

Alejandro Carvajal
Victoria Merizalde
Santiago Arias

En el análisis del problema se definieron los siguientes requerimientos funcionales para la implementación de este:

- Como estudiante quiero poder inscribirme a un Learning path donde pueda escoger las actividades que deseo completar.
- Como estudiante quiero poder marcar como completadas las actividades y revisar mis resultados tras completarla.
- Como estudiante quiero tener un rastreo de mi progreso al llevar a cabo diferentes actividades.
- Como estudiante quiero recibir recomendaciones sobre las actividades que debería llevar a cabo.
- Como profesor quiero poder crear Learning paths con sus respectivas actividades para que mis estudiantes puedan desarrollar distintas habilidades.
- Como profesor quiero llevar seguimiento del progreso de mis estudiantes en los distintos Learning Paths.
- Como profesor quiero poder acceder al feedback de mis estudiantes para mejorar el contenido de mis actividades.

A partir de estos requerimientos se construyeron los siguientes roles y responsabilidades para los distintos elementos del problema:

Asignación de roles:

1.Information holder: mantiene y entrega información:

- Feedback
- Progreso
- Quiz
- Encuesta
- Examen
- Recurso educativo
- tarea

2. Structurer: mantiene las relaciones entre objetos y provee información sobre esas relaciones - Learning Path

3. Service provider: hace algún trabajo para los demás (ofrece un servicio complejo)

- Authenticator
- Pregunta

4. Coordinator: su principal responsabilidad es delegar tareas a otros componentes

- Actividad
- Usuario

5. Controller: toma decisiones importantes y controla las acciones de otros componentes

- Estudiante
- Profesor

Para poder cumplir con los requerimientos del problema se debieron asignar las siguientes responsabilidades a sus componentes:

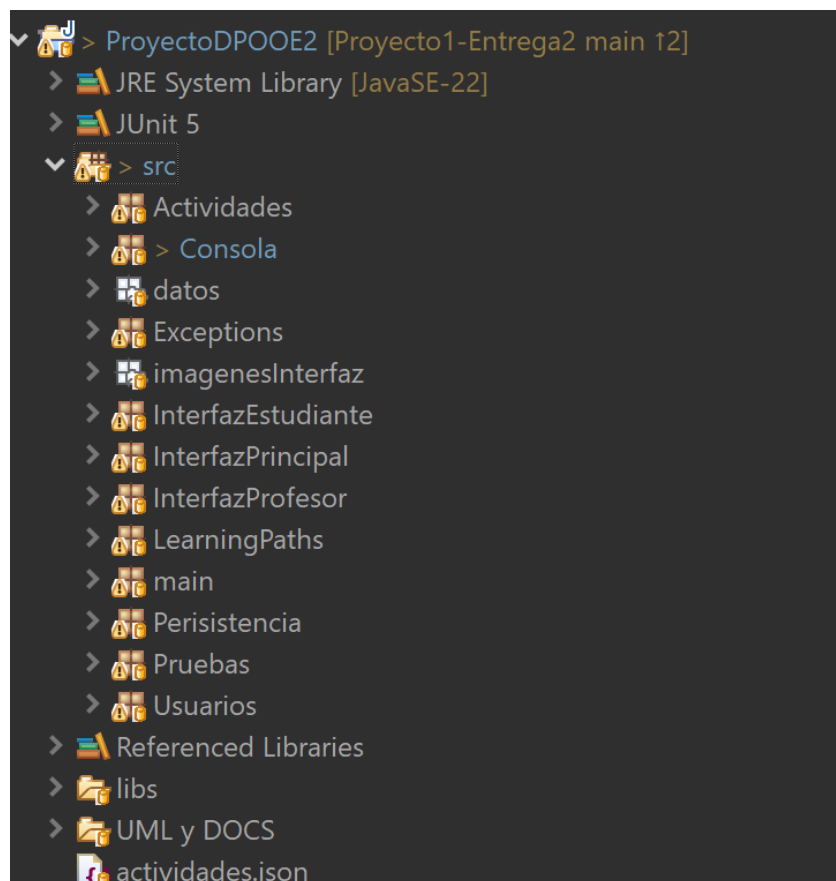
Responsabilidades:

Responsabilidad	Elemento encargado
Crear las preguntas y los recursos que se implementan dentro de una actividad para poder llevarla a cabo.	Profesor
Identificar el tipo de usuario para saber que tipo de acceso tiene a las distintas clases	Usuario
Permitir el acceso a un usuario determinado a las experiencias de aprendizaje	Authenticator
Establecer recomendaciones de Learning paths para un estudiante según sus intereses.	Profesor y Learning path
Inscribirse a un Learning Path	Estudiante
Completar las actividades del learning Path y marcarlas como enviadas.	Estudiante
Cambiar el estado de una actividad dependiendo de su resultado.	Profesor
Rastrear las actividades y progreso de los estudiantes en ellas.	Progreso
Rastrear el progreso de un estudiante dentro de un Learning Path	Progreso
Proporcionar información de los logros y avances del estudiante	Progreso

