# **Proyecto 2: Diseño e implementación** *grupo 9*

### Índice

1. Historias de usuario
2. Diagrama de clases de diseño
3. Diagrama de alto nivel
4. Diagramas de secuencia
5. Diagrama de clases interfaz.
6. Justificación de los métodos por clases
7. Documentación de la persistencia

## Historias de usuario

**Rol 1: Profesor**

1. En este rol quiero ser capaz de crear Learning Paths, en los cuales pueda definir el título, una descripción, sus objetivos y el nivel de dificultad.

Entradas: Título, descripción, objetivos y nivel de dificultad.

Salidas: Confirmación de creación y visualización del Learning Path.

1. En este rol quiero añadir actividades al Learning Path, estableciendo prerrequisitos para que los estudiantes tengan una orientación en su proceso de aprendizaje.

Entradas: Elección del tipo de actividad (Encuesta, Examen, Revisar Recurso, Tarea o Quiz), detalles de cada actividad específicos de cada una, actividades previas.

Salidas: Confirmación de la actividad añadida y visualización de la secuencia del Learning Path.

1. En este rol quiero revisar las reseñas y rating de las actividades para así poder hacer mejoras en cada una y mejorar la experiencia de aprendizaje de los estudiantes.

Entradas: Elección de la actividad a revisar.

Salidas: Visualización de las reseñas y ratings.

**Rol 2: Estudiante**

1. En este rol quiero ser capaz de registrarme en el sistema y poder acceder a los diferentes Learning Paths disponibles y tener un aprendizaje personalizado a mis preferencias.

Entradas: Información de registro (id, nombre, email, contraseña e intereses).

Salidas: Confirmación de creación de la cuenta y acceso a las funcionalidades de los estudiantes.

1. En este rol quiero poder inscribirme a diferentes Learning Paths de mi interés, para así aumentar mis conocimientos en áreas que me gustan.

Entradas: Elección de un Learning Path que el recomendador crea que es óptimo para mí.

Salidas: Confirmación de inscripción y visualización de la estructura que tiene el Learning Path.

1. En este rol quiero poder visualizar mi progreso en el Learning Path, donde este la información de las actividades realizadas, porcentaje de avance y tiempo invertido para ser consciente de todo lo que he hecho hasta ahora y mantener mi motivación.

Entradas: No hay.

Salidas: Un porcentaje generado, actividades por completar y tiempo invertido.

**Rol 3: Profesor (Seguimiento)**

1. En este rol quiero ser capaz de revisar las actividades que han sido enviadas por los estudiantes, marcarlas como exitosas o no.

Entradas: Seleccionar la actividad entregada, resultado de la evaluación

Salidas: Confirmación de la evaluación y actualización de progreso del estudiante.

1. En este rol quiero poder acceder a la información del progreso de cada estudiante en el Learning Path.

Entradas: Seleccionar al estudiante.

