# **Formulario: Vista de Desarrollo del Proyecto de Software**

**Proyecto:** KineCap  
**Grupo:** 3  
**Fecha:** Viernes 4 de Octubre del 2024

**1. Estructura del sistema**

Describe la estructura del sistema de tu proyecto, mencionando los módulos principales y cómo interactúan entre sí. Puede usar diagramas de arquitectura para complementar tu descripción.

**Descripción de la estructura:**

El sistema está organizado en módulos principales que interactúan con una base de datos central. Los módulos incluyen la autenticación de usuarios, la gestión de cursos, el sistema de pagos y la emisión automática de certificados. Estos módulos se comunican entre sí para permitir una experiencia fluida para los usuarios. La interfaz de usuario se conecta al backend para manejar la autenticación, procesar pagos y administrar la información del usuario y los cursos.

**Diagrama de la estructura del sistema:**   
*(Adjunta un diagrama UML o un diagrama de bloques que muestre los módulos principales y sus interacciones.)*

**2. Módulos principales**

Enumera y describe los módulos principales del sistema, explicando sus responsabilidades y cómo interactúan con otros módulos.

**Módulo 1 (Autenticación):**

**Responsabilidad:** Gestionar el registr**Módulo 2 (Gestión de cursos):**

o e inicio de sesión de los usuarios.

**Interacción con otros módulos:** Este módulo verifica las credenciales del usuario y permite el acceso a los demás módulos del sistema.

**Responsabilidad:** Administrar la inscripción a cursos, el seguimiento del progreso y la visualización del contenido de los cursos.

**Interacción con otros módulos:** Este módulo se comunica con la base de datos para obtener información sobre los cursos y permite la interacción del usuario.

**Módulo 3 (Sistema de pago):**

**Responsabilidad:** Gestionar los pagos en línea y proporcionar acceso a los cursos tras el pago.

**Interacción con otros módulos:** Este módulo actualiza el estado de los usuarios en la base de datos después de un pago exitoso y permite el acceso a los cursos pagados.

**3. Dependencias y Librerías**

Especifica las dependencias y librerías/frameworks que estás utilizando en tu proyecto, indicando para qué funcionalidad o módulo se usan.

**Dependencias y Librerías:**

* **Flask/Django:** Frameworks para el desarrollo backend y manejo de seguridad.
* **PostgreSQL:** Sistema de gestión de bases de datos para almacenar usuarios, cursos y certificados.
* **OAuth:** Sistema de autenticación seguro.
* **MercadoPago/Stripe API:** Integración para la gestión de pagos.

**4. Organización del Código**

Explica cómo está organizado el código en términos de carpetas y paquetes dentro del proyecto. Incluye ejemplos o un esquema de la estructura de directorios.

**Organización del código:**

De momento, a modo de guía, tenemos más o menos estructurada la forma y carpetas que contendrán los diversos códigos y componentes de nuestro sistema. Por ahora la estructura sería la siguiente:

* **/auth:** Gestión de autenticación de usuarios.
* **/courses:** Administración de cursos y seguimiento del progreso.
* **/payments:** Gestión de pagos.
* **/static:** Archivos de frontend (HTML, CSS, JS).
* **/templates:** Vistas de la interfaz de usuario.

*(Opcional: Adjunta una imagen o diagrama de la estructura de directorios del proyecto.)*

**5. Reglas de Integración**

Especifica las reglas que tu equipo sigue para la integración del código entre los módulos. Esto puede incluir restricciones sobre el acceso a datos, convenciones de desarrollo, etc.

**Reglas de integración:**

* Uso de Git como sistema de control de versiones, este es el principal programa donde estarán los repositorios .
* Convenciones estrictas para nombres de variables, métodos y clases, dejar claro desde el inicio los identificadores de datos..
* Todas las integraciones entre módulos pasan por revisiones de código y pruebas antes de fusionarse en la rama principal,esto para evitar problemas o errores.

**6. Prácticas de desarrollo**

Detalla las prácticas de desarrollo que estás utilizando en el proyecto, como las pruebas unitarias, integración continua, revisión de código, etc.

**Prácticas de desarrollo:**

En el desarrollo usaremos diferentes prácticas las principales son:

* Pruebas unitarias para cada módulo antes de la integración.
* Revisión de código entre los miembros del equipo para asegurar calidad y consistencia.
* Uso de integración continua para asegurar que los módulos funcionan correctamente de manera conjunta.

**7. Herramientas de desarrollo**

Mencione las herramientas que está utilizando para el desarrollo del proyecto (por ejemplo, IDE, sistemas de control de versiones, etc.).

**Herramientas:**

* **IDE:** Visual studio Code, Vim
* **Control de versiones:** GitHub.
* **Herramientas de pruebas:** Postman y JMeter para pruebas de API y rendimiento.

**8. Diagrama de Arquitectura del Sistema**

Adjunta un diagrama de arquitectura general del sistema que incluye los módulos, las interacciones entre ellos y las tecnologías involucradas.

**Diagrama de arquitectura general:**   
*(Adjunta el diagrama en esta sección)*