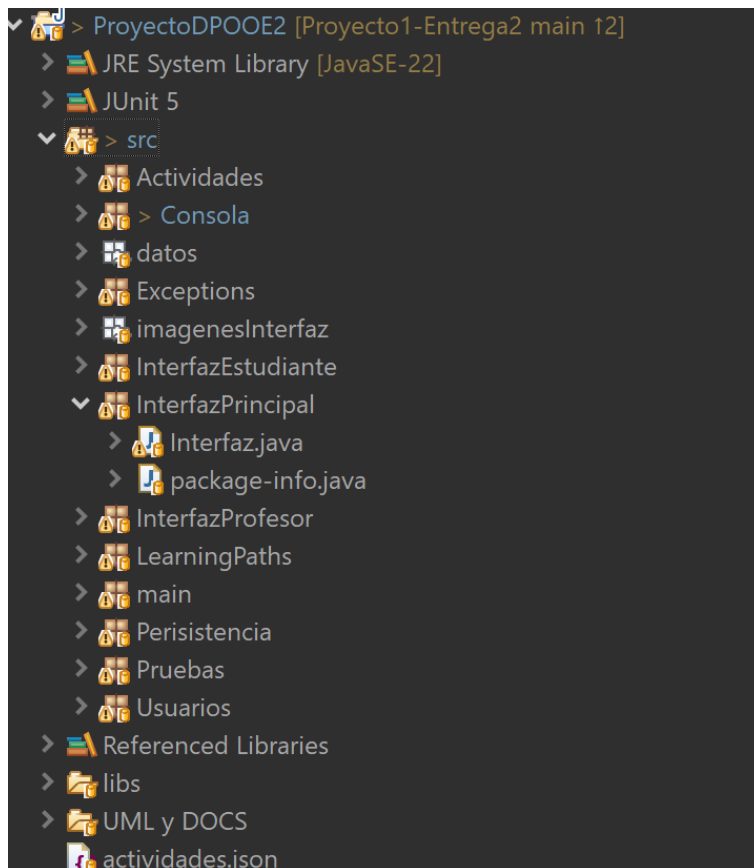
Advertir a un estudiante si va a completar una actividad sin sus pre requisitos	Learning Path y Actividad
Sugerir actividades previas y de seguimiento a una actividad dependiendo del desempeño del estudiante.	Learning Path y Actividad
Establecer la fecha límite para completar una actividad, dependiendo de la actividad previa	Actividad
Proporcionar feedback sobre el learning path	Estudiante
Revisar el feedback de un Learning Path para poder mejorarlo.	Profesor

Con las responsabilidades determinadas anteriormente, se creó el siguiente diagrama que relaciona los elementos entre ellos para que puedan cumplir con sus respectivas responsabilidades:

Funcionamiento general del proyecto, guía base:



Se encontrará en el proyecto el layout de la imagen, el cual incluye: paquetes como las actividades, la consola que se adaptó a la interfaz, los JSONs, excepciones, imágenes de la interfaz para hacerla más amena y las tres interfaces. Por otro lado, ya se tiene el mundo como tal, el cual incluye los learningPaths, la persistencia, los usuarios y las pruebas unitarias.