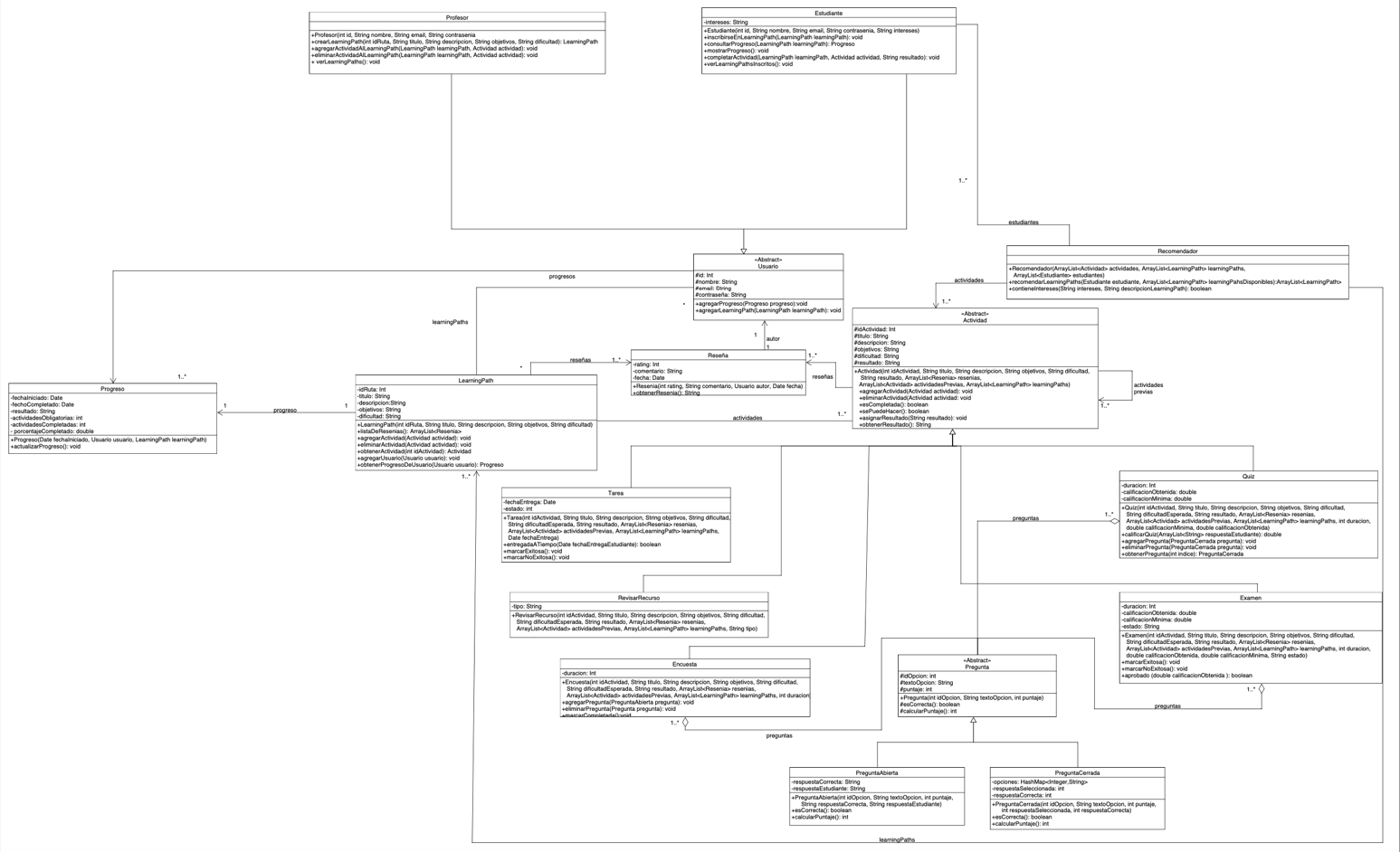
Salidas: Visualización del progreso.

1. En este rol quiero poder dejar una reseña y rating a una actividad

Entradas: Rating y comentarios de la actividad

Salidas: Confirmación de la reseña agregada.

