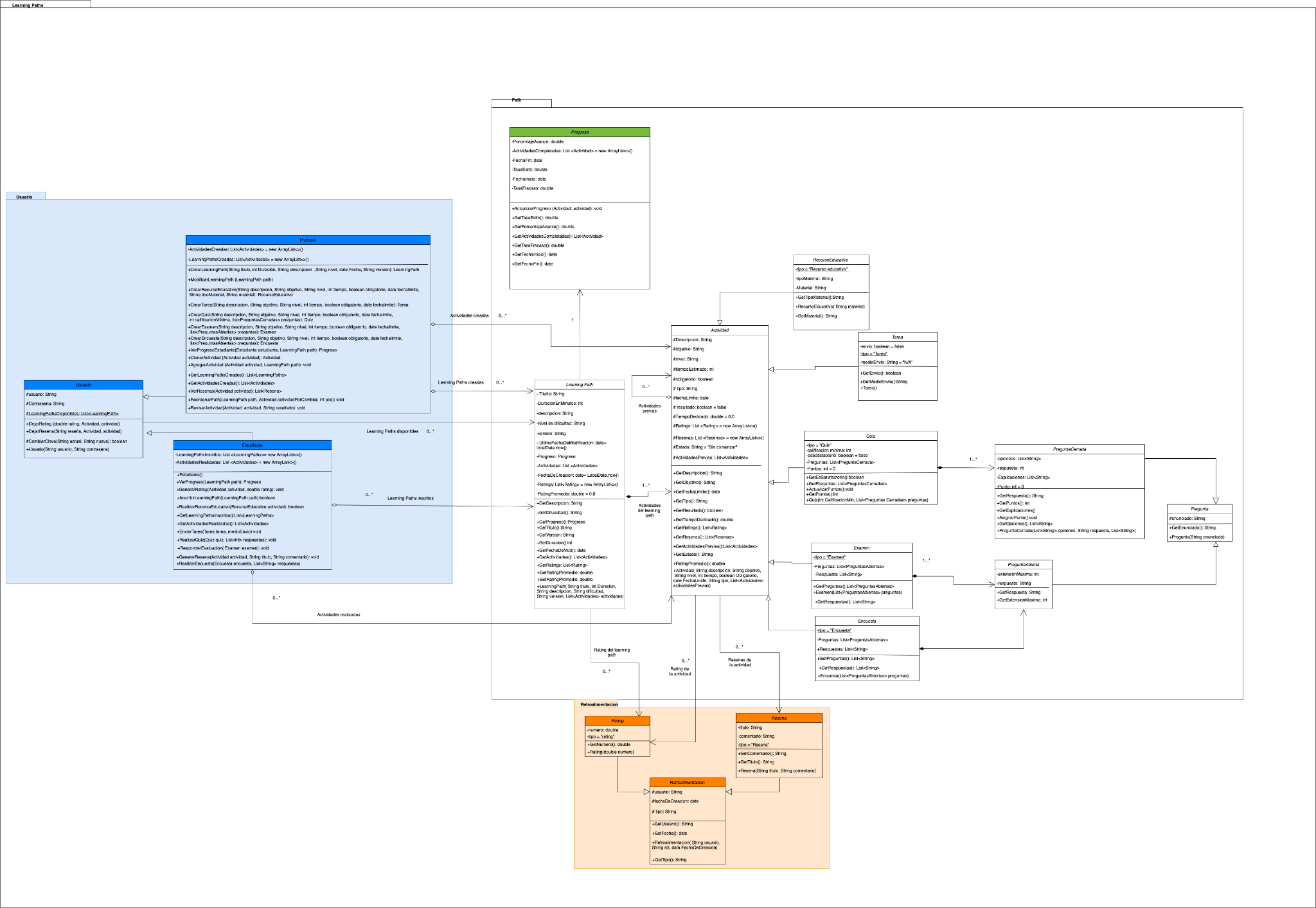
**Entrega 2 - Diseño e implementación del proyecto**

