

Actividad 1: Selección del Lenguaje y Paradigma de Programación

Este documento tiene como objetivo justificar la elección del lenguaje de programación y el paradigma de desarrollo para el MVP de EduTech IA, una plataforma digital que adapta cursos y contenidos al perfil y necesidades de cada usuario. Se ha seleccionado Java como lenguaje de programación y la Programación Orientada a Objetos (POO) como paradigma, decisiones fundamentadas en criterios de solidez, escalabilidad y seguridad, todos ellos esenciales para cumplir los requerimientos del proyecto.

Justificación de la Elección de Java

Madurez y Fiabilidad:

Java se ha consolidado en el ámbito empresarial gracias a su estabilidad y robustez. Esta característica lo convierte en una opción segura para aplicaciones de misión crítica, lo que es fundamental para garantizar la viabilidad operativa del MVP.

Amplio Ecosistema y Comunidad:

La diversidad de frameworks y herramientas disponibles (como Spring y Hibernate) permite un desarrollo rápido y de alta calidad. El soporte comunitario y la extensa documentación facilitan la resolución de problemas y optimizan el proceso de desarrollo.

Portabilidad y Flexibilidad:

La Máquina Virtual de Java (JVM) asegura que las aplicaciones desarrolladas en este lenguaje puedan ejecutarse en múltiples plataformas sin necesidad de modificaciones significativas. Esta portabilidad es crucial para que el sistema se adapte a futuros requerimientos de escalabilidad.

Integración en Flujos de CI/CD:

La compatibilidad de Java con herramientas de integración y entrega continua permite automatizar pruebas y despliegues, lo cual es esencial para ciclos ágiles y la detección temprana de errores. Esto se alinea con la necesidad del proyecto de contar con un sistema que evolucione de manera continua y segura.

Selección del Paradigma: Programación Orientada a Objetos (POO)

Modularidad y Mantenibilidad:

La POO facilita la creación de sistemas modulares en los que cada clase u objeto encapsula datos y funciones específicas. Esta modularidad permite gestionar, actualizar y ampliar el código de manera efectiva, lo que es esencial para un MVP en constante evolución.

Implementación del Patrón Modelo-Vista-Controlador (MVC):

El enfoque de POO simplifica la aplicación del patrón MVC, el cual separa claramente la lógica de negocio, la presentación y el control de la interacción. Esta separación mejora la organización del sistema y fomenta la reutilización de componentes, aspectos clave para mantener la escalabilidad y la coherencia del proyecto.

Análisis de Ventajas y Desventajas

Ventajas:

Robustez y Estabilidad: Java ofrece un entorno de desarrollo confiable, probado en aplicaciones empresariales.

Amplio Soporte Tecnológico: La disponibilidad de frameworks y herramientas mejora la eficiencia del desarrollo y la integración de sistemas.

Facilidad de Implementación de MVC: La orientación a objetos facilita la estructuración del código en capas, permitiendo una clara división de responsabilidades.

Integración con Procesos de CI/CD: La compatibilidad con herramientas de automatización asegura un ciclo de desarrollo ágil y seguro.

Desventajas:

Complejidad en la Gestión de Recursos: Java puede requerir un manejo cuidadoso de la memoria y otros recursos, lo que implica una mayor atención en la optimización del rendimiento.

Verbosidad en el Código: La naturaleza de Java puede resultar en un código más extenso, lo que a veces puede alargar los tiempos de desarrollo si no se gestionan adecuadamente.

Relación con los Requerimientos del Proyecto

La elección de Java y la adopción de la Programación Orientada a Objetos se alinean estrechamente con los objetivos del MVP de EduTech IA. La robustez y escalabilidad de Java aseguran un sistema estable y adaptable, mientras que la modularidad y mantenibilidad ofrecidas por la POO facilitan la implementación del patrón MVC. Esto permite desarrollar una plataforma capaz de personalizar cursos y contenidos de manera eficiente y segura, respondiendo a las necesidades específicas de cada usuario.