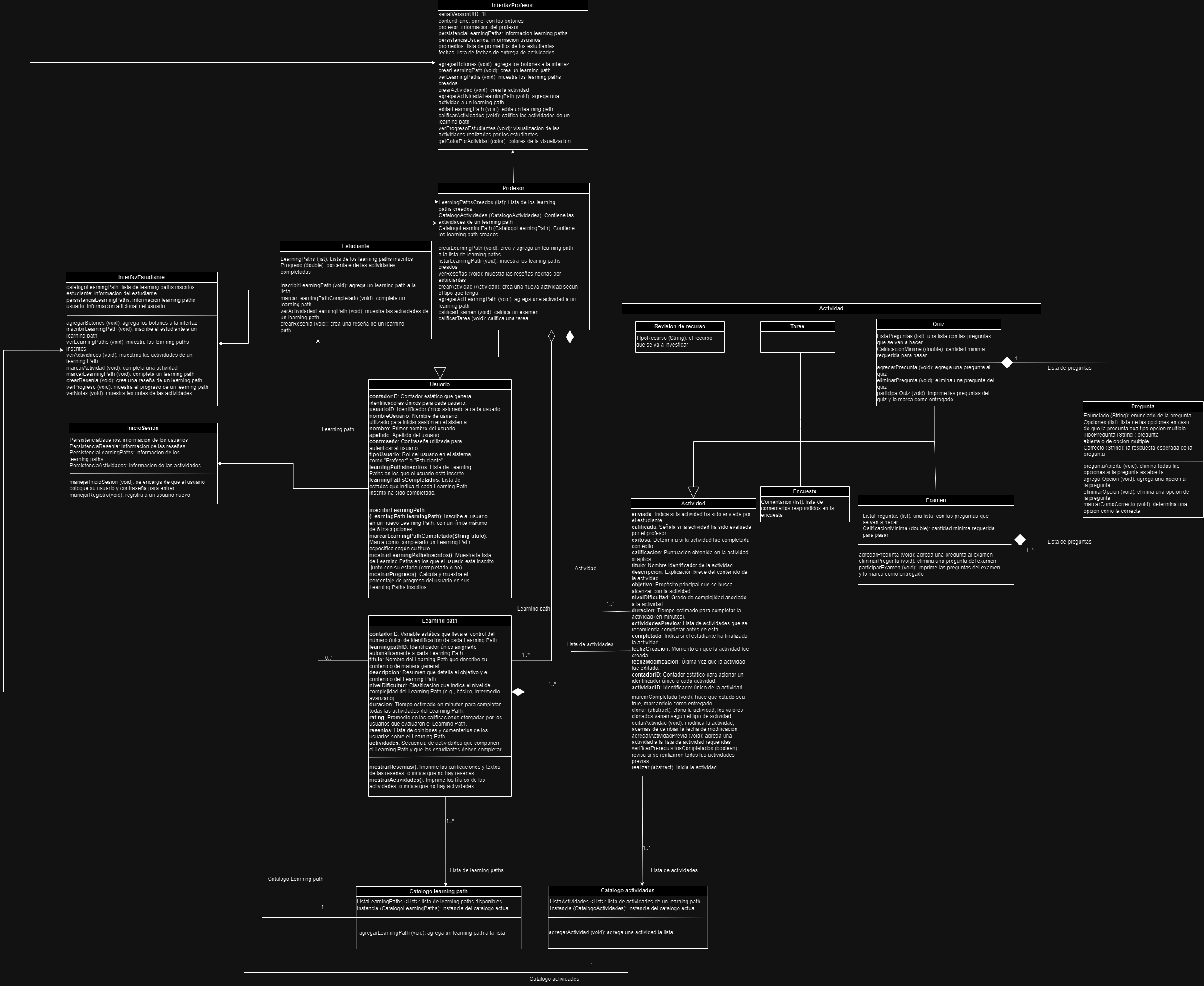
Análisis Learning Path Recommendation System

