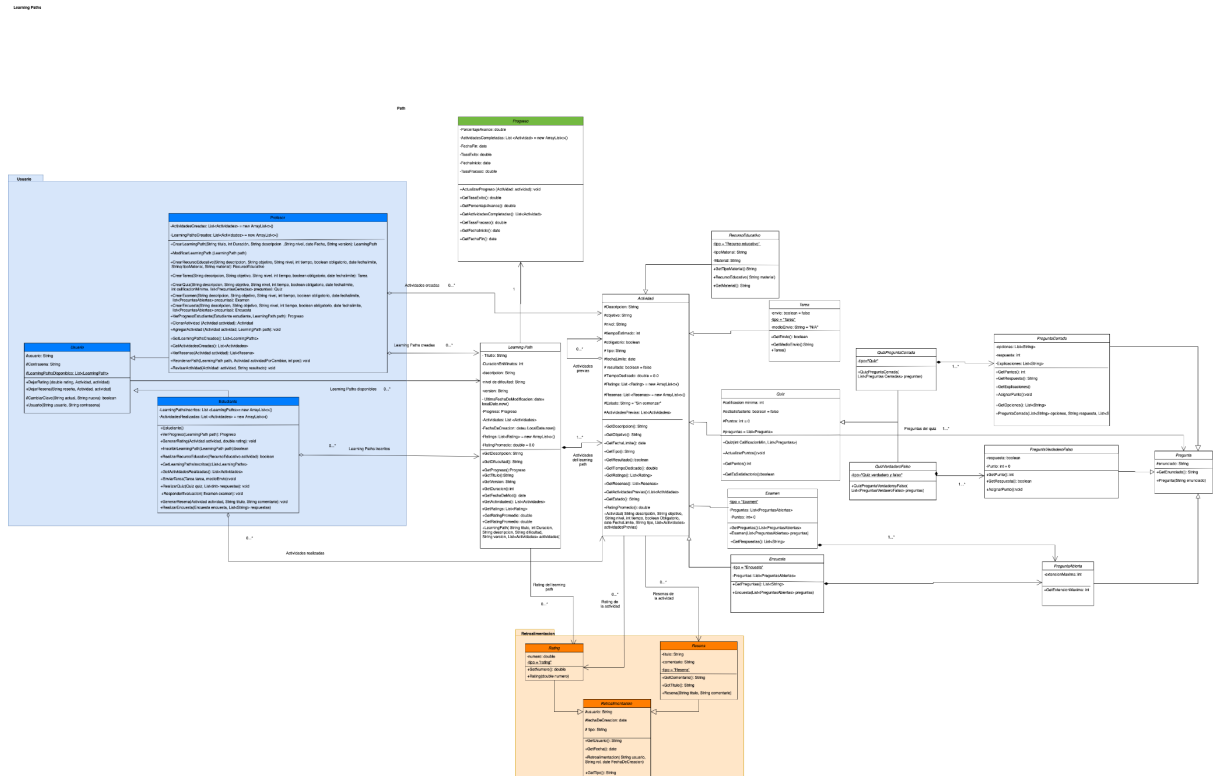


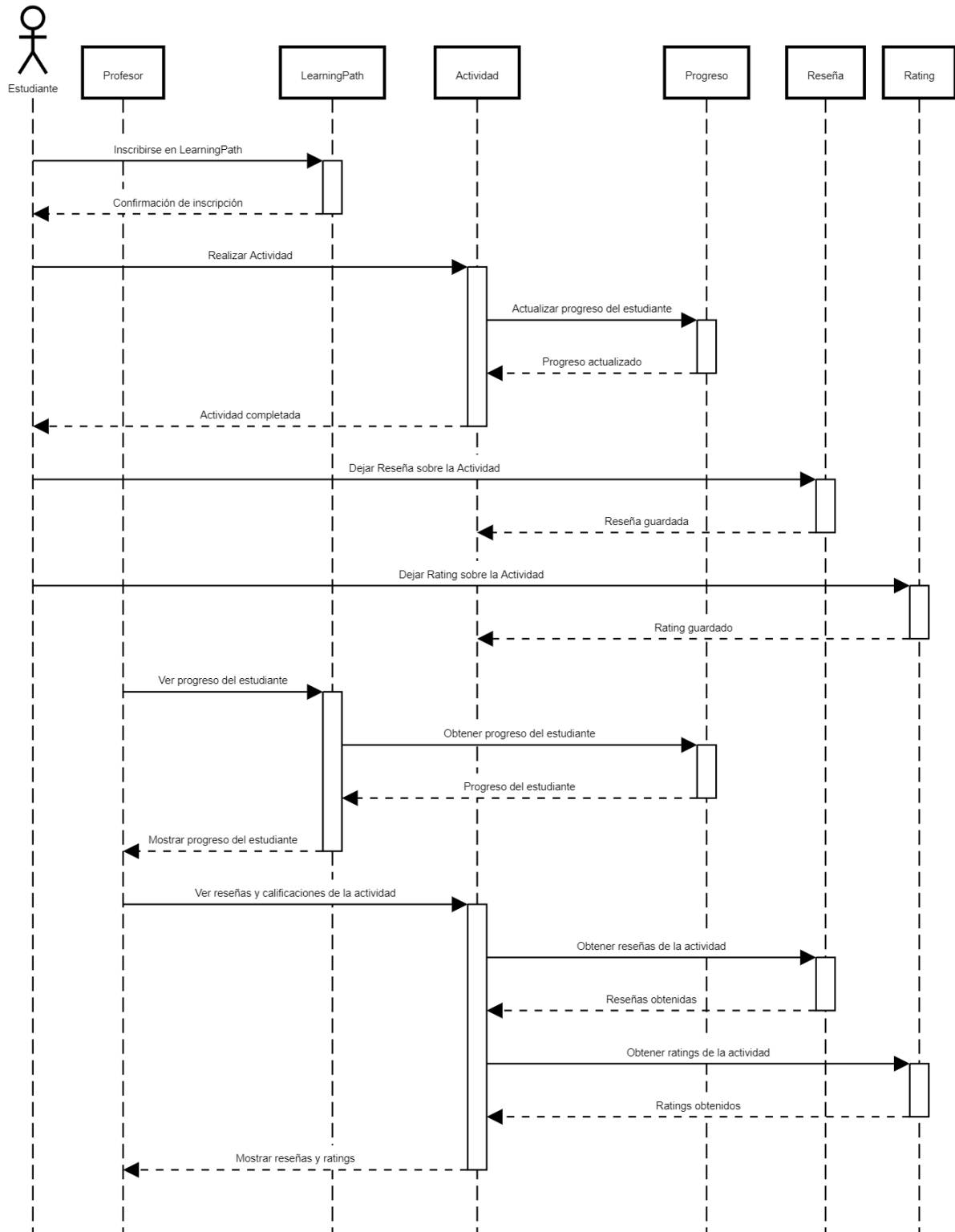
Lucas Valbuena Leon 202311538  
Juan Goyeneche Saenz 202320863  
Alejandro Garcia Rojas 202122516