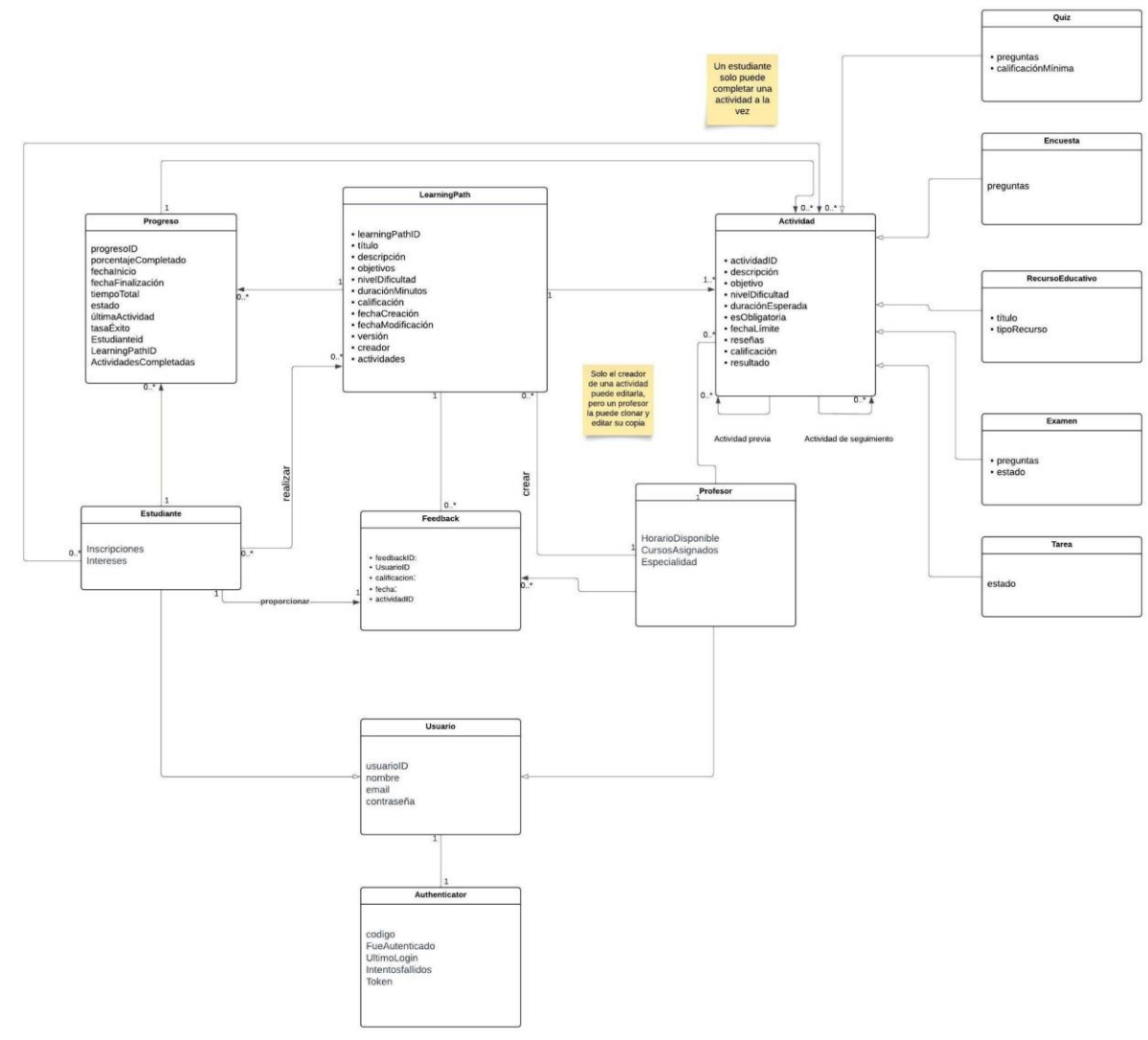


A continuación, para mirar la interfaz gráfica se ejecutará la clase *Interfaz* como java application, de forma que se pueda visualizar la interfaz.

Aclaraciones uso de interfaces:

1. La interfaz de estudiante solo sirve cuando se inicia sesión y se inscribe a de learning path, antes de eso no va a servir los otros botones.
 - a. Usuario: Victoria
 - b. Contraseña: zama
2. La interfaz de Profesor no tiene ningún requerimiento previo para poder usarla
 - a. Usuario: Santiago
 - b. Contraseña: cheto

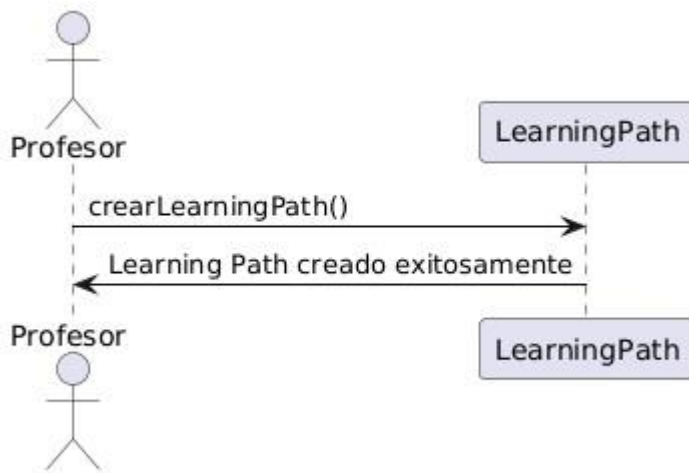
Diagrama de alto nivel:



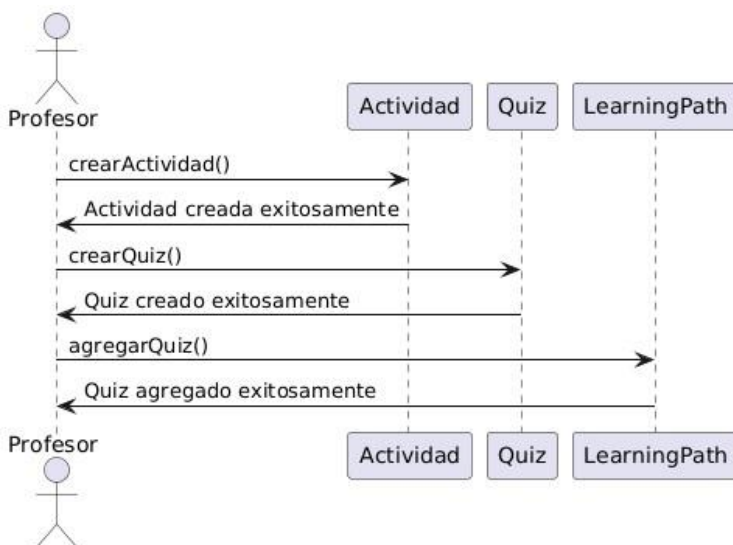
A partir de este modelo que representa las relaciones entre los diferentes modelos del problema, se pueden crear diagramas de secuencia que permiten ver exactamente qué métodos se deben implementar para poder realizar de manera correcta la interacción entre los diferentes elementos.

