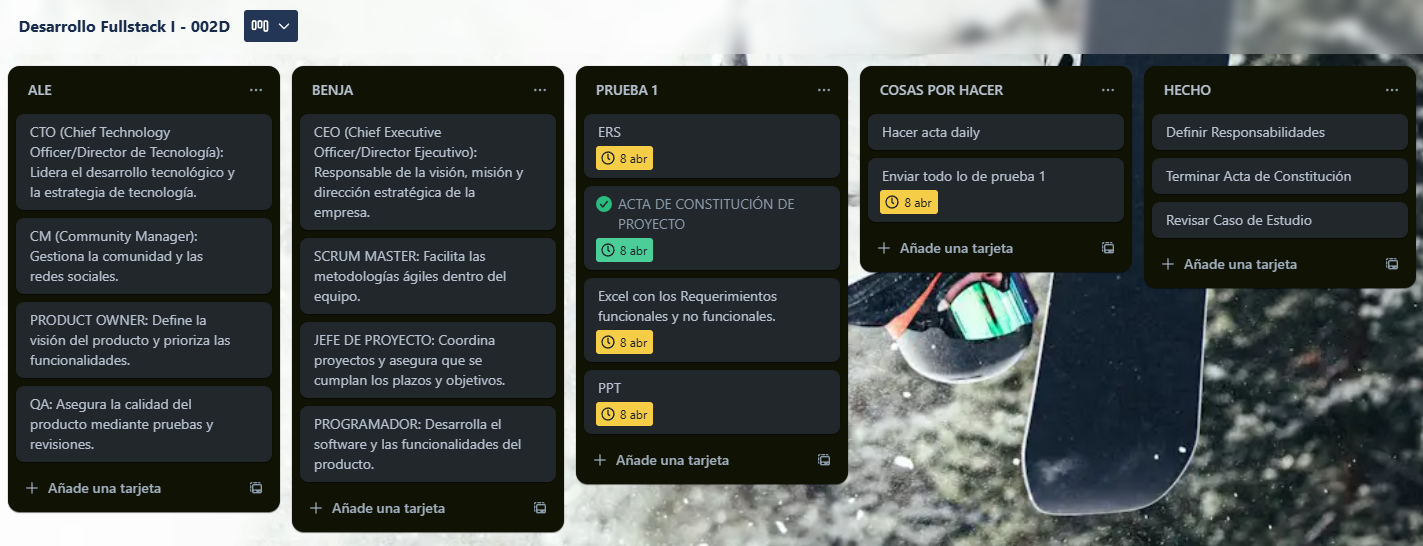
**7. Uso de Herramientas Colaborativas (Miro y Trello)**

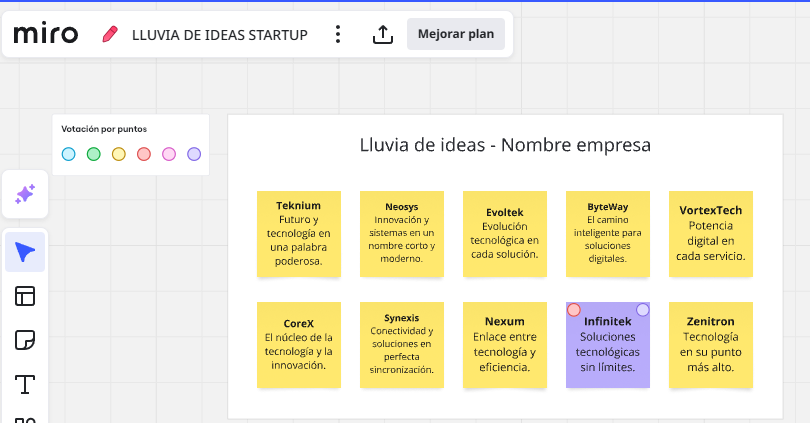
Durante el desarrollo del proyecto, se utilizaron herramientas colaborativas para organizar las actividades del equipo, asignar responsabilidades y fomentar la participación activa de sus miembros.

**Trello** fue implementado como tablero de planificación ágil, permitiendo visualizar el estado de las tareas mediante listas y tarjetas. En el tablero se definieron claramente los roles del equipo, como CEO, CTO, Scrum Master, Product Owner, entre otros. También se organizaron actividades en listas de responsabilidades y tareas por hacer, como la revisión del caso de estudio, la redacción del acta y reuniones de seguimiento (daily).

**Miro** se utilizó para sesiones de lluvia de ideas visuales, como la definición del nombre de la empresa y posibles identidades. Esto fomentó la creatividad y permitió una participación rápida e igualitaria entre los integrantes. Las notas adhesivas y paneles temáticos ayudaron a estructurar propuestas de forma clara.

Ambas plataformas permitieron que el equipo lograra mantener una comunicación efectiva, aún sin una alta carga de reuniones. La combinación de **Miro para ideación** y **Trello para ejecución** facilitó el flujo de trabajo y aportó estructura al proyecto desde sus primeras etapas.





**8. Bibliografía**

- AWS | Cloud Computing - Servicios de informática en la nube -

Documentación del sistema monolítico actual de EduTech Innovators SPA.

[▷ Diagrama de clases. Teoria y ejemplos.](https://diagramasuml.com/diagrama-de-clases/)

[▷ Diagrama de casos de uso. Teoría y ejemplos](https://diagramasuml.com/casos-de-uso/)

[Arquitectura de microservicios | Atlassian](https://www.atlassian.com/es/microservices/microservices-architecture)