

Angela Bárcenas -202320484
Tomas Hernandez 202320326
Juan David Uribe - 202322433

Historias de Usuario

Codigo	Descripción	Criterios de aceptación	Entradas	Salidas
HU-101 profesor creador	<p>Como profesor</p> <p>Quiero crear y editar learningpaths</p> <p>Para que el sistema pueda mostrarlos y los estudiantes se puedan inscribir</p>	<ol style="list-style-type: none">1. Los learnignPaths deben tener un profesor asignado (el profesor que los crea)2. Los estudiantes deben poder inscribirse al learningPath3. Los learningPaths deben tener una o mas actividades4. Los profesores deberían poder clonar y editar el learningPath	<ul style="list-style-type: none">• El profesor ingresa las actividades correspondientes del learningPath• Le da un nombre al learningPath	<ul style="list-style-type: none">• El learningPath queda terminado y disponible en el sistema para que los estudiantes se inscriban
HU-102 profesor creador	<p>Como profesor</p> <p>Quiero agregar diferentes tipos de actividades (tareas, quizzes, exámenes) a los Learning Paths,</p> <p>Para que el sistema pueda mostrarlos y los estudiantes tengan una variedad de recursos y evaluaciones en su ruta de aprendizaje.</p>	<ol style="list-style-type: none">1. El sistema permite seleccionar y añadir actividades de diferentes tipos (tarea, examen, quiz, recurso, encuesta).2. El sistema actualiza el Learning Path para incluir las nuevas actividades y muestra un esquema detallado.3. El sistema permite asociar pre-requisitos y determinar si la actividad es obligatoria u opcional.	<ul style="list-style-type: none">• El profesor ingresa Tipo de actividad, descripción, duración, pre-requisitos, obligatoriedad.	<ul style="list-style-type: none">• Actualización del Learning Path, vista de actividades añadidas.

Codigo	Descripción	Criterios de aceptación	Entradas	Salidas
HU-103 profesor creador	<p>Como profesor</p> <p>Quiero seleccionar, clonar y adaptar actividades ya existentes en el sistema,</p> <p>Para reutilizar contenido útil y crear rutas de aprendizaje de manera más eficiente.</p>	<ol style="list-style-type: none">1. El sistema permite al profesor buscar y seleccionar actividades existentes.2. El sistema permite editar una copia de la actividad seleccionada y agregarla al Learning Path.3. La actividad adaptada se muestra en el Learning Path con sus nuevas características.4. La copia de la actividad es totalmente independiente a la original.	<ul style="list-style-type: none">• El profesor selecciona la actividad e ingresa las nuevas características.	<ul style="list-style-type: none">• el sistema retorna la Actividad editada y añadida al Learning Path, y confirma la adaptación.
HU-104 profesor creador	<p>Como profesor</p> <p>Quiero consultar las reseñas y calificaciones de actividades previas al integrarlas en mi Learning Path</p> <p>Para asegurar que el contenido seleccionado sea de calidad , de agrado y adecuado para los estudiantes.</p>	<ol style="list-style-type: none">1. El sistema permite ver el promedio de calificaciones y reseñas de cada actividad.2. Las reseñas se muestran junto con una fecha y nombre del revisor.	<ul style="list-style-type: none">• El profesor selecciona la actividad que desea inpeccionar	<ul style="list-style-type: none">• el sistema retorna la información de la actividad (promedio y listado de reseñas)

Codigo	Descripción	Criterios de aceptación	Entradas	Salidas
HU-201 estudiante	<p>Como estudiante</p> <p>Quiero explorar los learning Paths disponibles</p> <p>Para elegir los que se alineen con mis intereses y objetivos de aprendizaje.</p>	<ol style="list-style-type: none">1. El sistema muestra una lista de Learning Paths con su descripción, nivel y duración.2. Los Learning Paths pueden ser filtrados por categoria, nivel de dificultad y duración.3. El sistema permite al estudiante seleccionar un Learning Path para obtener más detalles.	<ul style="list-style-type: none">• El estudiante ingresa sus criterios de búsqueda y seleccionó las LearningPaths de su interes	<ul style="list-style-type: none">• Se retorna una lista de Lerninpaths y los detalles de los Learningpaths seleccionados
HU-202 estudiante	<p>Como estudiante</p> <p>Quiero inscribirme en Learning Paths y ver su estructura y actividades</p> <p>Para organizar y planificar mi progreso de acuerdo con mis preferencias.</p>	<ol style="list-style-type: none">1. El sistema permite al estudiante inscribirse en un Learning Path.2. Una vez inscrito, el sistema muestra la estructura del Learning Path y sus actividades.3. El sistema registra la inscripción.	<ul style="list-style-type: none">• El estudiante selecciona el LeraningPath al cual se quiere inscribir	<ul style="list-style-type: none">• Se retorna la confirmación de la inscripción• se muestra la estructura completa del LearningPath