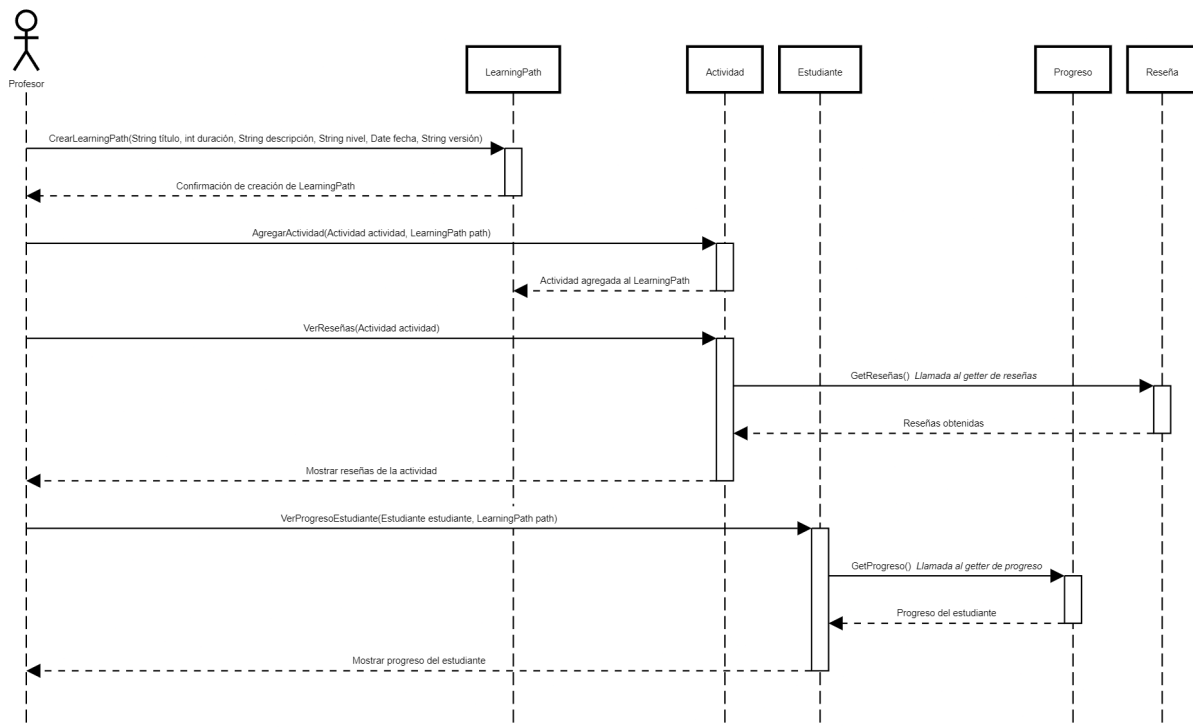
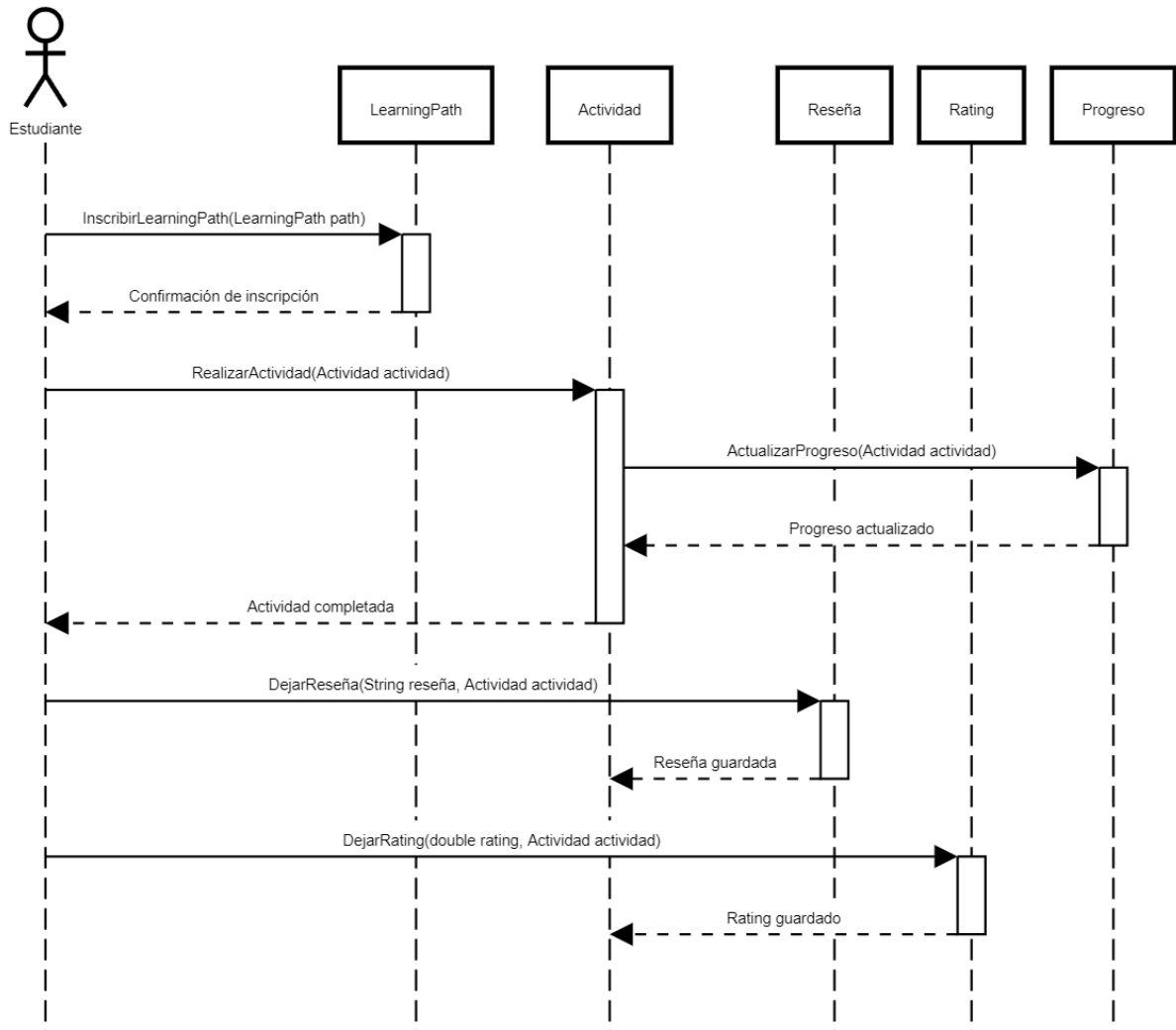
## 1. Diagrama de clases de diseño