Diagramas de secuencia:

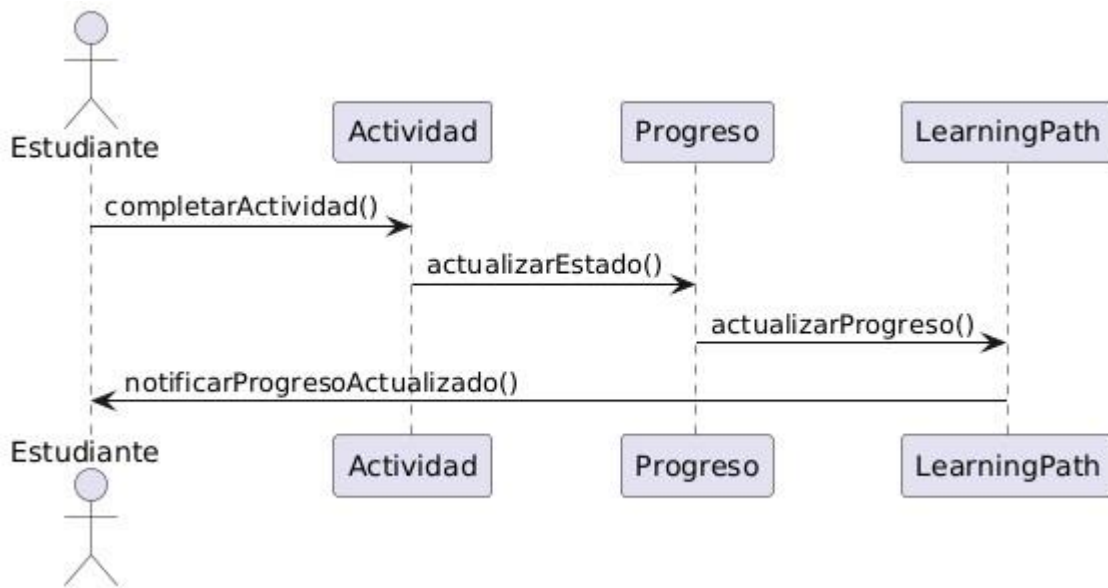
Este diagrama de secuencia muestra un proceso simple en el que un profesor crea un Learning Path. El profesor envía la solicitud para crear el Learning Path, y el sistema responde confirmando que el Learning Path ha sido creado exitosamente. Este flujo básico refleja la interacción inicial entre el profesor y el sistema para crear una Learning Path dentro de la plataforma.



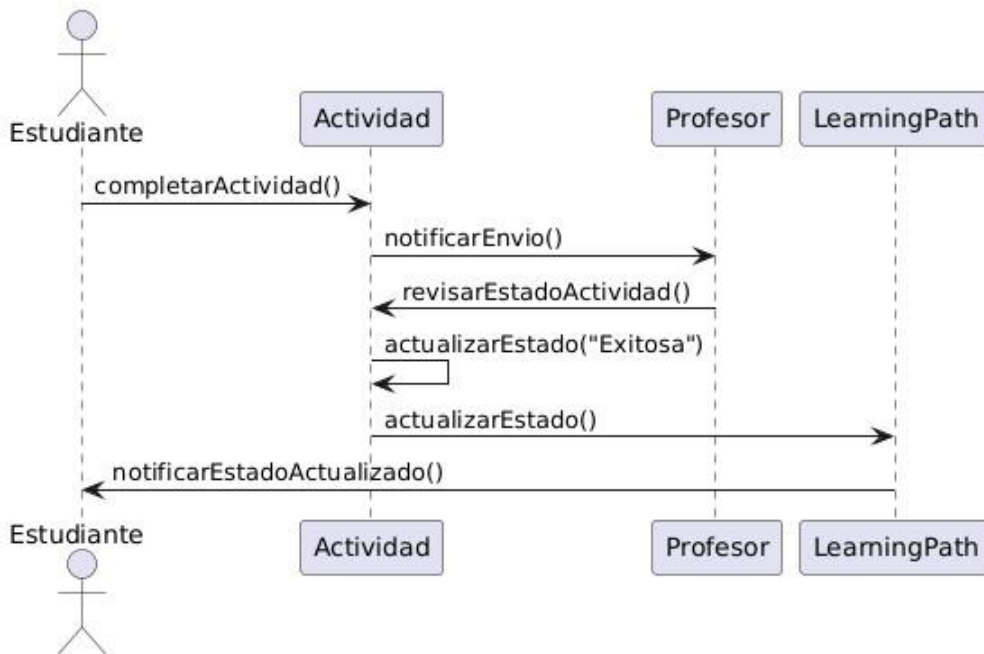
Este diagrama de secuencia muestra el proceso en el que un profesor crea una actividad, específicamente un cuestionario, y lo agrega a un Learning Path. Primero, el profesor crea una actividad, y el sistema confirma que ha sido creada exitosamente. Luego, el profesor crea un cuestionario, que también es confirmado por el sistema como creado correctamente. Finalmente, el profesor agrega el cuestionario al Learning Path, y el sistema confirma que el cuestionario ha sido agregado exitosamente.



Este diagrama de secuencia muestra cómo un estudiante completa una actividad y cómo se actualiza su progreso dentro del Learning Path. Primero, el estudiante completa la actividad, lo que desencadena una actualización del estado de la actividad. Luego, el progreso del estudiante en el Learning Path se actualiza en consecuencia. Una vez que el progreso ha sido actualizado, se notifica al estudiante que su progreso ha sido registrado.

