**Selección del Lenguaje y Paradigma de Programación**

Dado el objetivo y enfoque del proyecto, el cual es crear una plataforma de aprendizaje en la cual alumnos con su perfil propio interactúen con cursos de distintos tipos, el paradigma más adecuado sería orientado a objetos, y un lenguaje como Java o C++. En este caso usare el lenguaje Java.

**Elección de Lenguaje**

Java es un lenguaje robusto y maduro, ampliamente utilizado en el desarrollo de aplicaciones y en el entorno empresarial. Esto significa que la colaboración con otros programadores será más fácil, ya que es más probable que estos tengan experiencia con el lenguaje. Debido a su madurez, posee una amplia gama de librerías, como SpringBoot, para facilitar la creación de aplicaciones web, u otras funcionalidades. También es un lenguaje que se adapta bien al paradigma de programación que he escogido.

Sin embargo, Java también tiene sus inconvenientes. No es el lenguaje más simple y legible, y es más intensivo en cuanto a recursos comparado con otros lenguajes más ligeros. Además, las aplicaciones que involucran el uso de la inteligencia artificial son más comúnmente desarrolladas con Python, debido a su integración con librerías de IA robustas, como PyTorch.

**Elección del Paradigma de Programación**

El paradigma orientado a objetos permite estructurar el código en clases, las cuales representan entidades reales del problema, como pueden ser los cursos o los alumnos. Estas clases simplifican el código y más adelante se pueden reutilizar. La orientación a objetos también nos permite ocultar la funcionalidad interna compleja de los objetos, y ofrecerle al usuario tan solo una interfaz publica, lo cual mejora la claridad del diseño.

Otro beneficio del paradigma orientado a objetos es su alta flexibilidad y escalabilidad debido a sus características de herencia y polimorfismo. Una plataforma de un startup como este necesita adaptarse contantemente a nuevas necesidades, como puede ser actualizaciones al modelo de IA, o la oferta de nuevos cursos. Esto se puede implementar con nuevas clases (ej. nuevos cursos) que hereden funcionalidades del modelo existente.