Nicolás Mario Romero Colmenares – 202321309

Alejandro Guerra López -202122640

Joseph Eli Pulido Gómez - 202211365

1. Diagrama con métodos

Clases:

|  |  |
| --- | --- |
| Nombre | Usuario |
| Resumen | Persona (cliente o profesor) que interactúa con el programa |
| Entradas | * **contadorID**: Contador estático que genera identificadores únicos para cada usuario. * **usuarioID**: Identificador único asignado a cada usuario. * **nombreUsuario**: Nombre de usuario utilizado para iniciar sesión en el sistema. * **nombre**: Primer nombre del usuario. * **apellido**: Apellido del usuario. * **contraseña**: Contraseña utilizada para autenticar al usuario. * **tipoUsuario**: Rol del usuario en el sistema, como "Profesor" o "Estudiante". * **learningPathsInscritos**: Lista de Learning Paths en los que el usuario está inscrito. * **learningPathsCompletados**: Lista de estados que indica si cada Learning Path inscrito ha sido completado. |
| Metodos | Constructor, setters y getters, ademas de:   * **inscribirLearningPath(LearningPath learningPath)**: Inscribe al usuario en un nuevo Learning Path, con un límite máximo de 6 inscripciones. * **marcarLearningPathCompletado(String titulo)**: Marca como completado un Learning Path específico según su título. * **mostrarLearningPathsInscritos()**: Muestra la lista de Learning Paths en los que el usuario está inscrito junto con su estado (completado o no). * **mostrarProgreso()**: Calcula y muestra el porcentaje de progreso del usuario en sus Learning Paths inscritos. |

|  |  |
| --- | --- |
| Nombre | Estudiante |
| Resumen | Persona que puede inscribir, listar learning paths, o revisar su progreso en uno |
| Entradas | * LearningPaths (list): Lista de los learning paths inscritos * Progreso (double): porcentaje de las actividades completadas |
| Metodos | Constructor, setters, getters, ademas de:   * InscribirLearningPath (void): agrega un learning path a la lista * marcarLearningPathCompletado (void): completa un learning path * verActividadesLearningPath (void): muestra las actividades de un learning path * crearResenia (void): crea una reseña de un learning path |

|  |  |
| --- | --- |
| Nombre | Profesor |
| Resumen | El profesor crea una actividad y la asigna a el Learning Path especificado, el cual puede tener sugerencias de actividades previas y un estado que indica si fue enviada/completada o no. |
| Entradas | * LearningPaths (list): Lista de los learning paths creados * CatalogoActividades (CatalogoActividades): Catalogo de las actividades de un learning path * CatalogoLearningPath (CatalogoLearningPath): Catalogo de los learning path disponibles |
| Metodos | Constructor, setters, getters, ademas de:   * crearLearningPath (void): crea y agrega un learning path a la lista de learning paths * listarLearningPath (void): muestra los leaning paths creados * verReseñas (void): muestra las reseñas hechas por estudiantes * crearActividad (Actividad): crea una nueva actividad segun el tipo que tenga * agregarActLearningPath (void): agrega una actividad a un learning path * calificarExamen (void): califica un examen * calificarTarea (void): califica una tarea |

|  |  |
| --- | --- |
| Nombre | LearningPath |
| Resumen | Los learning path son el elemento principal del programa, los cuales contienen actividades para realizar |
| Entradas | * **contadorID**: Variable estática que lleva el control del número único de identificación de cada Learning Path. * **learningpathID**: Identificador único asignado automáticamente a cada Learning Path. * **titulo**: Nombre del Learning Path que describe su contenido de manera general. * **descripcion**: Resumen que detalla el objetivo y el contenido del Learning Path. * **nivelDificultad**: Clasificación que indica el nivel de complejidad del Learning Path (e.g., básico, intermedio, avanzado). * **duracion**: Tiempo estimado en minutos para completar todas las actividades del Learning Path. * **rating**: Promedio de las calificaciones otorgadas por los usuarios que evaluaron el Learning Path. * **resenias**: Lista de opiniones y comentarios de los usuarios sobre el Learning Path. * **actividades**: Secuencia de actividades que componen el Learning Path y que los estudiantes deben completar. |
| Metodos | Constructor, setters, getters, ademas de:   * **mostrarResenias()**: Imprime las calificaciones y textos de las reseñas, o indica que no hay reseñas. * **mostrarActividades()**: Imprime los títulos de las actividades, o indica que no hay actividades. |

