

Documento de diseño
Learning Recommendation System
Integrantes: Emile Bottagisio 202223504

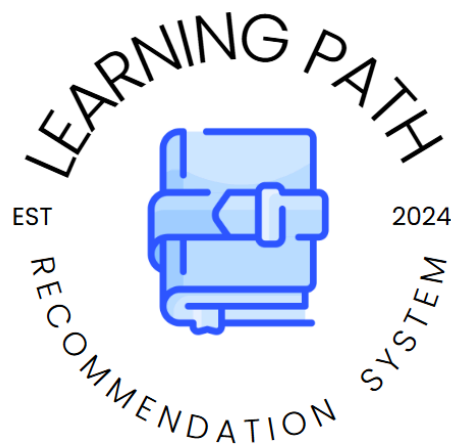
Nivel 1: Visión General del Sistema

1. Descripción General del Sistema

El sistema de recomendación de rutas de aprendizaje es una aplicación diseñada para ayudar a los usuarios a encontrar un camino personalizado para alcanzar sus objetivos educativos. Este sistema utiliza la información del usuario, como sus intereses, nivel de conocimiento previo y metas de aprendizaje, para generar una serie de módulos o cursos que mejor se adapten a sus necesidades.

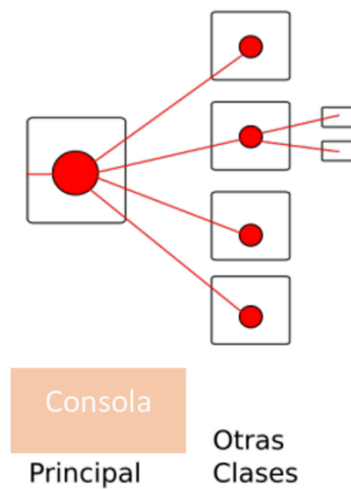
El propósito principal es optimizar la experiencia de aprendizaje de cada usuario al ofrecer un camino claro y estructurado que se adapte a sus preferencias y habilidades. Además, el sistema permitirá a los usuarios realizar un seguimiento de su progreso, revisar los temas cubiertos y recibir recomendaciones adicionales según sus avances.

El sistema se presenta como una aplicación web accesible desde cualquier dispositivo con conexión a internet, con una interfaz amigable y de fácil navegación. Su límite, o frontera, se establece en la capacidad de recomendar rutas basadas en los datos proporcionados, sin intervención manual por parte de tutores o educadores.



2. Estilos de control

Antes de iniciar con el proceso de diseño se definen las funcionalidades de alto nivel que el sistema debe estar en capacidad de satisfacer. Lo anterior se realiza con el propósito de que, independientemente de la implementación de la consola, sea apropiada para el entendimiento del usuario. En cuanto a la interacción con la consola, el usuario le corresponde según su rol, profesor o estudiante, otorgar las entradas acordes. La consola por su parte le brinda al usuario la respuesta a los requerimientos. Lo que sería un contrato en Java, que es una forma de especificar las expectativas de entrada y salida para un método o función. Esto garantiza que los datos se manejen correctamente y que se produzcan los resultados esperados. Con esta explicación, el estilo de control que estamos utilizando es centralizado.



3. Requisitos Funcionales (RF)

RF1: El sistema debe permitir a los usuarios iniciar sesión con nombre de usuario y contraseña.

RF2: El sistema debe permitir a los usuarios registrarse y crear un perfil, incluyendo información personal y preferencias de aprendizaje.

RF3: El sistema debe generar rutas de aprendizaje personalizadas basadas en la información ingresada por los usuarios.

RF4: El sistema debe permitir a los usuarios visualizar y realizar un seguimiento de su progreso a lo largo de la ruta de aprendizaje.

RF5: El sistema debe proporcionar recomendaciones adicionales basadas en el avance y desempeño del usuario.

RF6: El sistema debe permitir la actualización de la información del perfil del usuario en cualquier momento.

RF7: El sistema debe permitir a los usuarios marcar módulos o cursos como completados.

4. Requisitos No Funcionales (RNF)

RNF1: El sistema debe ser accesible desde navegadores web modernos y dispositivos móviles.

RNF2: El sistema debe tener tiempos de respuesta inferiores a 2 segundos para la mayoría de las operaciones del usuario.