Lucas Valbuena Leon 202311538

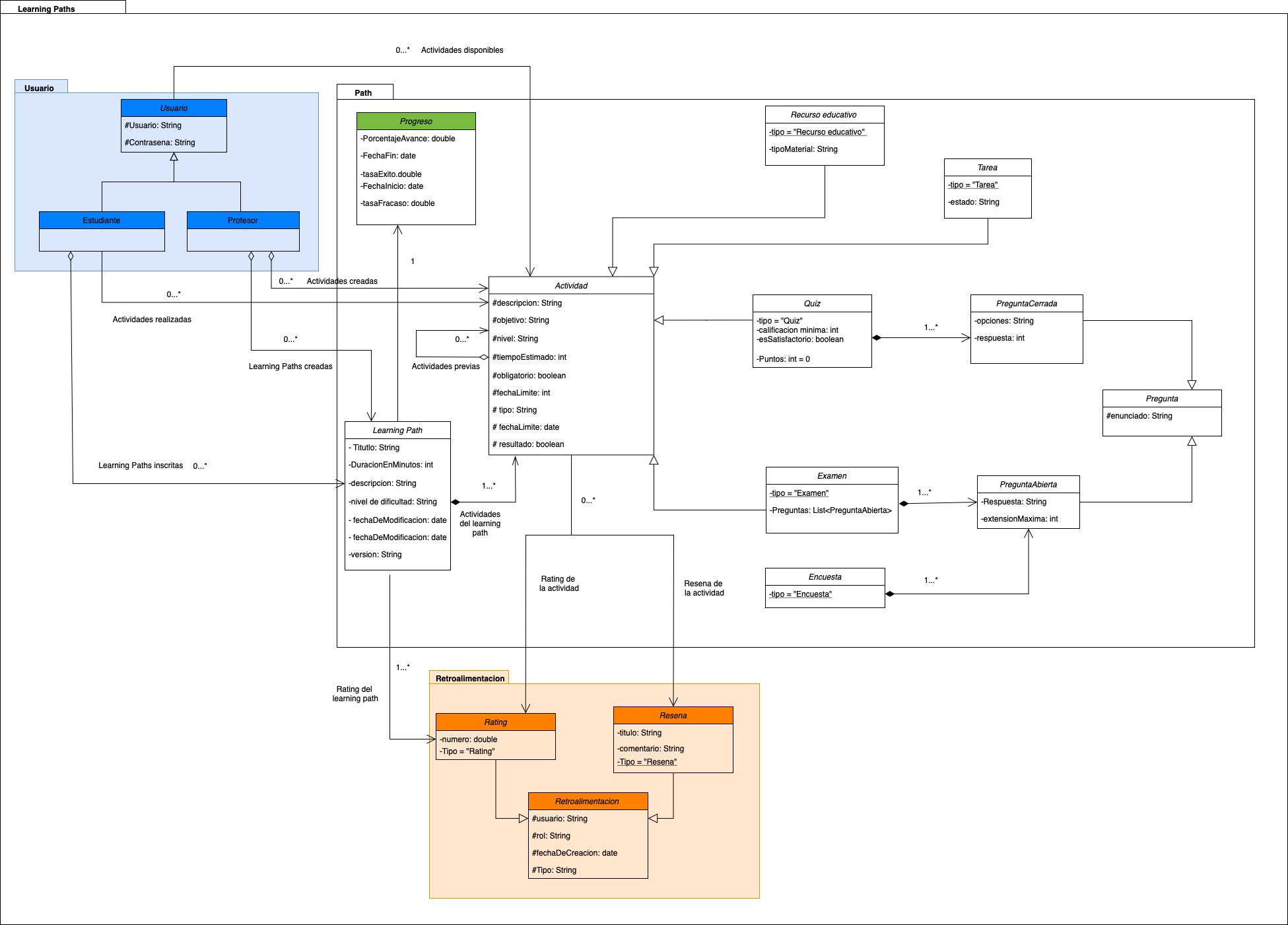
Juan Goyeneche Saenz 202320863