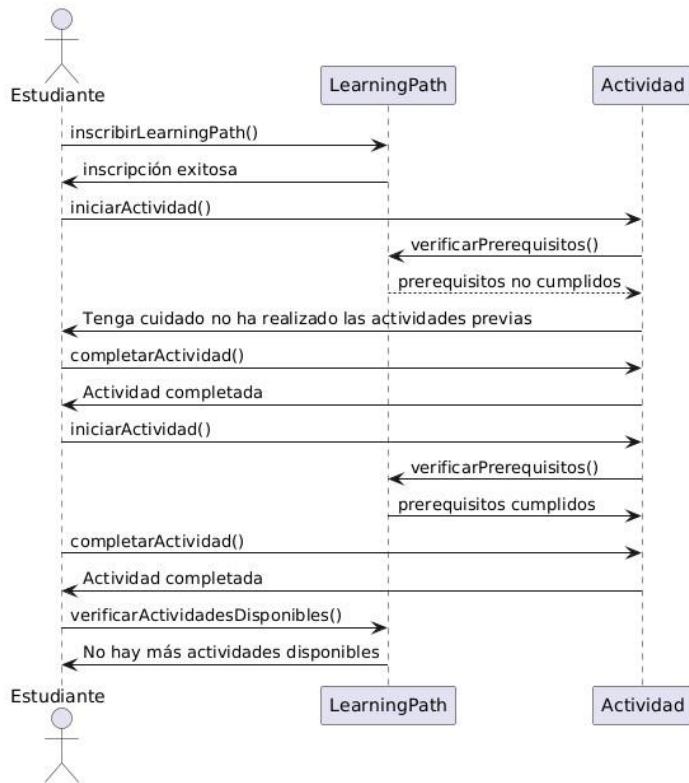


Este diagrama de secuencia muestra el proceso en el que un estudiante completa una actividad y el profesor revisa y actualiza el estado de dicha actividad. Primero, el estudiante completa la actividad, y el sistema notifica al profesor que se ha enviado. El profesor revisa el estado de la actividad y actualiza el estado como "Exitosa". Finalmente, el sistema notifica al estudiante que el estado de la actividad ha sido actualizado.

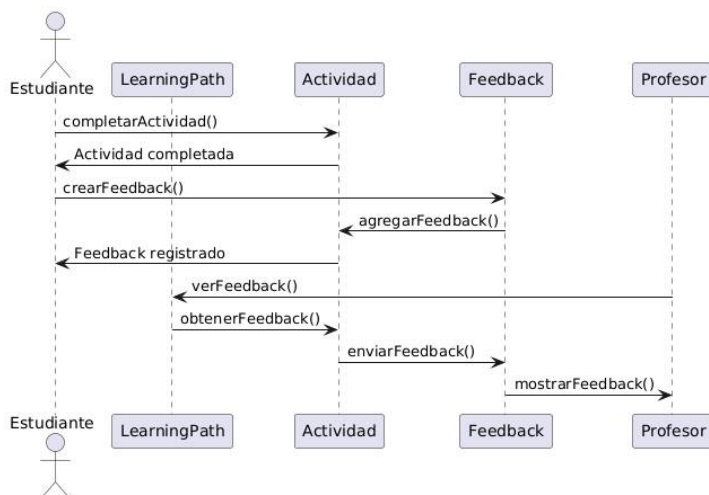


Este diagrama de secuencia muestra cómo un estudiante interactúa con un Learning Path y las actividades asociadas. El estudiante se inscribe exitosamente en el Learning Path. Luego, intenta iniciar una actividad, pero se le advierte que no ha cumplido con los requisitos previos. Después de completar una actividad, se le marca como completada y el estudiante puede iniciar otra actividad, cumpliendo esta vez con los requisitos previos. Al final, el sistema verifica si hay más actividades

disponibles y le informa al estudiante que no quedan más por realizar. El diagrama refleja el flujo de validación de requisitos previos y el registro del progreso del estudiante en su Learning Path.