RNF3: El sistema debe garantizar la seguridad de los datos personales de los usuarios, incluyendo el cifrado de contraseñas.

RNF4: El sistema debe tener una interfaz amigable e intuitiva, diseñada para facilitar la navegación incluso para usuarios sin experiencia técnica.

RNF5: El sistema debe ser escalable para soportar un número creciente de usuarios sin degradar el rendimiento.

5. Restricciones

- El sistema en un futuro se integrará con una plataforma LMS como Bloque Neón, pero nos dicen que nosotros no debemos realizar esta integración, por tanto, las plataformas LMS no irían en el UML.
- Los estudiantes tienen acceso a ciertas actividades que tienen un resultado, en su mayoría “Completada” o “No completada”. Si es un estudiante marca que fue “Completada” el sistema le cree, es decir que no debemos hacer una verificación minuciosa de que esto sea verdad.
- La carpeta en la que esté la persistencia no debe ser la misma que la del código fuente.
- El sistema debe estar diseñado con separación de responsabilidades permitiendo la fácil adición de nuevas funcionalidades en el futuro, como la integración con nuevas plataformas o la ampliación del catálogo de actividades disponibles para los estudiantes. Aunque esto último no está cubierto en este proyecto.
- Deben definirse casos de prueba para cada funcionalidad importante del sistema asegurando que el comportamiento sea el esperado.

6. Metodología usada en el diseño

Responsibility-Driven Design es una metodología propuesta por Wirfs-Brock y McKean para diseñar software. Está basada en tres conceptos principales: Roles, Responsabilidades y Colaboraciones. A lo largo de este documento descompondremos nuestras decisiones alrededor de estos aspectos.

Objetos/Roles

Paso 1: Identificar objetos/roles candidatos

1. Profesor: crea y administra rutas de aprendizaje.
2. Estudiante: Sigue las rutas de aprendizaje y completa actividades.
3. Learning Path: Contiene una secuencia de actividades.
4. Rating: contiene una calificación de una actividad o Learning Path.
5. Reseña: contiene un comentario sobre una actividad o Learning Path.
6. Actividad: compuesta por quices, lecturas, tareas, etc.
7. Base de Datos (DB): Almacena información de los usuarios, rutas, actividades y progreso.
8. Login: Gestiona la autenticación de los usuarios.
9. Sistema de Progreso: Monitorea y almacena el progreso de los estudiantes.
10. Evaluación: Proporciona retroalimentación y mide el éxito o el fracaso.

Paso 2: Asignar Estereotipos:

1. Profesor – Coordinator: Coordina la creación y modificación de rutas de aprendizaje y actividades.
2. Estudiante – Service provider: Sigue las rutas de aprendizaje y completa actividades.
3. Learning Path - Structurer: Estructura la secuencia de actividades y gestiona las interdependencias entre ellas.
4. Rating – Information holder: contiene una calificación de una actividad o Learning Path.
5. Reseña – Information holder: contiene un comentario sobre una actividad o Learning Path.
6. Actividad – Information holder: Mantiene información de las actividades, como la descripción, duración y objetivo.
7. Base de Datos (DB) – Information holder: Almacena información de los usuarios, rutas, actividades y progreso.
8. Login - Interfacer: Gestiona la autenticación y proporciona el acceso.
9. Sistema de Progreso – Information holder: Monitorea y almacena el progreso de los estudiantes.
10. Evaluación – Information Holder: Proporciona retroalimentación y mide el éxito o el fracaso.

Responsabilidades

Colaboraciones

7. Frontera del Sistema

Desde el punto de vista externo, el sistema ofrece las siguientes funcionalidades para los usuarios:

Registro y perfil de usuario: Los usuarios podrán registrarse en el sistema y definir su perfil, incluyendo preferencias y objetivos de aprendizaje.

Generación de rutas de aprendizaje personalizadas: A partir de la información ingresada por los usuarios, el sistema generará un plan de estudios adaptado a sus necesidades.

Seguimiento de progreso: Los usuarios podrán visualizar su progreso a lo largo de la ruta de aprendizaje, con indicadores que muestren el nivel de avance y los módulos completados.

Interfaz amigable y accesible: La aplicación ofrecerá una interfaz sencilla para que los usuarios naveguen, accedan a los cursos, y reciban recomendaciones de forma intuitiva.