## 2. Diagrama de clases de alto nivel







#### **4. Historias de usuario**

**Título:** Profesores de Seguimiento

### User story

**Como** Profesor de seguimiento,

**Quiero** hacerle seguimiento a los estudiantes que realizan un Learning Paths

**Para entonces**, poder ver y calificar el progreso de los estudiantes

### Acceptance criteria

- Calificar Learning Paths
- Revisar progreso de los estudiantes
- Ver rating o feedback de los estudiantes
- Dar feedback o rating de una actividad
- Administrar el correcto orden de Learning Paths para los estudiantes

**Título:** Estudiante

### User story

**Como** estudiante,

**Quiero** ver e inscribirme en Learning Paths

**Para entonces**, participar y realizar los Learning Paths y de esta manera realizar actividades correspondientes

### Acceptance criteria

- Inscribir Learning Paths
- Mostrar fechas, tiempos, éxito y cuando fueron completados los Learning Paths
- Elegir los Learning Paths que deseo empezar o continuar
- Dar feedback o rating de una actividad

**Título:** Profesor creador

### User story

**Como** profesor,

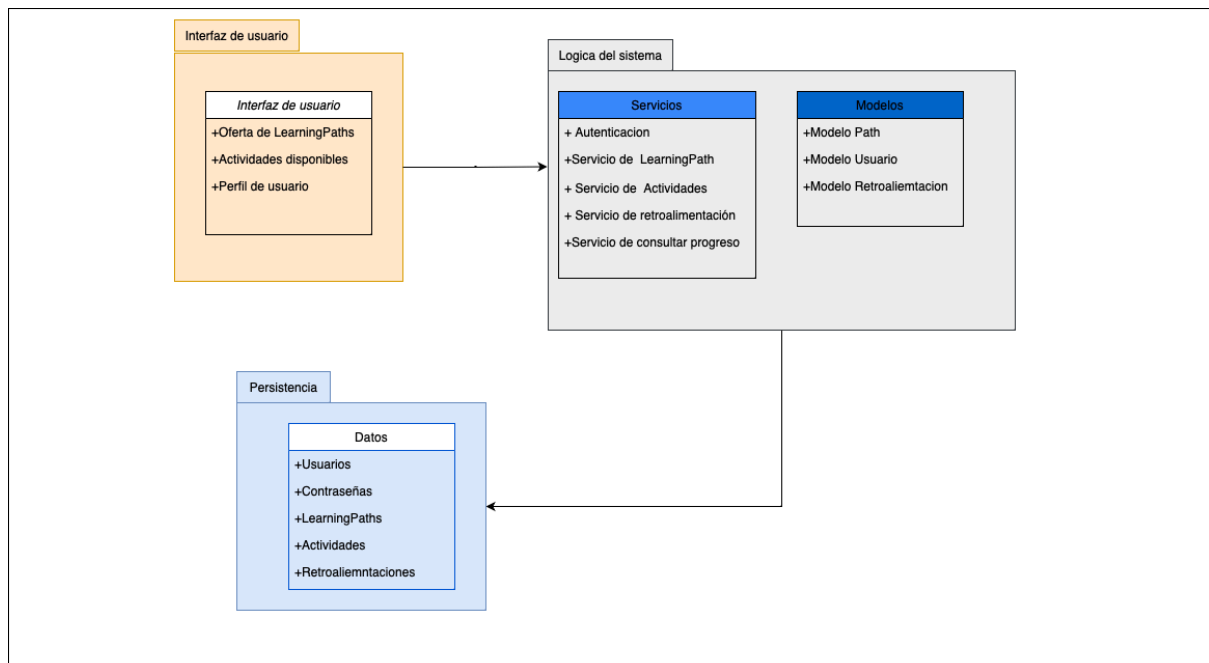
**Quiero** crear Learning Paths y actividades