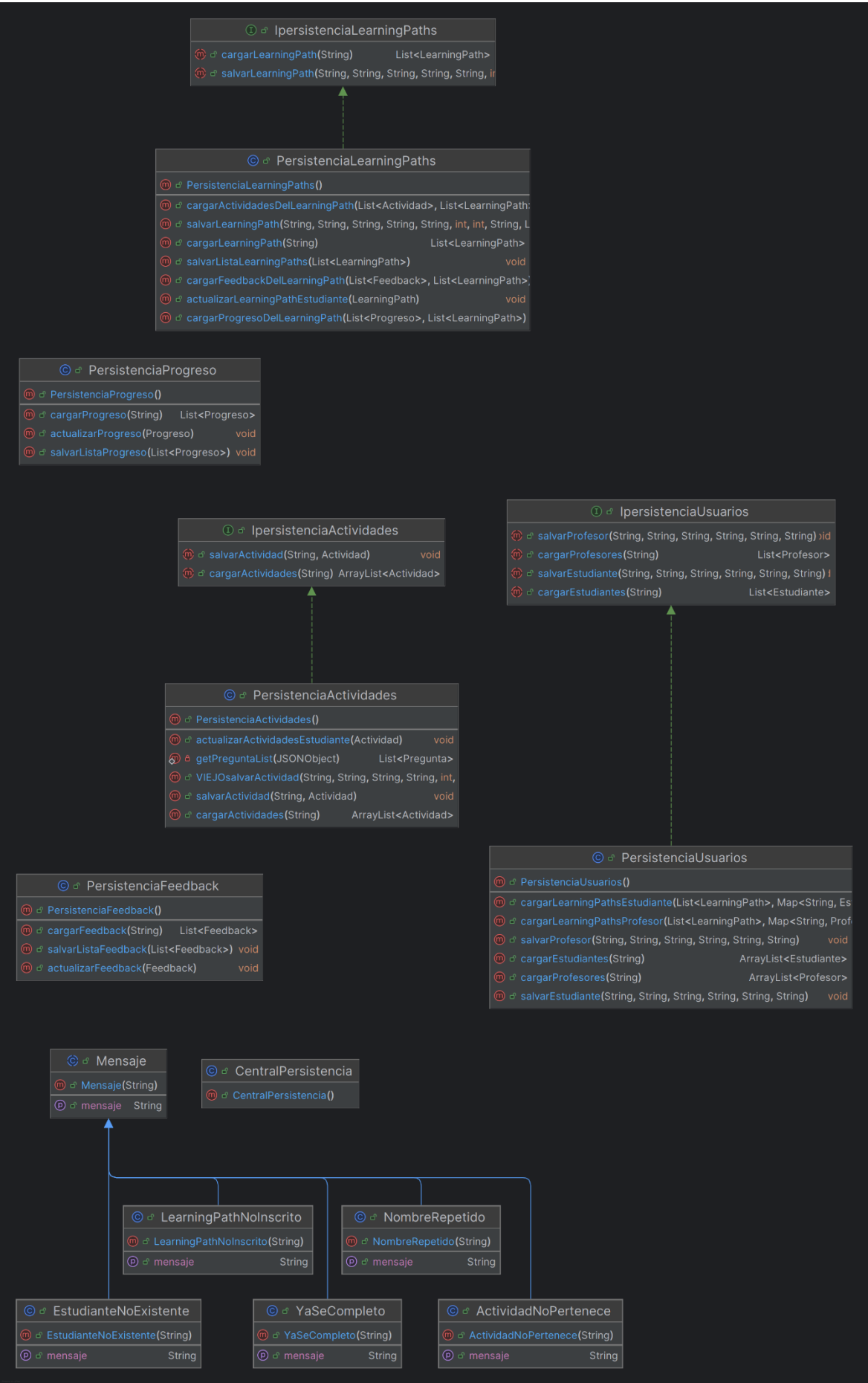
Este diagrama de secuencia muestra el proceso mediante el cual un estudiante completa una actividad y luego proporciona feedback sobre la misma. Primero, el estudiante completa la actividad, y el sistema confirma que ha sido completada. A continuación, el estudiante crea un feedback sobre la actividad, el cual es registrado en el sistema. El estudiante puede ver el feedback previamente registrado, y el sistema envía este feedback al profesor, quien lo revisa. Finalmente, el profesor puede visualizar el feedback enviado por el estudiante. Este proceso refleja cómo se gestiona el ciclo de retroalimentación entre el estudiante y el profesor dentro de un Learning Path.



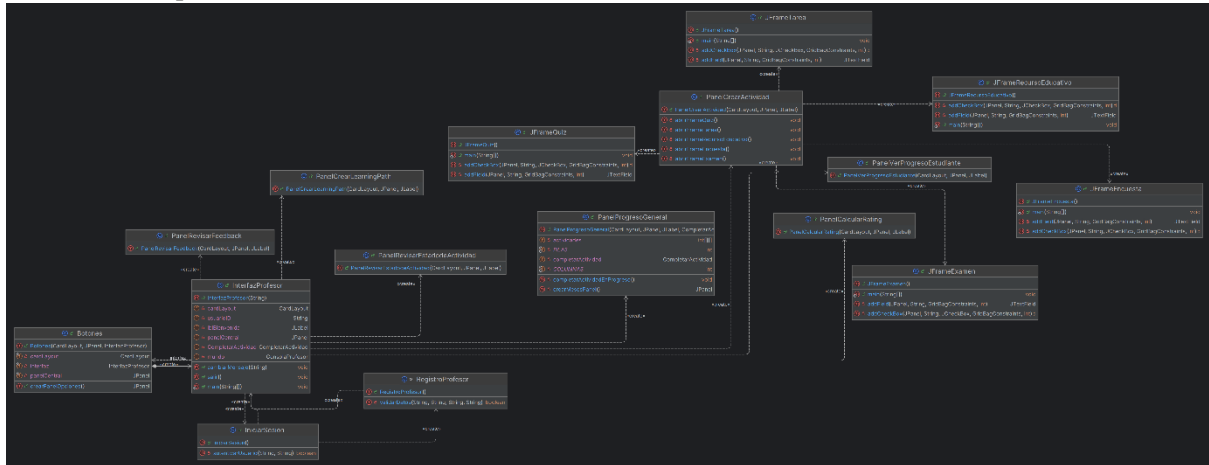
Finalizando, con los diagramas de secuencia se pudo construir con mayor facilidad el diagrama de clases UML final del problema.