## Diagrama de clases de diseño

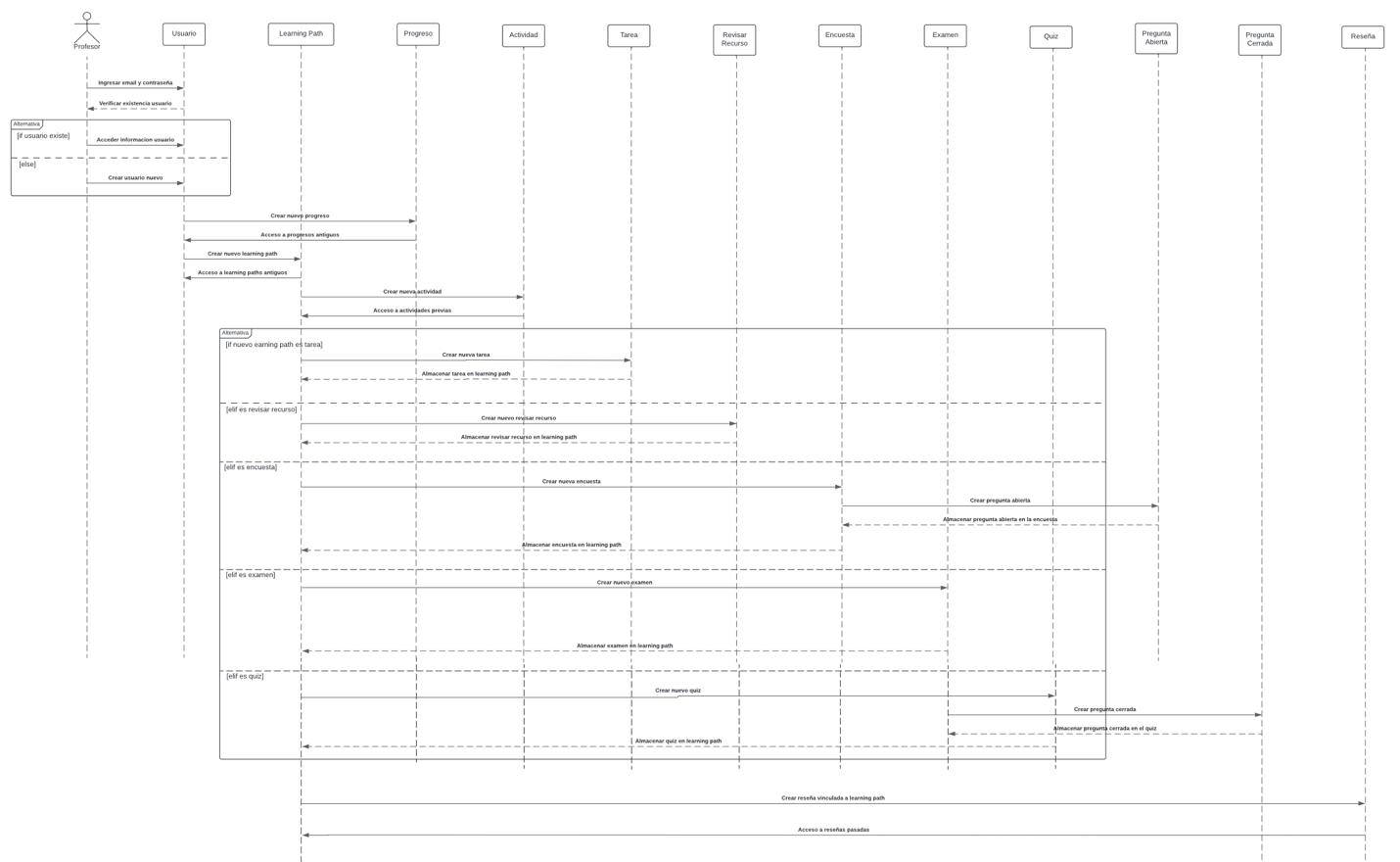


## Diagrama de clases de alto nivel

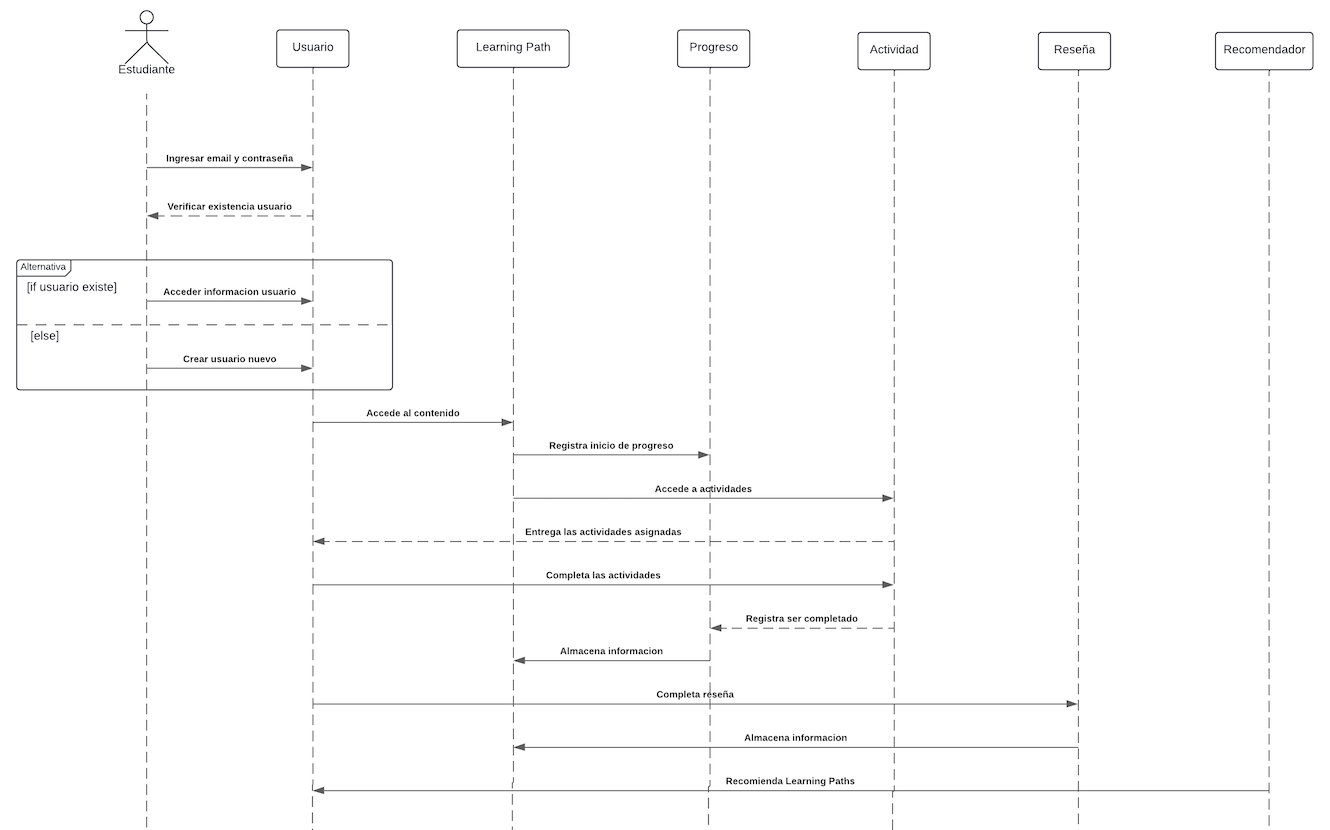
## 

## Diagramas de secuencia

#### Profesor



#### Estudiante



## Diagrama de Clases de la interfaz

A diagram of a computer

Description automatically generated with medium confidence

## Justificación de los métodos por clases

#### Usuario

* agregarProgreso(…): Genera una nueva instancia para el registro de un progreso, registrando los datos necesarios de la clase Progreso.
* agregarLearningPath(…): Genera una nueva instancia para el registro de un Learning Path, registrando los datos necesarios de la clase LearningPath.

#### Profesor

* Profesor(…): Registra un nuevo usuario profesor.
* crearLearningPath(…): Genera un nuevo Learning Path con la información requerida, registrada por el usuario profesor.
* agregarActividadAlLearningPath(…): Agrega la instancia de actividad nueva creada al Learning Path relacionado al profesor.
* eliminarlActividadAlLearningPath(…): Elimina la instancia de actividad referenciada en el LearningPath relacionado al profesor.
* verLearningPaths(…): Muestra los Learning Paths vinculados al usuario professor.

#### Estudiante

* Estudiante(…): Registra un nuevo usuario estudiante.
* inscribirseEnLearningPath(…): Inscribe al usuario estudiante al Learning Path de elección.
* consultarProgreso(…): Muestra el progreso en el Learning Path relacionado al usuario estudiante.
* mostrarProgreso(…): Muestra el progreso en el Learning Path relacionado al usuario estudiante.