**Para entonces**, generar Learning Paths para que los estudiantes los realicen y así aprendan.

### Acceptance criteria

- Crear Learning Paths
- Modificar Learning Paths
- Clonar actividades

## 5. Diagrama de paquetes



## 6. Justificación de diseño:

a. **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 contención de 0 a más learning paths. Para poder cumplir con los requerimientos específicos las siguientes dos clases heredan de usuario:

- i. 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.
- ii. 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.

- b. **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.
- c. **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 necesidades educativas, por lo que heredan de actividad las siguientes clases:

- i. Recurso educativo: Este tipo de actividad incluye un archivo con material educativo adicional y se marca como completado una vez el estudiante lo realiza.
- ii. Tarea: Esta clase se encarga de rastrear el proceso en el que un estudiante envía una tarea y es calificada por un profesor.
- iii. Quiz: Esta actividad tiene contiene 1 o varias preguntas para existir. El quiz puede tener 2 tipos, de pregunta verdadero y falso o de preguntas cerradas ambas clases que heredan de esta.

- 1. Quiz pregunta cerrada: para poder resolver el nuevo requerimiento se abstraigo quiz que contenía los métodos generales para la construcción de un quiz y su evaluación pero se hace esta herencia, debido a que es un tipo específico de preguntas que contiene un enunciado pero tiene unas opciones de respuesta, junto con unas explicaciones y una respuesta asociada. 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

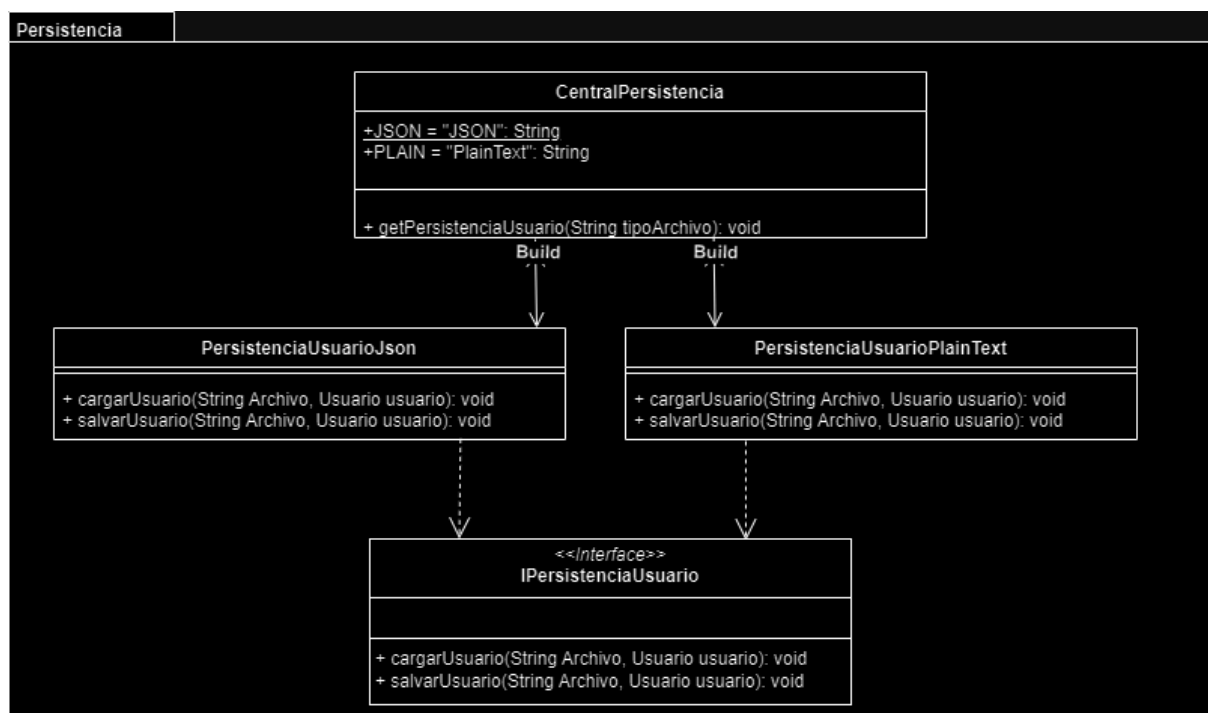
cada pregunta y se aprueba cuando el puntaje supera la calificación mínima.