UML Mundo



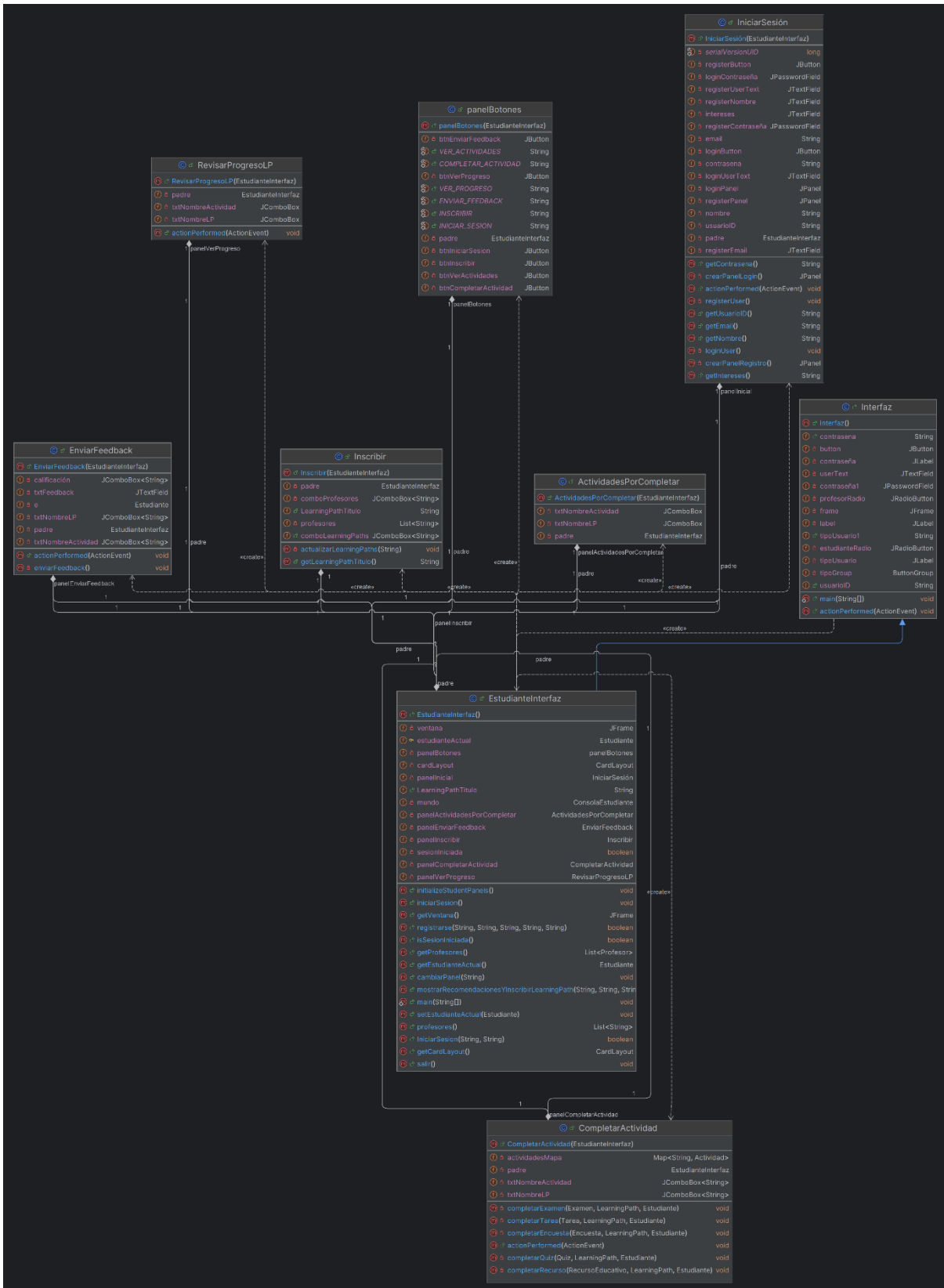


UML Interfaz profesor



•

UML Interfaz estudiante y principal:



Luego de ya implementado el código correspondiente a los diagramas utilizados anteriormente, se pueden realizar diferentes pruebas para confirmar la correcta ejecución del sistema.

Requisitos para revisar la correcta implementación del programa:

Finalmente, al haber ejecutado la implementación del código correspondiente a cada uno de los diagramas expuestos anteriormente, se van a realizar las siguientes pruebas para comprobar que se haya implementado de manera correcta:

1. Iniciar sesión como usuario: Se crea un nuevo profesor y un nuevo estudiante (para poder probar ambos) y se utiliza la función iniciar Sesión Usuario, donde el autenticador permite o no el inicio de sesión.
2. Profesor crea un learning Path: se llama a la función crearLearningPath(), implementada en la clase de Profesor, y se revisa que esta haya quedado correctamente creada buscando el Learning path ID en los learning paths del profesor.
3. El profesor crea una actividad y la añade al learning path: utilizando el learning path creado anteriormente, se utiliza la función crearActividad() implementada en la clase Profesor, para crear la nueva actividad. Luego se utiliza la función getActividades() implementada en learning paths para confirmar si el número de actividades en el learning path aumentó, y quedó correctamente inscrita la actividad.
4. El estudiante busca recomendaciones sobre a qué learning Path inscribirse: se llama a la función obtenerRecomendaciones() implementada en la clase Estudiante, ingresando el interés del estudiante. Esta función debería retornar todos los Learning Paths del profesor del estudiante que cumplan con su interés.
5. El estudiante se inscribe a un Learning Path: se llama a la función inscribirLearningPath() implementada en la clase estudiante que debería retornar un diccionario que contenga la descripción del Learning Path y de sus actividades.
6. El profesor revisa el estado de la actividad: se llama a la función getEstado() de la clase Actividad, antes y después de llamar a la función revisarEstadoActividad() implementada en la clase profesor, para verificar que luego de que el profesor revise el estado este se actualice.
7. El estudiante completa la actividad exitosamente: se llama a la función completarActividad() implementada en la clase estudiante, y se revisa que el estado de la actividad cambie a “Exitoso” para asegurar que esta fue completada exitosamente. Se asegura que el sistema imprima la advertencia de que el estudiante no ha completado las actividades previas y de que presente una recomendación de con cuál actividad debería continuar.
8. Revisar fecha límite de la actividad: Luego de completar la primera actividad se completa la actividad de seguimiento para revisar que al llamar la función getFechaLimite() implementada en actividad, de una respuesta coherente con el momento en que se terminó la actividad previa.