* completarActividad(…): Genera acceso a la actividad que el usuario estudiante desea completar dentro de un Learning Path definido.
* verLearningPathsInscritos(…): Muestra todos los Learning Paths relacionados con el usuario estudiante.

#### Recomendador

* Recomendador(…): Genera una nueva instancia de un estudiante recomendador.
* recomendarLearningPaths(…): Muestra las recomendaciones de Learning Paths sugeridas al estudiante de acuerdo a los progresos e intereses.
* contieneIntereses(…): Booleano que indica si el Learning Path contiene los intereses del estudiante.

#### Progreso

* Progreso(…): Genera una nueva instancia de progreso.
* actualizarProgreso(…): Actualiza la información del progreso.

#### Learning Path

* LearningPath(…): Crea un nuevo Learning Path con toda la información necesaria ingresada por el usuario.
* ListaDeResenias(…): Muestra todas las reseñas existentes.
* agregarActividad(…): Genera una nueva instancia para una actividad.
* eliminarActividad(…): Elimina la actividad seleccionada.
* obtenerActividad(…): Obtiene la información relevante a la actividad en búsqueda.
* agregarUsuario(…): Crea un nuevo vinculo de acceso de un usuario a este Learning Path.
* obtenerProgresoDeUsuario(…): De acuerdo a la búsqueda ingresada, se obtiene el progreso relacionado a ese usuario.

#### Reseña

* Resenia(…): Crea una nueva reseña con la información otorgada por el usuario.
* obtenerResenia(…): Obtiene la información relevante a la reseña en búsqueda.