Alejandro Garcia Rojas 202122516

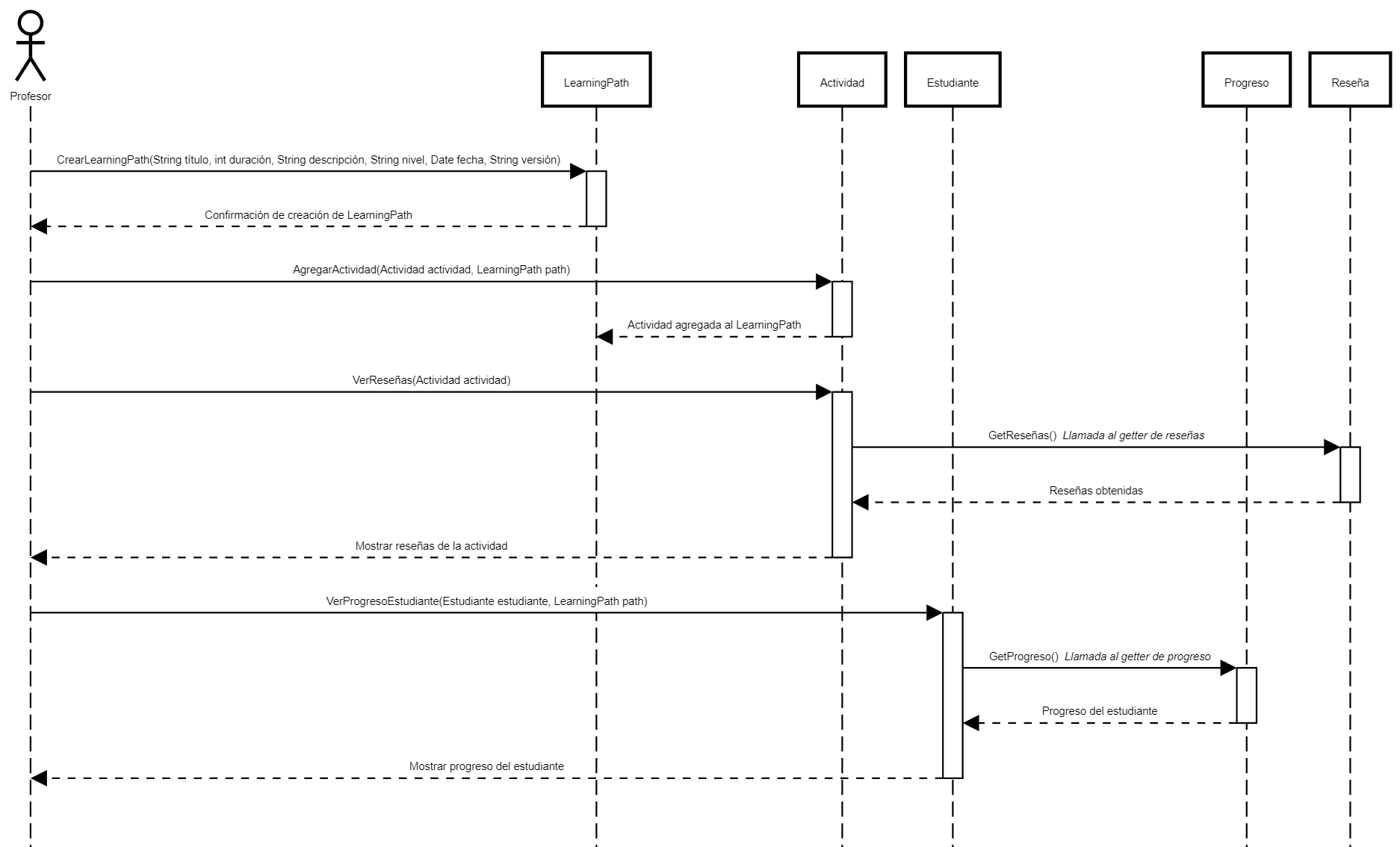