9. El estudiante revisa qué actividades debe completar del learning Path: llamando a la función actividadesDisponibles() implementada en la clase estudiante, se debería poder revisar las actividades que todavía no han sido completadas. Para esto se revisa la función antes y después de completar una actividad, asegurando que esta sea eliminada de la lista de actividades disponibles.
10. Revisar el porcentaje de éxito del estudiante: Se llama a la función getProgresoLearningPath() implementada en la clase estudiante y al ya haber completado todas las actividades subidas al learning Path, se espera que este retorne 100%.
11. El profesor revisa el feedback del learning path: se debe llamar a la función enviarFeedback() implementada por la clase estudiante, y luego se llama a la función revisarFeedback() implementada en la clase profesor. Al revisar el feedback debería aparecer el feedback añadido por el estudiante anteriormente.
12. Revisar rating del Learning Path: Dos estudiantes envían un feedback con un rating diferente sobre un mismo Learning Path, y se llama a la función calcularRating() implementada en la clase profesor. El resultado debería ser el promedio de los dos ratings enviados en el feedback del estudiante.
13. Revisar progreso del estudiante: Se llama a la función verProgresoEstudiante(), implementado en la clase profesor, que debería retornar un mapa con todos los atributos específicos del progreso del estudiante.

Si todos los requerimientos anteriores se cumplen se puede asumir que la implementación del sistema funciona, ya que en las funciones llamadas se implementan todas las funciones del programa.

Historias de usuario

Profesor:

- Como profesor quiero poder registrarme en el sistema, ingresando un correo electrónico, un usuario y una contraseña.
- Como profesor quiero poder iniciar sesión con mi usuario y contraseña para recibir un menú con las opciones de actividades que puedo realizar.
- Como profesor quiero poder crear Learning paths, indicando su nombre, objetivos, nivel de dificultad y duración, además de sus respectivas actividades, para que mis estudiantes puedan desarrollar distintas habilidades.
- Como profesor quiero crear una actividad, indicando su descripción, objetivos, duración, nivel de dificultad y preguntas (y respuestas si es de tipo quiz), para poder añadirla al learning path que me parezca relevante.

- Como profesor quiero llevar seguimiento del progreso de mis estudiantes, en los distintos Learning Paths, que luego pueda revisar a partir del ID del estudiante y ver cosas como el estado de las actividades, el porcentaje de éxito, el tiempo que ha tomado en completarse, entre otros.
- Como profesor quiero poder acceder al feedback de mis estudiantes a partir del ID del aprendizaje que deseo revisar, para mejorar el contenido de mis actividades a partir de las recomendaciones dadas por los estudiantes y revisar el rating que los estudiantes han dejado en este.
- Como profesor quiero poder actualizar el estado de una actividad, a partir del ID de la actividad, para poder modificarlo dependiendo de la respuesta enviada por el estudiante, luego de haberla revisado.

Estudiante:

- Como estudiante quiero poder registrarme en el sistema, ingresando un correo electrónico, un usuario y una contraseña.
- Como estudiante quiero poder iniciar sesión con mi usuario y contraseña para recibir un menú con las opciones de actividades que puedo realizar.
- Como estudiante quiero recibir recomendaciones sobre qué learning Path completar a partir de mis intereses, para poder escoger a qué Learning paths es más adecuado inscribirme.
- Como estudiante quiero poder inscribirme a un Learning path, a partir de su ID y el ID de mi profesor, donde pueda escoger las actividades que deseo completar y desarrollar las habilidades que deseo mejorar.
- Como estudiante quiero poder completar una actividad, a partir de su ID, respondiendo las preguntas de esta, y marcar como completadas las actividades una vez completadas.
- Como estudiante quiero tener un rastreo de mi progreso al llevar a cabo diferentes actividades.
- Como estudiante quiero recibir recomendaciones sobre las actividades que debería llevar a cabo, a partir del orden que el profesor considere más apropiado para mis habilidades.
- Como estudiante quiero poder revisar la fecha límite para completar una actividad, a partir del ID de la actividad.
- Como estudiante quiero poder revisar qué actividades todavía debo completar para finalizar un aprendizaje path, a partir del ID del aprendizaje path, para poder llevar registro sobre lo que debo trabajar.
- Como estudiante quiero poder agregar feedback al learning path, ingresando un comentario y un rating, para que mi profesor pueda revisarlo y mejorar el camino de aprendizaje construido.