|  |  |
| --- | --- |
| Nombre | Actividad |
| Resumen | Una actividad es creada por un profesor, que es colocado en un learning path, donde los estudiantes pueden realizarlos, una actividad puede ser revisar un recurso, una tarea, una encuesta, un quiz o una evaluacion |
| Entradas | * **enviada**: Indica si la actividad ha sido enviada por el estudiante. * **calificada**: Señala si la actividad ha sido evaluada por el profesor. * **exitosa**: Determina si la actividad fue completada con éxito. * **calificacion**: Puntuación obtenida en la actividad, si aplica. * **titulo**: Nombre identificador de la actividad. * **descripcion**: Explicación breve del contenido de la actividad. * **objetivo**: Propósito principal que se busca alcanzar con la actividad. * **nivelDificultad**: Grado de complejidad asociado a la actividad. * **duracion**: Tiempo estimado para completar la actividad (en minutos). * **actividadesPrevias**: Lista de actividades que se recomienda completar antes de esta. * **completada**: Indica si el estudiante ha finalizado la actividad. * **fechaCreacion**: Momento en que la actividad fue creada. * **fechaModificacion**: Última vez que la actividad fue editada. * **contadorID**: Contador estático para asignar un identificador único a cada actividad. * **actividadID**: Identificador único de la actividad. |
| Metodos | Constructor, setters, getters, ademas de:   * marcarCompletada (void): hace que estado sea true, marcandolo como entregado * clonar (abstract): clona la actividad, los valores clonados varian segun el tipo de actividad * editarActividad (void): modifica la actividad, ademas de cambiar la fecha de modificacion * agregarActividadPrevia (void): agrega una actividad a la lista de actividad requeridas * verificarPrerequisitosCompletados (boolean): revisa si se realizaron todas las actividades previas * realizar (abstract): inicia la actividad |

|  |  |
| --- | --- |
| Nombre | ActividadRevision |
| Resumen | Actividad donde se revisa un recurso. Ej: documental |
| Entradas | * TIpoRecurso (String): el recurso que se va a investigar |
| Metodos | Constructor, setters y getters |

|  |  |
| --- | --- |
| Nombre | ActividadTarea |
| Resumen | Actividad para hacer en casa. Ej: taller |
| Entradas | Entradas son iguales en la super clase |
| Metodos | Constructor, setters, getters |

|  |  |
| --- | --- |
| Nombre | ActividadEncuesta |
| Resumen | Actividad donde se obtienen comentarios por parte de los estudiantes en cuanto a algun tema. Ej: evaluacion de profesor |
| Entradas | * Comentarios (list): lista de comentarios respondidos en la encuesta |
| Metodos | Constructor, setters y getters |

|  |  |
| --- | --- |
| Nombre | ActividadQuiz |
| Resumen | Prueba pequeña en donde todas las preguntas son de opcion multiple, hay una calificacion minima para aprovar |
| Entradas | * ListaPreguntas (list): una lista con las preguntas que se van a hacer * CalificacionMinima (double): cantidad minima requerida para pasar |
| Metodos | Constructor, setters, getters, ademas de:   * agregarPregunta (void): agrega una pregunta al quiz * eliminarPregunta (void): elimina una pregunta del quiz * participarQuiz (void): imprime las preguntas del quiz y lo marca como entregado |

|  |  |
| --- | --- |
| Nombre | ActividadExamen |
| Resumen | Prueba donde las preguntas pueden ser abiertas o cerradas, al entregarlos, el profesor debe calificar la actividad similar a una tarea |
| Entradas | * ListaPreguntas (list): una lista con las preguntas que se van a hacer * CalificacionMinima (double): cantidad minima requerida para pasar |
| Metodos | Constructor, setters, getters, ademas de:   * agregarPregunta (void): agrega una pregunta al examen * eliminarPregunta (void): elimina una pregunta del examen |

|  |  |
| --- | --- |
| Nombre | Pregunta |
| Resumen | Una pregunta que puede aparecer en un quiz, puede ser abierta o cerrada, en caso de ser cerrada debe tener un máximo de 4 opciones, una de las cuales debe ser correcta |
| Entradas | * Enunciado (String): enunciado de la pregunta * Opciones (list): lista de las opciones en caso de que la pregunta sea tipo opcion múltiple * TipoPregunta (String): pregunta abierta o de opción múltiple * Correcto (String): la respuesta esperada de la pregunta |
| Resultados | Constructor, setters, getters, ademas de:   * preguntaAbierta (void): elimina todas las opciones si la pregunta es abierta * agregarOpcion (void): agrega una opción a la pregunta * eliminarOpcion (void): elimina una opción de la pregunta * marcarComoCorrecto (void): determina una opción como la correcta |