No se tienen detalles sobre cómo se realiza la lógica interna del sistema (algoritmos de recomendación, almacenamiento de datos, etc.) en esta fase, pero se conoce que su frontera incluye la interacción del usuario con la interfaz para recibir recomendaciones personalizadas.

Nivel 2: Primera Descomposición

1. Componentes del Sistema

En esta fase, se realiza una descomposición del sistema en cuatro componentes principales. La suma de estos componentes permite que el sistema cumpla con todos los requisitos funcionales (RF), no funcionales (RNF) y restricciones definidos en el Nivel 1. A continuación, se describen los cuatro componentes, sus fronteras y cómo interactúan entre sí.

Componente 1: Gestión de Usuarios

Qué hace: Este componente se encarga de gestionar todo lo relacionado con los usuarios, incluyendo el registro, autenticación, actualización del perfil y almacenamiento de datos personales.

Cómo se ve: Ofrece una interfaz donde los usuarios pueden registrarse, iniciar sesión y editar su perfil. Desde el punto de vista externo, este componente permite la creación y modificación de la información del perfil.

Interacción con otros componentes: Proporciona la información de autenticación y preferencias del usuario al componente de generación de rutas de aprendizaje, y permite que el componente de seguimiento de progreso registre los avances del usuario.

Componente 2: Generación de Rutas de Aprendizaje

Qué hace: Genera rutas de aprendizaje personalizadas basadas en la información proporcionada por el usuario (intereses, conocimientos previos, objetivos, etc.).

Cómo se ve: Este componente actúa internamente para generar recomendaciones, pero desde la perspectiva del usuario, se presenta como una lista de módulos o cursos sugeridos que forman su ruta de aprendizaje.

Interacción con otros componentes: Utiliza la información del componente de Gestión de Usuarios para personalizar las rutas, y envía las recomendaciones al componente de Interfaz de Usuario para mostrarlas al usuario.

Componente 3: Seguimiento de Progreso

Qué hace: Permite realizar un seguimiento detallado del progreso del usuario en su ruta de aprendizaje, incluyendo módulos completados y recomendaciones adicionales basadas en el avance.

Cómo se ve: Desde la perspectiva del usuario, se muestra como un tablero o panel que indica el progreso actual y los logros alcanzados.

Interacción con otros componentes: Recibe la información del componente de Generación de Rutas de Aprendizaje y la utiliza para actualizar el estado del usuario. También interactúa con el componente de Gestión de Usuarios para almacenar el progreso.

Componente 4: Interfaz de Usuario

Qué hace: Se encarga de proporcionar una interfaz visual amigable y accesible que permita a los usuarios interactuar con el sistema.

Cómo se ve: Este componente es la cara visible del sistema para los usuarios. Incluye todas las pantallas de navegación, formularios de registro, paneles de progreso y las secciones de recomendaciones.

Interacción con otros componentes: Recibe datos de los componentes de Gestión de Usuarios, Generación de Rutas de Aprendizaje y Seguimiento de Progreso para mostrarlos al usuario de manera clara e intuitiva.

Nivel 3: Segunda Descomposición

En esta fase, se realiza una descomposición de cada uno de los componentes principales identificados en el Nivel 2. A continuación, se detallan los subcomponentes internos de cada uno de estos componentes, su frontera y cómo interactúan entre sí para cumplir con los compromisos del sistema.

1. Descomposición del Componente 1: Gestión de Usuarios

Subcomponente 1.1: Registro de Usuarios

Qué hace: Gestiona el proceso de registro de nuevos usuarios, validando los datos proporcionados y creando un perfil en el sistema.

Cómo se ve: Se presenta como un formulario de registro donde los usuarios ingresan sus datos personales.

Interacción con otros subcomponentes: Envía los datos al subcomponente de Almacenamiento de Datos para guardar la información del nuevo usuario.

Subcomponente 1.2: Autenticación de Usuarios

Qué hace: Gestiona el inicio de sesión de los usuarios mediante la validación de credenciales.

Cómo se ve: Se presenta como una interfaz de inicio de sesión donde el usuario ingresa sus credenciales.

Interacción con otros subcomponentes: Utiliza el subcomponente de Almacenamiento de Datos para verificar las credenciales del usuario.

Subcomponente 1.3: Actualización de Perfil

Qué hace: Permite a los usuarios actualizar la información de su perfil, incluyendo preferencias y objetivos.