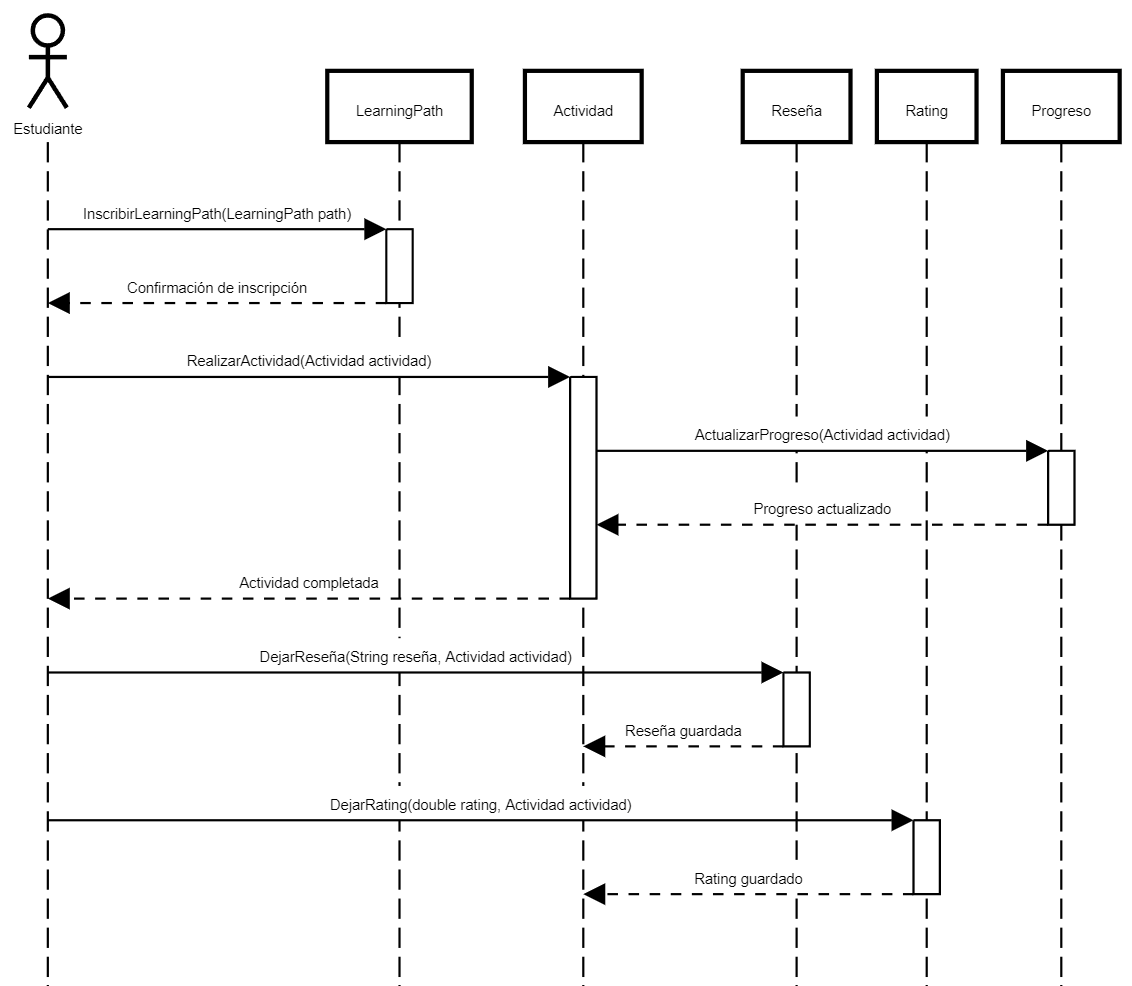
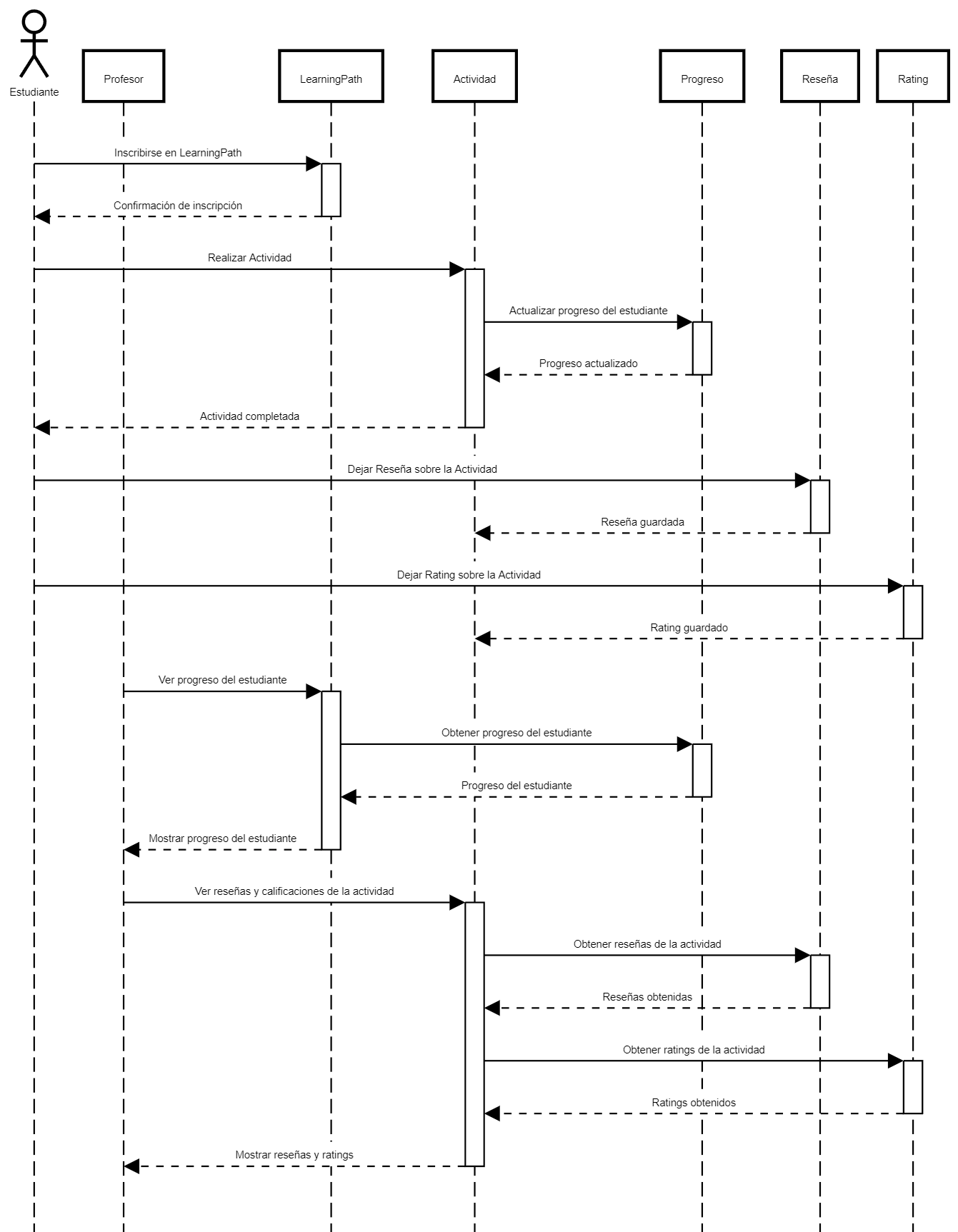
1. **Diagrama de clases de diseño**