2. Quiz pregunta verdadero o falso: Para solucionar el nuevo requerimiento, se creó pregunta verdadero o falso, la cual hereda de pregunta. Esta nueva clase tiene un atributo adicional de respuesta, pero esta solo puede ser booleanos. Al igual que la clase quiz pregunta cerrada, debido a que es un tipo específico de quiz que solo contiene preguntas verdadero o falso, posee los atributos de quiz y se califica igual que en el caso anterior, por medio de listas donde se asocia 1 a 1 cada respuesta.
- iv. 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 pendiente por calificar hasta que un profesor lo revisa.
- v. Encuesta: Tiene también una relación de 1 a varias preguntas abiertas pero se completa una vez se responden las preguntas.
- d. **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 check de inicio y fin por cada learning path.
- e. **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:

- i. 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.
- ii. 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.

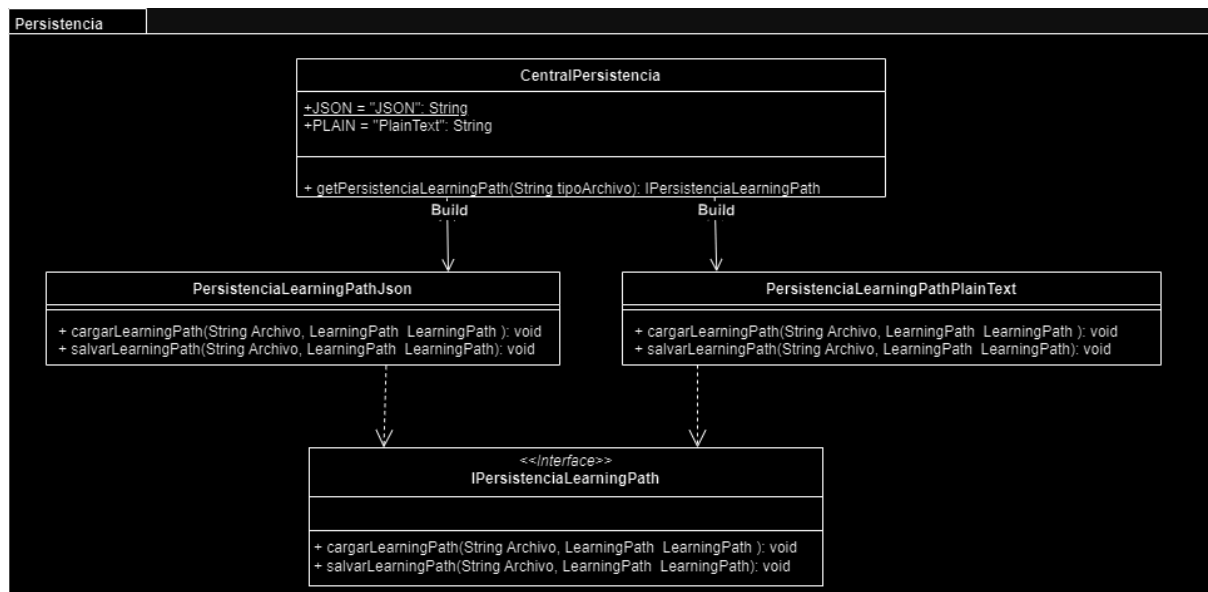
## 7. Diagrama de persistencia



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.

La persistencia planteada previamente mencionada es para métodos más generales. Para poder alimentar la consola se generó "persistencia concreta" que facilita la autenticación de los usuarios, para cumplir con este requerimiento de forma más específica y eficiente.



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.**

## 7. Consola

Para el diseño de la consola, se abordó desde una abstracción de consola básica que permite después especializar la consola en profesor y estudiante. Las consolas tienen la capacidad de autenticar un usuario, usando la persistencia del JSON para validar la información suministrada por el usuario, confirmando que el usuario y contraseña sean correctos. Del mismo modo, el sistema tiene la capacidad de agregar un usuario en caso de que no exista almacenando la información en un JSON. El sistema permite realizar las operaciones principales después de desplegar un menú, como lo es crear un learning path y mostrar toda la información relevante de los mismos.