Cómo se ve: Se presenta como una sección del perfil del usuario donde se pueden modificar los datos.

Interacción con otros subcomponentes: Envía las actualizaciones al subcomponente de Almacenamiento de Datos.

Subcomponente 1.4: Almacenamiento de Datos

Qué hace: Gestiona el almacenamiento y recuperación de los datos de los usuarios.

Cómo se ve: Es un subcomponente interno que no tiene una interfaz visible para el usuario.

Interacción con otros subcomponentes: Proporciona y almacena la información requerida por los subcomponentes de Registro, Autenticación y Actualización de Perfil.

2. Descomposición del Componente 2: Generación de Rutas de Aprendizaje

Subcomponente 2.1: Motor de Recomendación

Qué hace: Genera rutas de aprendizaje personalizadas a partir de los datos del usuario.

Cómo se ve: No tiene una interfaz visible para el usuario, pero sus resultados se muestran como recomendaciones en la interfaz de usuario.

Interacción con otros subcomponentes: Utiliza la información del subcomponente de Análisis de Datos para ajustar las recomendaciones según el avance del usuario.

Subcomponente 2.2: Análisis de Datos

Qué hace: Analiza la información del usuario para identificar sus necesidades y ajustar las rutas de aprendizaje.

Cómo se ve: No tiene una interfaz visible para el usuario.

Interacción con otros subcomponentes: Proporciona datos al Motor de Recomendación para personalizar las rutas de aprendizaje.

3. Descomposición del Componente 3: Seguimiento de Progreso

Subcomponente 3.1: Registro de Progreso

Qué hace: Registra el avance del usuario a medida que completa módulos o cursos.

Cómo se ve: No tiene una interfaz visible directa, pero la información se refleja en el tablero de progreso del usuario.

Interacción con otros subcomponentes: Envía datos al subcomponente de Visualización de Progreso.

Subcomponente 3.2: Visualización de Progreso

Qué hace: Muestra al usuario su progreso en la ruta de aprendizaje.

Cómo se ve: Se presenta como un tablero o panel con indicadores de avance.

Interacción con otros subcomponentes: Recibe datos del subcomponente de Registro de Progreso.

4. Descomposición del Componente 4: Interfaz de Usuario

Subcomponente 4.1: Pantallas de Navegación

Qué hace: Proporciona las diferentes pantallas para que el usuario pueda navegar por la aplicación.

Cómo se ve: Incluye menús, botones de navegación y enlaces.

Interacción con otros subcomponentes: Utiliza los datos proporcionados por los subcomponentes de otros componentes para mostrar información relevante al usuario.

Subcomponente 4.2: Formularios de Entrada

Qué hace: Permite al usuario ingresar información, como datos de registro y preferencias.

Cómo se ve: Se presenta como formularios con campos para llenar.

Interacción con otros subcomponentes: Envía la información al subcomponente correspondiente del sistema (por ejemplo, Registro de Usuarios).

Subcomponente 4.3: Panel de Recomendaciones

Qué hace: Muestra las rutas de aprendizaje recomendadas al usuario.

Cómo se ve: Se presenta como una lista o un tablero con las recomendaciones de cursos y módulos.

Interacción con otros subcomponentes: Recibe información del componente de Generación de Rutas de Aprendizaje.

Nivel 4: Descomposición Final

En esta fase, se realiza una descomposición final de cada uno de los subcomponentes identificados en el Nivel 3, hasta llegar a elementos implementables. Estos elementos generalmente serán clases que se encargan de funcionalidades específicas y que se pueden implementar directamente en el código del sistema.

1. Descomposición del Componente 1: Gestión de Usuarios

Subcomponente 1.1: Registro de Usuarios

Clase 1.1.1: Validador de Datos

Qué hace: Valida los datos proporcionados por el usuario durante el registro, asegurando que cumplan con los requisitos (por ejemplo, formato de correo electrónico, longitud de la contraseña).

Interacción con otras clases: Utilizada por la clase Controlador de Registro para validar los datos ingresados.

Clase 1.1.2: Controlador de Registro

Qué hace: Gestiona el proceso de registro, interactuando con el Validador de Datos y la clase Repositorio de Usuarios.

Interacción con otras clases: Envía los datos validados al Repositorio de Usuarios para su almacenamiento.

Clase 1.1.3: Repositorio de Usuarios

Qué hace: Almacena la información de los usuarios en la base de datos.