****

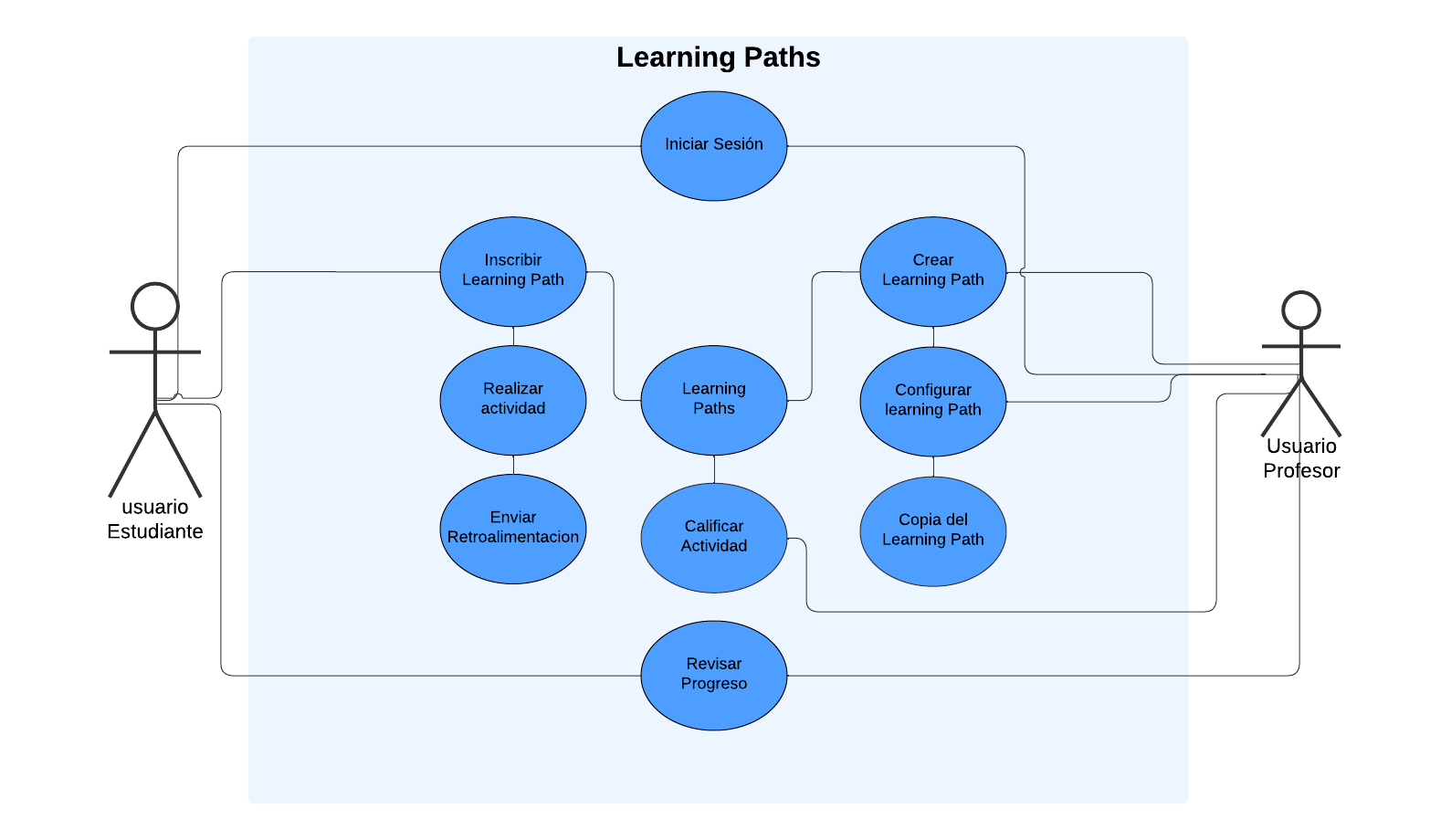
1. **Diagrama de clases de alto nivel**

****

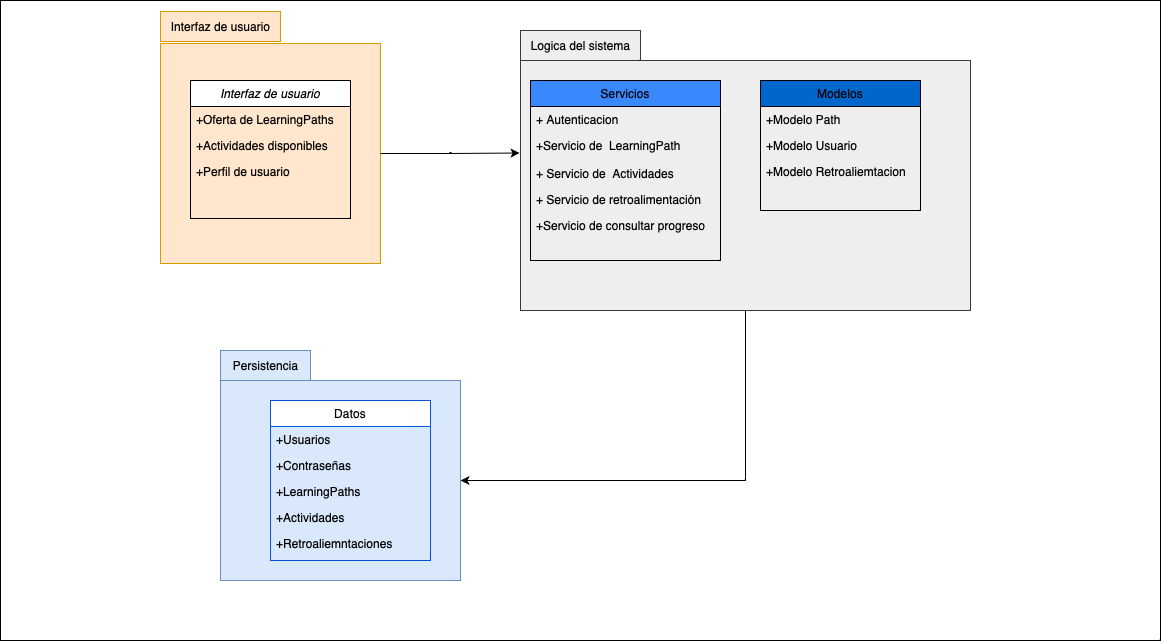
1. **Diagramas de secuencia**

****

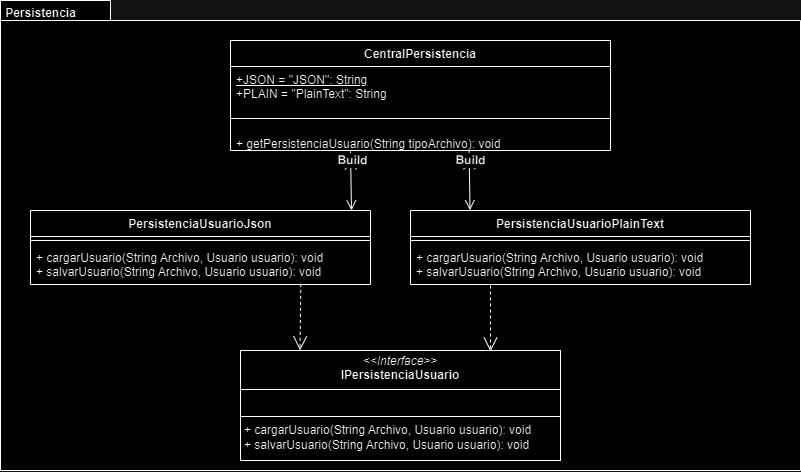
1. **Diagramas de casos de uso**

****

1. **Diagrama de paquetes**

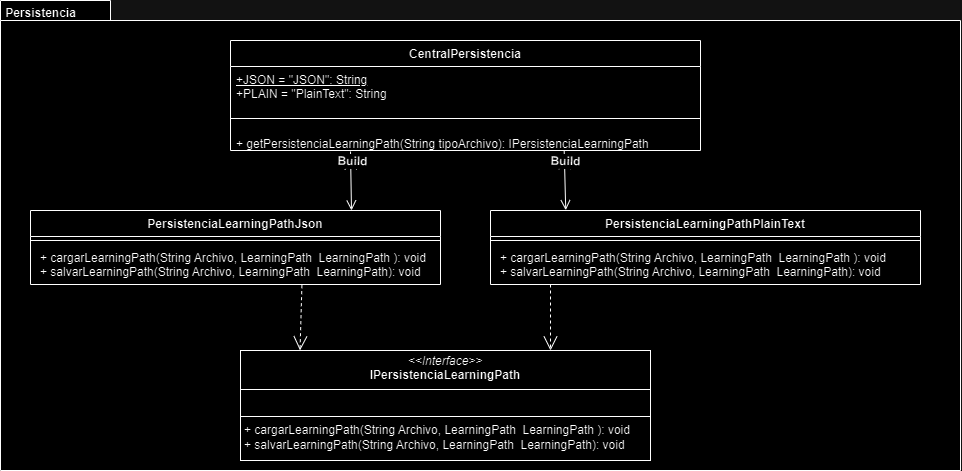
****

1. **Justificación:**
   1. **Usuario:** Para poder cumplir con los requerimientos asociados a profesor y estudiante, se tuvo en cuenta la clase usuario que permite cumplir en ambos casos de una forma adecuada con el registro, identificación y autenticación de los distintos tipos de usuarios. Además, posee un atributo público y estático donde se almacenan todas las learning paths disponibles que facilitará la implementación de los requerimientos funcionales, por lo que tiene una relación de contenencia de o más learning paths. Para poder cumplir con los requerimientos específicos las siguientes dos clases heredan de usuario:
      1. Profesor: Este tipo de usuario debe poder crear learning paths como actividades, por lo que como usuario profesor debe tener un registro de estos, teniendo así una relación de 0 a varias actividades o learning paths. De igual forma, al tener un registro de las actividades y learning paths creadas por el profesor se puede responder al requerimiento funcional de que los profesores pueden editar las que han creado. Sin embargo, los profesores pueden clonar actividades de otros profesores y editar sus propias copias.
      2. Estudiante: Por otra parte, este tipo de usuario se debe encontrar en la capacidad de interactuar con los Learning Paths y las actividades pero no editarlas. Por esto, el usuario estudiante tiene relación con 0 o varias actividades o learning paths. Asimismo, para cumplir con los requerimientos funcionales el estudiante debe poder ver los learning paths y decidir a cuál inscribirse. Del mismo modo, el estudiante puede ver las actividades y decidir cuál quiere hacer. Finalmente, el estudiante puede generar ratings o reseñas de las actividades que haya completado.