|  |  |
| --- | --- |
| Nombre | Catalogo Learning Paths |
| Resumen | Los learning paths que han sido creados para la visualizacion |
| Entradas | * ListaLearningPaths <List>: lista de learning paths disponibles * Instancia (CatalogoLearningPaths): instancia del catálogo actual |
| Resultados | Constructor, setters, getters, ademas de:   * agregarLearningPath (void): agrega un learning path a la lista |

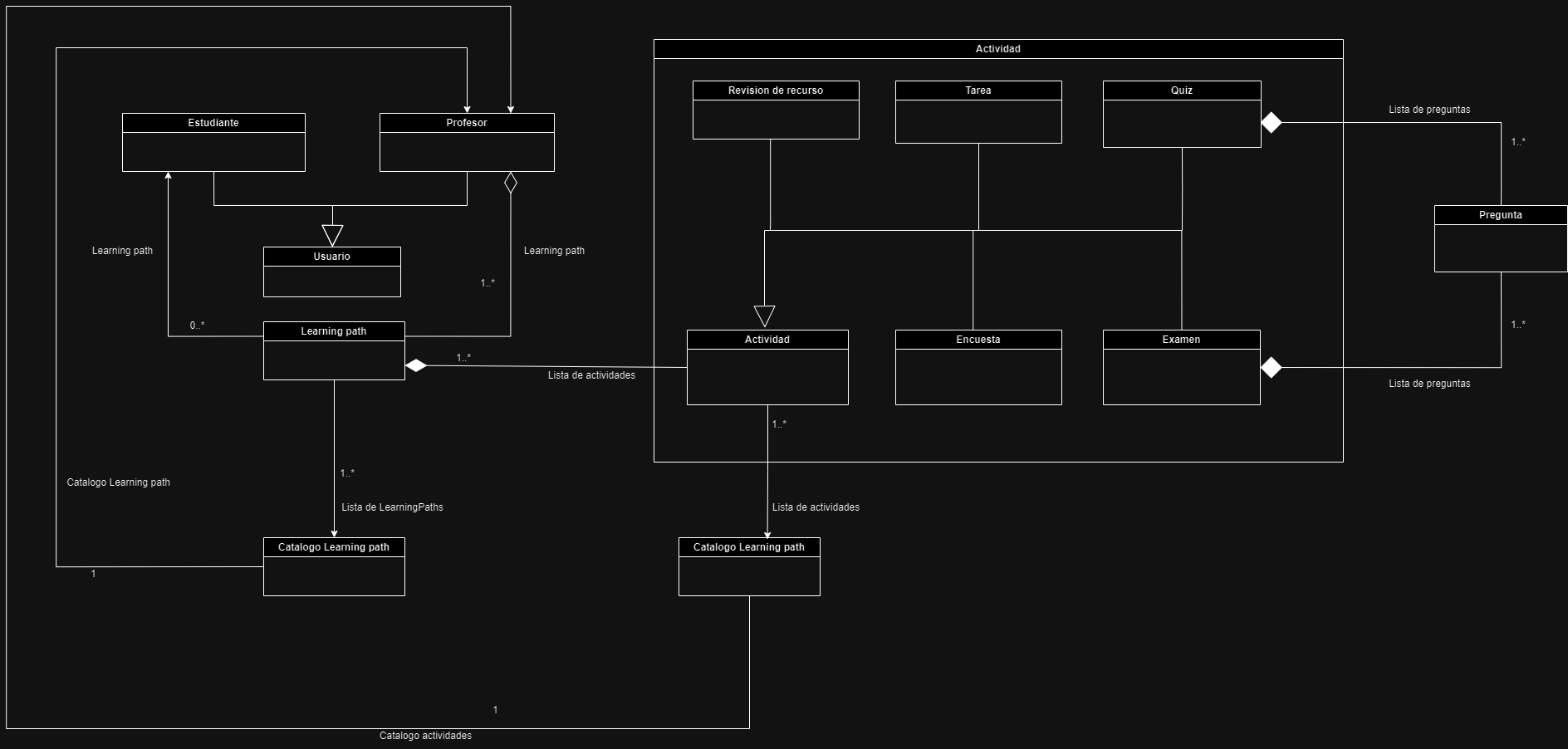
|  |  |
| --- | --- |
| Nombre | Catalogo Actividades |
| Resumen | Las actividades que hacen parte de un learning path |
| Entradas | * ListaActividades <List>: lista de actividades de un learning path * Instancia (CatalogoActividades): instancia del catálogo actual |
| Resultados | Constructor, setters, getters, ademas de:   * agregarActividad (void): agrega una actividad la lista |