#### Actividad

* Actividad(…): Crea una nueva actividad con toda la información necesaria ingresada por el usuario.
* agregarActividad(…): Genera una nueva instancia para una actividad.
* eliminarActividad(…): Elimina la actividad seleccionada.
* esCompletada(): Booleano que indica si la pregunta fue completada.
* sePuedeHacer(): Booleano que indica si la pregunta está disponible para que el usuario estudiante pueda completar.
* asignarResultado(…): Asigna un resultado en escrito por el profesor de lo actividad.
* obtenerResultado(…): Se refleja el resultado previo.

#### Tarea

* Tarea(…): Crea una nueva tarea con toda la información necesaria ingresada por el usuario.
* entregadaATiempo(…): Booleano que indica si fue entregada la tarea a tiempo o no.
* marcarExitosa(): Booleano que indica si la pregunta fue aprobada.
* marcarNoExitosa(): Booleano que indica que la pregunta no fue aprobada.

#### Revisar Recurso

* revisarRecursos(…)

#### Encuesta

* Encuesta(): Crea una nueva encuesta con toda la información necesaria ingresada por el usuario.
* agregarPregunta(): Genera una instancia para adicionar una nueva pregunta.
* eliminarPregunta():Genera una instancia para eliminar una pregunta existente.
* marcarCompletada(): Booleano que indica que la encuesta fue completada.

#### Quiz

* Quiz(…): Crea un nuevo quiz con toda la información necesaria ingresada por el usuario.
* calificarQuiz(): Calcula el puntaje y retorna el entero correspondiente.
* agregarPregunta(): Genera una instancia para adicionar una nueva pregunta.
* eliminarPregunta(): Elimina la pregunta referenciada.
* obtenerPregunta(): Muestra la pregunta referenciada.

#### Examen

* Examen(…): Crea un nuevo examen con toda la información necesaria ingresada por el usuario.
* marcarExitosa(): Booleano que indica si la pregunta fue aprobada.
* marcarNoExitosa(): Booleano que indica que la pregunta no fue aprobada.
* aprobado(): Booleano que indica si el examen fue aprobado o no.

#### Pregunta

* Pregunta(…): Crea una nueva pregunta con la información ingresada por el usuario.
* esCorrecta(): Booleano que indica si la respuesta es correcta o no.
* calcularPuntaje(...): Calcula el puntaje y retorna el entero correspondiente.

#### Pregunta Abierta

* PreguntaAbierta(…): Crea una nueva pregunta con la información ingresada por el usuario.
* esCorrecta(): Booleano que indica si la respuesta es correcta o no.
* calcularPuntaje(): Calcula el puntaje y retorna el entero correspondiente.

#### Pregunta Cerrada

* PreguntaCerrada(…): Crea una nueva pregunta con la información ingresada por el usuario.
* esCorrecta(): Booleano que indica si la respuesta es correcta o no.
* calcularPuntaje(): Calcula el puntaje y retorna el entero correspondiente.

#### Pregunta Verdadero Falso

* PreguntaVerdaderoFalso (…): Crea una nueva pregunta con la información ingresada por el usuario.
* esCorrecta(): Booleano que indica si la respuesta es correcta o no.
* calcularPuntaje(): Calcula el puntaje y retorna el entero correspondiente.

## Documentación de la persistencia

Los datos que se guardan a largo plazo incluyen: el progreso de cada estudiante, los avances que han hecho en cada learningPath, información básica de cada usuario, las reseñas que dejan, y el contenido creado para cada learningPath. Esto asegura que el sistema pueda personalizar la experiencia, llevar un seguimiento detallado y mejorar con base en la retroalimentación de los usuarios.

Es fundamental el almacenamiento de estos datos para su reutilización dentro del ciclo de uso del programa, así como la recopilación de estos para formar una base de datos completa de reseñas, learningPaths, y usuarios con sus debidos atributos de perfil como historia y progreso.

Además de estos, datos como las respuestas ingresadas por el usuario estudiante a preguntas abiertas o cerradas no son almacenadas con tanta permanencia, ya que este puede mejorar su puntaje en caso de que el usuario profesor lo establezca de esa manera. La persistencia de estos y otros datos de este estilo dentro del mismo proyecto demuestra ser mucho menor que los previamente mencionados; el rol que juegan los datos tales como la información de los usuarios al registrarse es fundamental para el funcionamiento del ciclo del programa, mientras que datos como la respuesta a una pregunta de verdadero o falso no es crucial en comparación.

Cabe resaltar, además, que esto no debe representar insignificancia de cualquier dato, sin embargo, la persistencia de los primeros datos tiene mayor peso y relevancia en el funcionamiento del programa que de los segundos mencionados.

## Documentación de la interfaz