   2. **Learning Path:** En primer lugar, los learning paths tienen una relación de una a varias actividades, debido a que son por definición una secuencia de actividades que tiene como objetivo enseñar algo. En segundo lugar, los learning paths contiene 1 módulo de progreso que rastrea cómo los estudiantes avanzan a través de estos y permiten rastrear cuando un estudiante ha terminado un learning path (junto con otra información). En tercer lugar, los learning paths poseen una retroalimentación promedio en forma de rating que les permite a los estudiantes que buscan learning paths tener una forma cuantitativa de compararlos. Por último, la idea es que los estudiantes inscriban learning paths y los profesores los creen, por lo que los usuarios contienen 0 o varias de estos.
   3. **Actividad:** Los learning paths estaban compuestos por 1 o varias actividades en secuencia, pero las actividades también guardan relación con sí mismas en forma de actividades sugeridas previas que van de 0 a varias. Por otra parte, las actividades guardan relación con retroalimentación, donde pueden contener 0 o varias reseñas y 0 o varios rating. Las retroalimentación de las actividades son creadas por los estudiantes, pero solo pueden ser vistas por el usuario que la creó. Aunque los profesores pueden ver estas retroalimentaciones, con el fin de saber qué actividades podrían ser útiles o no. No obstante, actividad engloba lo que debe tener como mínimo una actividad, pero existen muchos tipos de actividades que se ajustan a las distintas necesidad educativas, por lo que heredan de actividad las siguientes clases:
      1. Recurso educativo: Este tipo de actividad incluye un archivo con material educativo adicional y se marca como completado una vez el estudiante lo realiza.
