

Selección del Lenguaje y Paradigma de Programación

Proyecto MVP – EduTech IA

Introducción

El presente documento tiene como objetivo justificar la elección del lenguaje de programación y el paradigma de desarrollo más adecuados para el MVP de la plataforma digital de EduTech IA según lo pedido en el enunciado de la actividad. La selección se realiza considerando criterios de robustez, escalabilidad, integración continua y seguridad, aspectos clave para garantizar la viabilidad técnica y la eficiencia operativa del sistema.

Lenguaje de Programación Seleccionado: Java

Justificación:

Madurez y Robustez: Java es un lenguaje consolidado en el ámbito empresarial, conocido por su estabilidad y robustez. Su larga trayectoria y amplio uso en aplicaciones de misión crítica lo posicionan como una opción confiable.

Amplio Ecosistema y Comunidad: La disponibilidad de frameworks, bibliotecas y herramientas (como Spring, Hibernate, Maven) facilita el desarrollo rápido y de calidad. Además, el soporte comunitario y la documentación extensa permiten solucionar problemas de forma eficiente.

Portabilidad y Escalabilidad: Gracias a la JVM (Java Virtual Machine), las aplicaciones desarrolladas en Java son altamente portables entre diferentes plataformas, lo cual es esencial para la escalabilidad del proyecto.

Integración CI/CD: Java se integra fácilmente en pipelines de integración y entrega continua, favoreciendo la automatización de pruebas y despliegues, aspectos críticos para iteraciones ágiles y seguras.

Paradigma de Programación Seleccionado: Programación Orientada a Objetos (OOP)

Justificación:

Modularidad y Mantenibilidad: La orientación a objetos permite diseñar sistemas modulares, donde cada componente (clase u objeto) encapsula datos y comportamientos, facilitando el mantenimiento y la evolución del código.

Facilidad para la Implementación del Patrón MVC: El paradigma OOP es ideal para estructurar la aplicación siguiendo la arquitectura Modelo-Vista-Controlador, permitiendo una separación clara de responsabilidades y promoviendo la reutilización de código.

Escalabilidad y Flexibilidad: La utilización de clases y objetos facilita la ampliación del sistema a medida que se incorporen nuevos módulos o funcionalidades, garantizando que la aplicación pueda crecer sin perder cohesión.

Aspecto	Ventajas	Desventajas
Java y OOP	<ul style="list-style-type: none"> - Alta robustez y estabilidad en entornos productivos. - Gran cantidad de frameworks y herramientas. - Excelente soporte para CI/CD. - Permite diseño modular y escalable. 	<ul style="list-style-type: none"> - Verbosidad en el código, lo que puede aumentar el tiempo de desarrollo. - Requiere de un manejo adecuado de la memoria y recursos.
Compatibilidad con MVC	<ul style="list-style-type: none"> - Facilita la separación de la lógica de negocio, presentación y control, optimizando el mantenimiento y la evolución del sistema 	<ul style="list-style-type: none"> - La implementación incorrecta del patrón MVC puede generar sobrecarga en el diseño si no se sigue una buena práctica.

Relación con los Requerimientos del Proyecto

El uso de Java y el paradigma OOP se alinea directamente con los requerimientos de EduTech IA, ya que:

Seguridad y Confiabilidad: La robustez de Java permite implementar protocolos de seguridad eficientes y minimizar vulnerabilidades en todas las fases del desarrollo.

Escalabilidad y Mantenimiento: La modularidad y la separación de responsabilidades propias de OOP facilitan la gestión de entornos (desarrollo, preproducción y producción) y la escalabilidad del sistema.

Integración y Despliegue Continuo: El ecosistema de Java facilita la integración con herramientas de CI/CD, garantizando despliegues automáticos y eficientes que cumplen con los altos estándares de calidad y rapidez requeridos.