# HERRAMIENTAS Y AMBIENTE DE DESARROLLO

* **Metodología de Desarrollo:**

Para el desarrollo del proyecto EduTech INNOVATORS SPA, se ha optado por un marco de trabajo híbrido basado en Scrum y Kanban (Scrumban). Esta elección permite una planificación iterativa y colaborativa, propia de Scrum, junto con un enfoque visual y flexible en la gestión de tareas, característico de Kanban.

La metodología Scrumban facilita la adaptación constante a los cambios en los requerimientos, especialmente útil en proyectos donde el backend, la estructura de datos y la interfaz del usuario evolucionan simultáneamente.

Durante el desarrollo, se organizaron las tareas en tableros visuales (Trello) y se aplicaron ciclos de trabajo cortos (sprints), que permitieron construir módulos funcionales del sistema de forma incremental, tales como: gestión de cursos, evaluaciones, preguntas y respuestas.

* **Herramientas y/o Técnicas de desarrollo:** El stack de desarrollo y las herramientas a utilizar son las siguientes:

| **Área** | **Herramienta / Tecnología** |
| --- | --- |
| Lenguaje Frontend | HTML, CSS, JavaScript |
| Framework Principal | Spring Boot |
| Estilos | sessionStorage y JavaScript para mantener el estado de usuario y navegación. |
| Manejo de Estados |  |
| Backend | Java + Spring Boot |
| Base de Datos | MySQL (vía Spring Data JPA) |
| Control de Versiones | Git + GitHub |
| Gestión de Proyecto | Trello |
| Entorno de Desarrollo | VS Code |
| Testing | Postman para backend y Jest + Testing Library (Pruebas de Componentes) |
| Integración Continua | Git Action |
| Despliegue | Localhost (puerto 8081, 8082,8083) |
| Modelado UML | Draw IO |

MODELADO DEL SOFTWARE

Para el diseño arquitectónico del sistema EduTech Innovators SPA, se ha optado por utilizar el modelo 4+1 de vistas arquitectónicas propuesto por Philippe Kruchten. Esta decisión responde a la necesidad de estructurar la arquitectura del sistema educativo en línea de manera clara, escalable y comprensible para los distintos actores involucrados: desarrolladores, docentes, administradores y usuarios finales.

El modelo 4+1 permite abordar la arquitectura desde cinco perspectivas complementarias:

* Vista lógica: Describe los principales módulos del sistema como Gestión de Usuarios, Gestión Académica, Pagos y Certificación, representando cómo se relacionan los objetos y funcionalidades clave.
* Vista de desarrollo: Muestra la organización interna del código fuente, dividido en microservicios desarrollados con Spring Boot, cada uno con responsabilidades específicas y desacopladas.
* Vista de procesos: Enfocada en los aspectos dinámicos del sistema, como la concurrencia de usuarios, la comunicación entre servicios mediante APIs REST, y la escalabilidad de los componentes más críticos como el servicio de pagos y evaluaciones.
* Vista física o de despliegue: Describe la distribución del sistema en infraestructura real, incluyendo contenedores Docker desplegados en AWS EC2, almacenamiento en S3, bases de datos en RDS PostgreSQL, y balanceadores para alta disponibilidad.
* Vista de casos de uso: Representa los escenarios más relevantes como la inscripción de estudiantes, la evaluación automatizada, el seguimiento del progreso académico y la generación de certificados, permitiendo validar la coherencia de las demás vistas.

#### Justificación de uso:

* Comunicación efectiva: Cada vista está orientada a distintos stakeholders, lo que facilita el entendimiento transversal del sistema: desde docentes hasta el equipo técnico.
* Documentación clara y estructurada: Permite representar desde los módulos lógicos y el diseño de clases, hasta los nodos físicos de despliegue, incluyendo cómo fluye la información en tiempo de ejecución.
* Validación del diseño: Los escenarios de uso (como inscripciones y pagos) permiten verificar que la integración de los componentes responde correctamente a los objetivos funcionales.
* Adaptabilidad tecnológica: Este modelo es compatible con el enfoque moderno de desarrollo utilizado en EduTech, donde se combina frontend React, backend con microservicios Java Spring Boot y servicios cloud.

En conclusión, el uso del modelo de vistas 4+1 en EduTech garantiza una visión integral y coherente del sistema, útil no solo para su diseño e implementación, sino también para su documentación técnica, escalabilidad futura y mantenimiento continuo

*Modelo de vistas 4+1*

| **Vista del modelo 4+1** | **Diagramas UML recomendados** |
| --- | --- |
| **Vista Lógica** | Diagrama de Clases |
| **Vista de Implementación** | Diagrama de Componentes |
| **Vista de Procesos** | Diagrama de Actividades |
| **Vista de Despliegue** | Diagrama de Despliegue |
| **Vista de Escenarios (Casos de Uso)** | Diagrama de Casos de Uso |

Para aplicar el modelo se diseñará:

**PATRÓN DE DISEÑO DE SOFTWARE / ARQUITECTURA DE DISEÑO**

Definir entre:

Patrón por capas.

MVC

Microservicios

Monolítico

Cliente Servidor

Maestro Esclavo

Entre otros.