      2. Tarea: Esta clase se encarga de rastrear el proceso en el que un estudiante envía una tarea y es calificada por un profesor.
      3. Quiz: Esta actividad tiene contiene 1 o varias preguntas cerradas para existir. Pregunta cerrada hereda de pregunta, debido a que es un tipo específico de pregunta que contiene un enunciado pero tiene unas opciones de respuesta, junto con unas explicaciones y unos puntos asociados. El quiz se califica por sí mismo una vez terminado a través de una revisión de las preguntas por medio de una lista numérica que compara las opciones seleccionadas con las respuestas a cada pregunta y se aprueba cuando el puntaje supera la calificación mínima.
      4. Examen: igual que el tipo de actividad anterior, examen está compuesto de 1 o más preguntas abiertas, clase que hereda de pregunta pero tiene como atributo un número de caracteres permitidos para la respuesta. Al igual que tarea queda en pendiente por calificar hasta que un profesor lo revisa.
      5. Encuesta: Tiene también una relación de 1 a varias preguntas abiertas pero se completa una vez se responden las preguntas.
   4. **Progreso:** Más allá de ser un atributo decimal, esta clase permite rastrear aparte del porcentaje de avance las tasa de éxitos, tasa de fracaso, actividades obligatorias que fueron completadas y checa de inicio y fin por cada learning path.
   5. **Retroalimentación:** Para cumplir con el requerimiento funcional de poder dejar reseñas y ratings la clase retroalimentación es la generalización de estos, donde se rastrean datos importantes asociados a la creación del tipo de retroalimentación. De acá heredan las siguientes dos clases:
      1. Rating: es un número decimal de 0 a 5 que permite una evaluación cuantitativa de las actividades o el learning path en general. Los usuarios pueden crearlas y ver sus propios ratings generados, pero no los ratings individuales de otros usuarios.