Interacción con otras clases: Proporciona acceso a los datos del usuario a otras clases dentro del sistema.

Subcomponente 1.2: Autenticación de Usuarios

Clase 1.2.1: Validador de Credenciales

Qué hace: Valida las credenciales ingresadas por el usuario durante el inicio de sesión.

Interacción con otras clases: Utilizada por el Controlador de Autenticación.

Clase 1.2.2: Controlador de Autenticación

Qué hace: Gestiona el inicio de sesión y la autenticación de los usuarios.

Interacción con otras clases: Interactúa con el Repositorio de Usuarios para verificar las credenciales.

Subcomponente 1.3: Actualización de Perfil

Clase 1.3.1: Controlador de Perfil

Qué hace: Permite al usuario actualizar su perfil, interactuando con el Validador de Datos y el Repositorio de Usuarios.

Interacción con otras clases: Utiliza el Validador de Datos para verificar las actualizaciones y el Repositorio de Usuarios para guardar los cambios.

2. Descomposición del Componente 2: Generación de Rutas de Aprendizaje

Subcomponente 2.1: Motor de Recomendación

Clase 2.1.1: Algoritmo de Recomendación

Qué hace: Contiene la lógica para generar recomendaciones personalizadas en base a los datos del usuario.

Interacción con otras clases: Utiliza la clase Análisis de Datos para ajustar las recomendaciones según el avance del usuario.

Clase 2.1.2: Controlador de Recomendaciones

Qué hace: Gestiona la interacción entre el Algoritmo de Recomendación y la interfaz de usuario.

Interacción con otras clases: Envía las recomendaciones generadas al Panel de Recomendaciones.

Subcomponente 2.2: Análisis de Datos

Clase 2.2.1: Analizador de Progreso

Qué hace: Analiza el progreso del usuario para ajustar las recomendaciones.

Interacción con otras clases: Proporciona datos al Algoritmo de Recomendación.

3. Descomposición del Componente 3: Seguimiento de Progreso

Subcomponente 3.1: Registro de Progreso

Clase 3.1.1: Controlador de Registro de Progreso

Qué hace: Gestiona el registro del avance del usuario en los módulos.

Interacción con otras clases: Almacena el progreso en el Repositorio de Usuarios.

Subcomponente 3.2: Visualización de Progreso

Clase 3.2.1: Generador de Tablero de Progreso

Qué hace: Genera el tablero de progreso para ser mostrado al usuario.

Interacción con otras clases: Recibe los datos del Controlador de Registro de Progreso.

4. Descomposición del Componente 4: Interfaz de Usuario

Subcomponente 4.1: Pantallas de Navegación

Clase 4.1.1: Controlador de Navegación

Qué hace: Gestiona la navegación entre las diferentes pantallas de la aplicación.

Interacción con otras clases: Recibe comandos del usuario y muestra la pantalla correspondiente.

Subcomponente 4.2: Formularios de Entrada

Clase 4.2.1: Controlador de Formularios

Qué hace: Gestiona los datos ingresados por el usuario a través de los formularios.

Interacción con otras clases: Envía los datos ingresados a los correspondientes controladores del sistema.

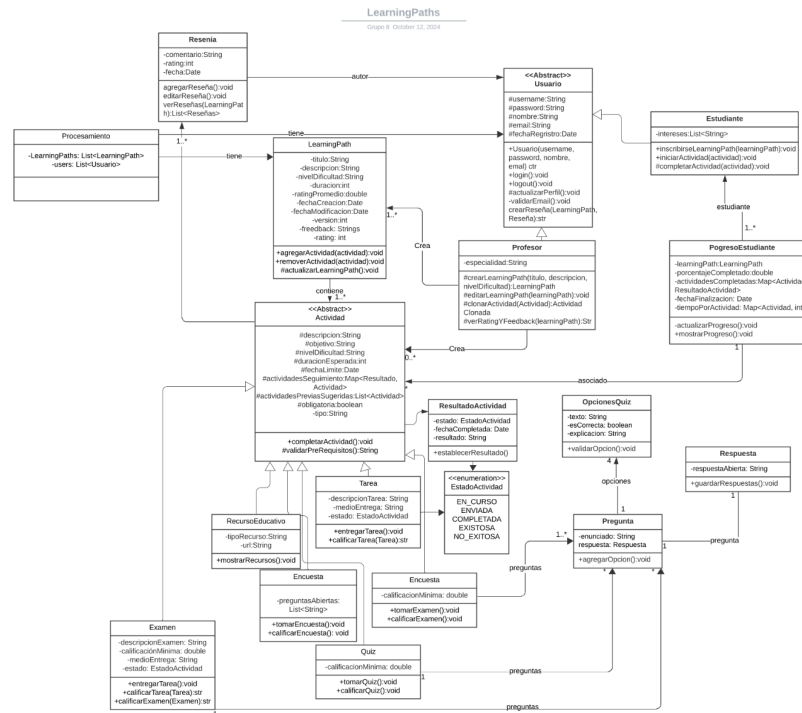
Subcomponente 4.3: Panel de Recomendaciones