|  |  |
| --- | --- |
| Nombre | InicioSesion |
| Resumen | Sistema de inicio de sesion |
| Entradas | * PersistenciaUsuarios: informacion de los usuarios * PersistenciaResenia: informacion de las reseñas * PersistenciaLearningPaths: informacion de los learning paths * PersistenciaActividades: informacion de las actividades |
| Resultados | * manejarInicioSesion (void): se encarga de que el usuario coloque su usuario y contraseña para entrar * manejarRegistro(void): registra a un usuario nuevo |

|  |  |
| --- | --- |
| Nombre | InterfazEstudiante |
| Resumen | Interfaz que ve un estudiante al usar la aplicacion |
| Entradas | * catalogoLearningPath: lista de learning paths inscritos * estudiante: informacion del estudiante * persistenciaLearningPaths: informacion learning paths * usuario: informacion adicional del usuario |
| Resultados | * agregarBotones (void): agrega los botones a la interfaz * inscribirLearningPath (void): inscribe el estudiante a un learning path * verLearningPaths (void): muestra los learning paths inscritos * verActividades (void): muestras las actividades de un learning Path * marcarActividad (void): completa una actividad * marcarLearningPath (void): completa un learning path * crearResenia (void): crea una reseña de un learning path * verProgreso (void): muestra el progreso de un learning path * verNotas (void): muestra las notas de las actividades |

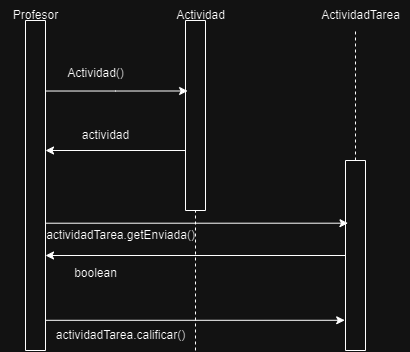
|  |  |
| --- | --- |
| Nombre | InterfazProfesor |
| Resumen | Interfaz que ve un profesor al usar la aplicacion |
| Entradas | * serialVersionUID: 1L * contentPane: panel con los botones * profesor: informacion del profesor * persistenciaLearningPaths: informacion learning paths * persistenciaUsuarios: informacion usuarios * promedios: lista de promedios de los estudiantes * fechas: lista de fechas de entrega de actividades |
| Resultados | * agregarBotones (void): agrega los botones a la interfaz * crearLearningPath (void): crea un learning path * verLearningPaths (void): muestra los learning pathscreados * crearActividad (void): crea la actividad * agregarActividadALearningPath (void): agrega una * actividad a un learning path * editarLearningPath (void): edita un learning path * calificarActividades (void): califica las actividades de un learning path * verProgresoEstudiantes (void): visualizacion de las * actividades realizadas por los estudiantes * getColorPorActividad (color): colores de la visualizacion |

1. Diagrama de relaciones

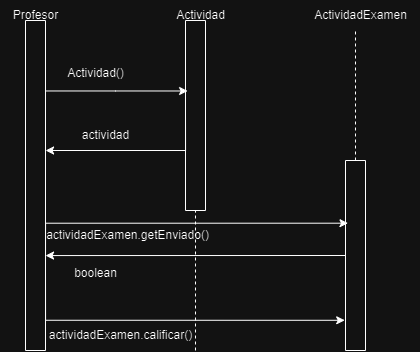


Relaciones:

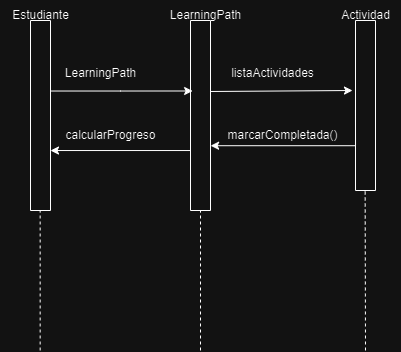
* Usuario:
  + Estudiante y profesor heredan de usuario
    - Learning path a estudiante: asociacion, puede tener varios learning paths inscritos a la vez
    - Learning path a profesor: Un Profesor puede crear múltiples Learning Paths, pero un Learning Path puede existir sin ese profesor (otro profesor podría copiarlo o modificarlo).
  + Profesor accede a CatalogoLearningPath y CatalogoActividades para poder visualizar la informacion de un learning path
* Learning path:
  + Learning path a actividad: Composición, un learning path puede existir sin actividades, pero una actividad no puede existir sin un learning path, un learning path puede tener varias actividades
* Actividad:
  + ActividadRecurso, ActividadTarea, ActividadEncuesta, ActividadQuiz y ActividadEvaluacion heredan de Actividad ya que Actividad es una clase abstracta
  + Tanto ActividadQuiz como Actividad Evaluación tienen relación de composición con Pregunta ya que un quiz o examen puede tener una o varias preguntas, pero las preguntas dependen de la existencia del quiz
* CatalogoLearningPath:
  + Este accede a los diferentes learning paths creados
* CatalogoActividad:
  + Este accede a las actividades de un learning path
* InicioSesion:
  + Este accede a la información de los usuarios
* InterfazEstudiante:
  + Este accede a estudiante y su infomacion ademas de acceder a usuario y a learning path
* InterfazLearningPath
  + Este accede a profesor y su información además de acceder a usuario y a learning path

1. Diagramas de secuencia
   1. Calificación de tarea

El profesor empieza creando la actividad, la cual, para calificarla debe de asegurarse de que el estudiante la haya entregado, por lo que se ve si getEnviada() es true, en caso de que lo sea, el proceso de calificación sigue como es usual, de lo contrario, mencionara que no es posible calificar al estudiante todavia.

* 1. Calificación de examen

Proceso similar a la tarea, en donde difiere es durante el proceso de calificación, donde el profesor revisa las respuestas del estudiante.

* 1. Revision de progreso

El estudiante accede al learning path, donde este muestra una lista de actividades que este contiene, de ahi, se puede ver cuáles han sido marcadas como completadas para que learning path haga un cálculo y así saber el proceso del learning path

1. Historias de usuarios
   1. Estudiante:

Carlos es un estudiante de la Universidad de Los Andes, el cual tiene inscrito un learning path titulado “historia de la música en Colombia”, el cual tiene un id de: 3472 con una dificultad mediana y duracion de 180 minutos por semana, el estudiante tiene que enviar la tarea denominada “Taller de música indígena” donde tiene un documento con la tarea lista, el estudiante está listo para enviar la tarea, sin embargo, resulta que el estudiante no ha realizado una actividad de revisión de recurso previa con el nombre “Ejemplos de canciones indígenas”, por lo que si quiere enviar la tarea, tiene que primero revisar los ejemplos, y en caso de que vea algo que cambiar o agregar, modificar el taller para que la tarea sea correcta

* 1. Profesor creador:

Jaime es un profesor, el cual va a crear un learning path, el cual vea el tema de biodiversidad en la flora colombiana. Jaime creo el learning path de forma en la que el nivel de dificultad es bajo, con 120 minutos por semana, y ahora el Javier va a agregar 3 actividades de revisión de recurso, 2 de tareas y 2 de exámenes para que los estudiantes puedan realizar a medida que se va progresando por el learning path.

* 1. Profesor seguimiento:

Mariana es una profesora que hace parte del learning path con el título “Cinematografía de la década 1950”, este learning path abarca temas históricos, culturales y políticos, con una dificultad intermedia que requiere 240 minutos por semana, tiene 3.2 en rating y contiene 4 tareas, 5 quizes y 1 examen. Debido a que los estudiantes están en la mitad del semestre, 3 de las tareas deberían de haber sido enviadas para que Mariana las revise, sin embargo, un grupo de 4 estudiantes tan solo ha enviado 2 de las tareas asignadas, por lo que Mariana debe de calificar a esos estudiantes sea con 0 o con una penalización por haber entregado la tarea tarde, despues de haber calificado a cada estudiante, Mariana marca las tareas como completadas.