      2. Reseña: Esta es la forma cualitativa de evaluar una actividad. Los usuarios solo pueden ver las reseñas creadas por ellos. Los profesores si están en la capacidad de ver las reseñas de todos los estudiantes por actividad.
2. **Diagrama de persistencia**

****

A través de la interfaz IPersistenciaUsuario, se definen los métodos cargarUsuario y salvarUsuario, los cuales son implementados por las clases concretas PersistenciaUsuarioJson y PersistenciaUsuarioPlainText, permitiendo la lectura y escritura de datos en formatos distintos (JSON y texto plano).

El propósito específico de esta persistencia es gestionar de manera eficiente el almacenamiento y recuperación de los datos del usuario (nombre de usuario y contraseña) en archivos, utilizando diferentes formatos según las necesidades del sistema o del contexto. Al implementar tanto la persistencia en JSON como en texto plano, se busca que la aplicación pueda manejar la información de los usuarios dependiendo del formato de archivo requerido.



A través de la interfaz IPersistenciaLearningPath, se definen los métodos cargarLearningPath y salvarLearningPath, los cuales son implementados por las clases concretas PersistenciaLearningPathJson y PersistenciaLearningPathPlainText, permitiendo la lectura y escritura de los Learning Paths en distintos formatos, como JSON y texto plano.

El propósito específico de esta persistencia es gestionar de manera eficiente el almacenamiento y recuperación de los datos de los Learning Paths (incluyendo atributos como título, descripción, actividades, y más) en archivos, utilizando el formato más adecuado según las necesidades del sistema o el contexto.

**Esta persistencia comparte la misma estructura centralizada con la de los usuarios, a través de la clase CentralPersistencia, lo que permite una fácil gestión de diferentes tipos de persistencia bajo una única interfaz.**