Clase 4.3.1: Generador de Panel de Recomendaciones

Qué hace: Muestra las rutas de aprendizaje recomendadas al usuario.

Interacción con otras clases: Recibe datos del Controlador de Recomendaciones para mostrar las sugerencias al usuario.

Diagrama de clases



Historias de usuario:

1. Estudiante: Modelamos el rol de estudiante para que se pueda inscribir en un learning path y hacer las actividades recomendadas.

Como estudiante quiero poder ver las actividades asignadas por mi profesor, para poder desarrollarlas a tiempo.

Como estudiante, quiero inscribirme en un learning path para acceder al contenido y quices para mejorar mis conocimientos y habilidades en el tema.

Como estudiante quiero acceder a mi perfil para ver el progreso y las calificaciones obtenidas en los quices para saber en qué áreas necesito mejorar.

Como estudiante quiero realizar un quiz de verdadero o falso al finalizar un módulo para evaluar la comprensión del contenido.

2. Profesor: Modelamos el rol de profesor para que pueda crear, interactuar con los learning paths, verificar el progreso de los estudiantes, asignar tareas, etc.

Como profesor quiero crear un nuevo learning path con módulos y quices de verdadero y falso para guiar a los estudiantes en su proceso de aprendizaje de manera estructurada.

Como Profesor quiero editar y actualizar el contenido de un learning path [Para] asegurar que el material esté actualizado y relevante para los estudiantes.

Como Profesor quiero acceder a un reporte del progreso de cada estudiante en mis learning paths para hacer seguimiento de su desempeño y brindar retroalimentación personalizada.

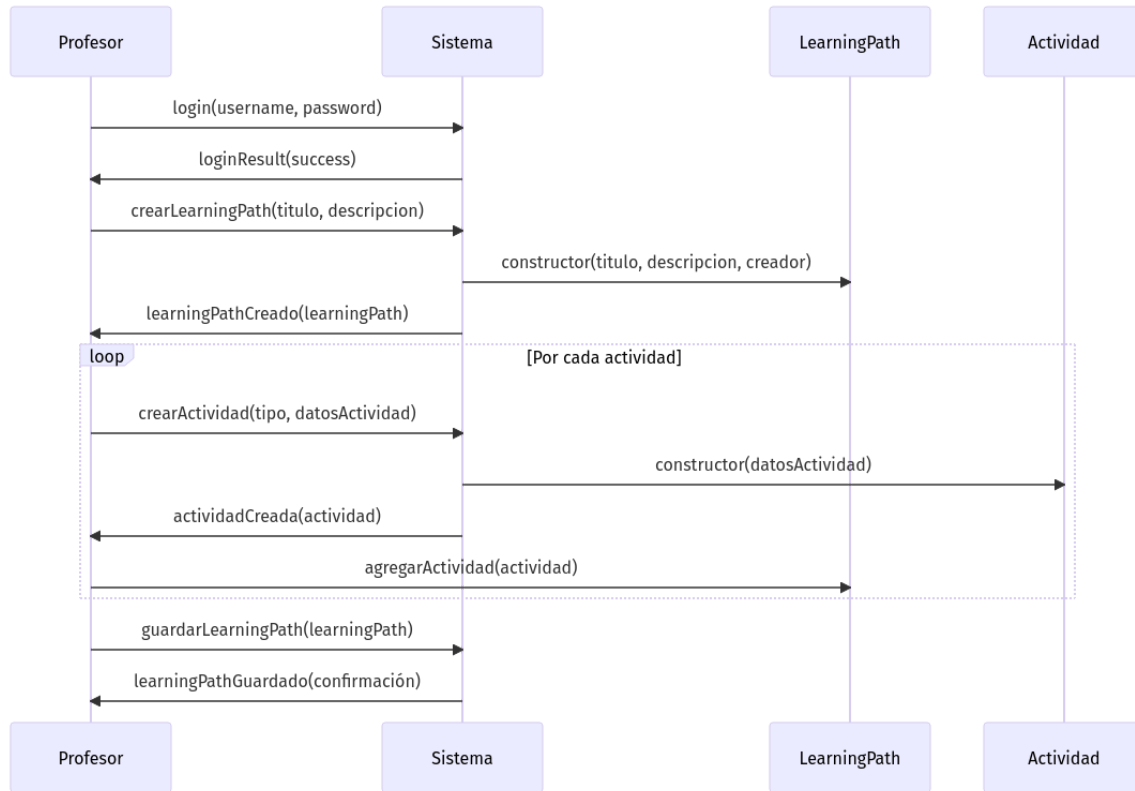
Como Profesor quiero ver los resultados de los quices de los estudiantes para identificar áreas de dificultad común y ajustar el contenido o quices en función de esas necesidades.

Como Profesor quiero configurar los quices para que se califiquen automáticamente para ahorrar tiempo en la revisión y proporcionar retroalimentación inmediata a los estudiantes.

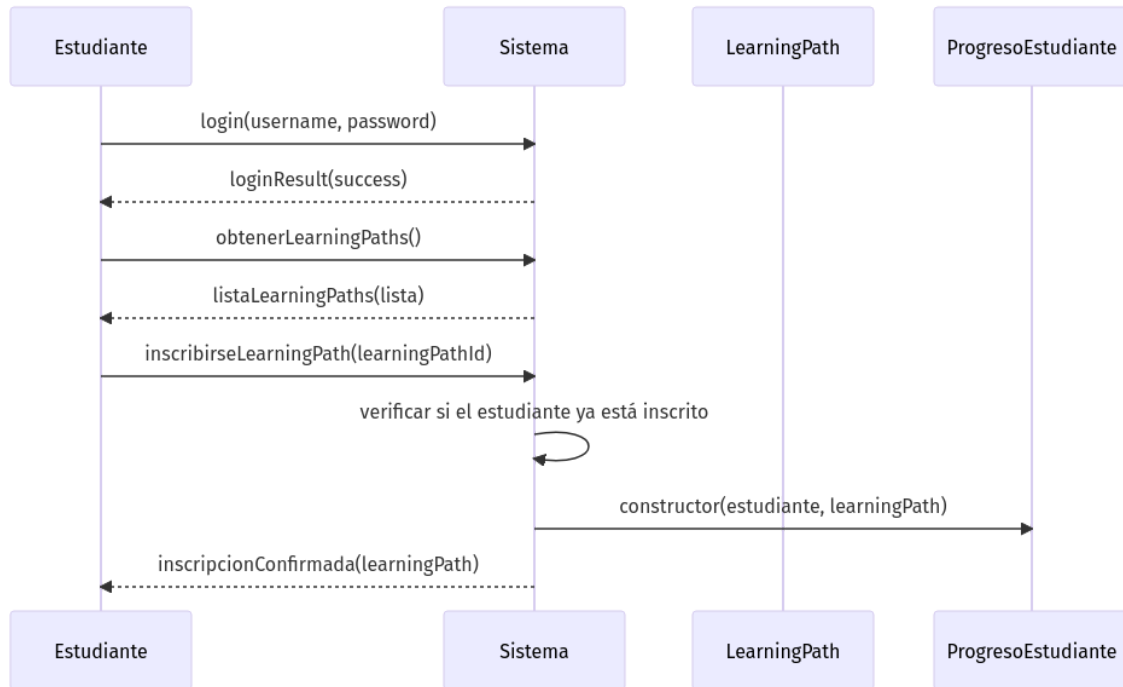
Diagrama de clases de alto nivel

Diagrama de secuencia

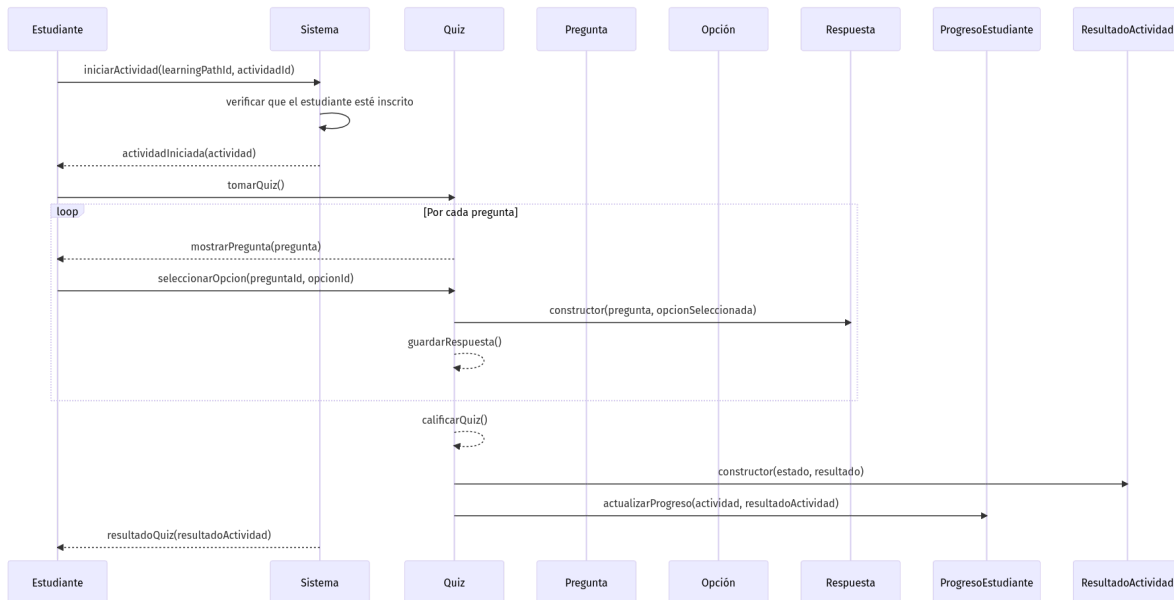
Funcionalidad 1: Profesor crea un Learning Path



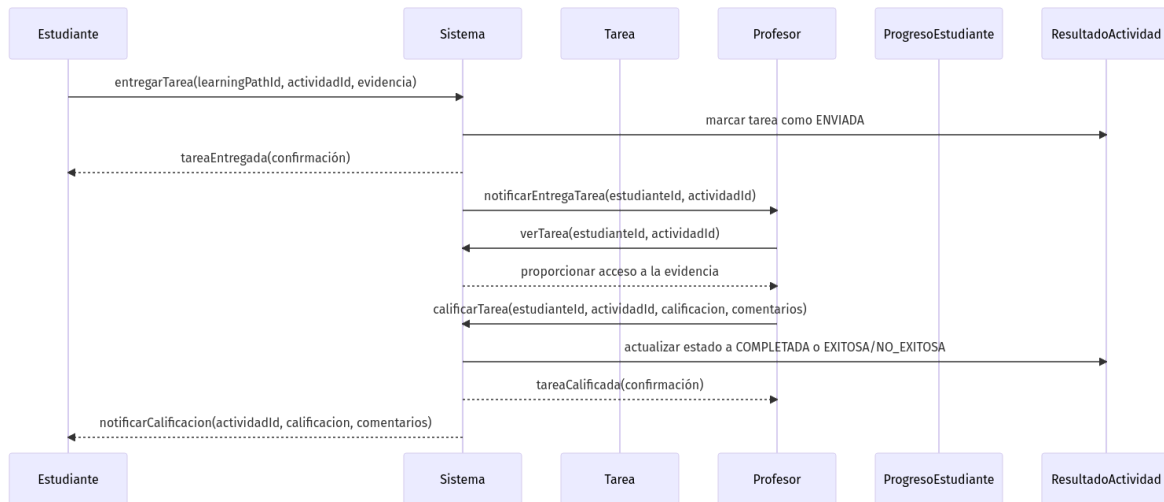
Funcionalidad 2: Estudiante se inscribe en un Learning Path.



Funcionalidad 3: Estudiante completa una actividad (Quiz).



Funcionalidad 4: Profesor califica una tarea enviada por el estudiante.



Funcionalidad 5: Estudiante recibe recomendaciones basadas en su progreso.