Codigo	Descripción	Criterios de aceptación	Entradas	Salidas
HU-203 estudiante	<p>Como estudiante</p> <p>Quieroexplorar actividades y ver el avance que tengo en las mismas</p> <p>Para mantener el seguimiento de mi progreso y completar todas las tareas necesarias</p>	<ol style="list-style-type: none">1. El sistema permite al estudiante iniciar una actividad y registrar cuando la completa.2. El sistema actualiza automáticamente el porcentaje de progreso en función de las actividades completadas.3. El sistema muestra el estado de la actividad como "completada" o "en progreso".	<ul style="list-style-type: none">• El estudiante selecciona la actividad que desea realizar y se muestra en que estado se encuentra la actividad	<ul style="list-style-type: none">• Se retorna el estado de avance de la actividad• se muestrael porcentaje de avance del LearningPath
HU-204 estudiante	<p>Como estudiante</p> <p>Quiero recibir feedback sobre mis respuestas en quizzes.</p> <p>Para comprender mis errores y mejorar mi aprendizaje</p>	<ol style="list-style-type: none">1. El sistema muestra los resultados del quiz inmediatamente después de finalizar.2. Cada pregunta del quiz incluye la corrección y la respuesta correcta.3. El sistema indica si el estudiante aprobó o no el quiz basado en el puntaje mínimo	<ul style="list-style-type: none">• El estudiante da su respuesta en el quiz.	<ul style="list-style-type: none">• Se retorna el resultado del quiz, las respuestas correctas y el estado de aprobado o no aprobado.

Codigo	Descripción	Criterios de aceptación	Entradas	Salidas
HU-205 estudiante	<p>Como estudiante</p> <p>Quiero calificar y comentar las actividades que realizo.</p> <p>Para mejorar el contenido de los Learning Paths</p>	<ol style="list-style-type: none">1. El sistema permite al estudiante asignar una calificación y escribir un comentario para cada actividad.2. La calificación y el comentario se guardan y son accesibles para el profesor.3. Los estudiantes pueden revisar y editar su calificación y comentario después de publicarlos.4. Los comentarios son visibles para todos los usuarios registrados en la plataforma.	<ul style="list-style-type: none">• El estudiante ingresa su calificación numérica y su comentario sobre la actividad.	<ul style="list-style-type: none">• Se guarda la modificación de la califiacio2n y el comentario del estudiante y se guarda en el historial de calificaciones.

Codigo	Descripción	Criterios de aceptación	Entradas	Salidas
HU-301 profesor en rol de seguimiento y evaluador	<p>Como profesor</p> <p>Quiero monitorear el proceso de los estudiantes en un LerningPath</p> <p>Para identificar su nivel de avance y dificultades en el cumplimiento de actividades.</p>	<ol style="list-style-type: none">1. El sistema muestra el porcentaje de progreso para cada estudiante inscrito en el Learning Path.2. Se detallan las actividades completadas, en progreso y pendientes, con sus respectivas fechas de inicio y finalización.3. El sistema muestra el progreso general de los estudiantes en un LearninPath	<ul style="list-style-type: none">• El profesor selecciona un LearningPath• El profesor puede seleccionar un estudiante el particular	<ul style="list-style-type: none">• Se retorna el porcentaje de progreso, la lista de actividades con estado (completada/ en progreso/ pendiente) y las fechas de inicio y finalización.
HU-302 profesor en rol de seguimiento y evaluador	<p>Como profesor</p> <p>Quiero evaluar y calificar las tareas y exámenes enviados por los estudiantes,</p> <p>Para que los estudiantes reciban retroalimentación y mejoren sus conocimientos en función de los criterios establecidos</p>	<ol style="list-style-type: none">1. El sistema permite al profesor ver la lista de tareas y exámenes enviados por los estudiantes en estado "pendiente de evaluación".2. El sistema permite al profesor calificar las respuestas y añadir comentarios de retroalimentación para cada tarea o examen.3. El sistema registra y actualiza el estado de la actividad una vez calificada (por ejemplo, "exitosa" o "no exitosa").	<ul style="list-style-type: none">• se recibe la actividad enviada por cada estudiante y el profesor ingresa su calificación y sus comentarios.	<ul style="list-style-type: none">• Se retorna el estado actualizado de la actividad (exitosa/no exitosa), comentarios de retroalimentación y la calificación.

Actualización del documento:

Los cambios realizados para esta nueva entrega son:

Creamos un módulo de persistencias que carga los archivos en formato json, para hacer la carga y almacenamiento de archivos de forma más rápida al permitir crear una instancia de cada clase independiente desde cada uno de los controladores disminuyendo así el acoplamiento.

- test

Para la implementación de pruebas automatizadas de la aplicación, estas se dividieron en tres categorías principales. En primer lugar, se desarrollaron pruebas unitarias para cada clase de lógica, abarcando elementos clave como actividades, estudiantes, profesores, preguntas de diferentes tipos, opciones y registros. Estas pruebas aseguran el correcto funcionamiento de cada componente individual, incluyendo clases como `ActividadTest`, `EstudianteTest` y `PreguntaAbiertaTest`. Luego, se implementaron pruebas unitarias para cada controlador, con el objetivo de verificar el funcionamiento adecuado de los métodos y las interacciones en controladores como `ControladorEstudianteTest` y `ControladorActividadTest`. Finalmente, se diseñaron pruebas de integración basadas en historias de usuario, las cuales simulan escenarios reales de uso. Estas pruebas están representadas por los códigos presentes en el documento de historias de usuario, como `HU10X.java`, `HU20X.java` y `HU30X.java`, evidenciando cómo interactúan las clases del programa en diferentes flujos de trabajo.

- pregunta verdadero o falso

Esta entrega nos pedía añadir este tipo de pregunta por ende creamos la clase abstracta de preguntas, de la cual las características generales las heredan las preguntas abiertas, las de selección múltiple y las de verdadero o falso.

- consolas

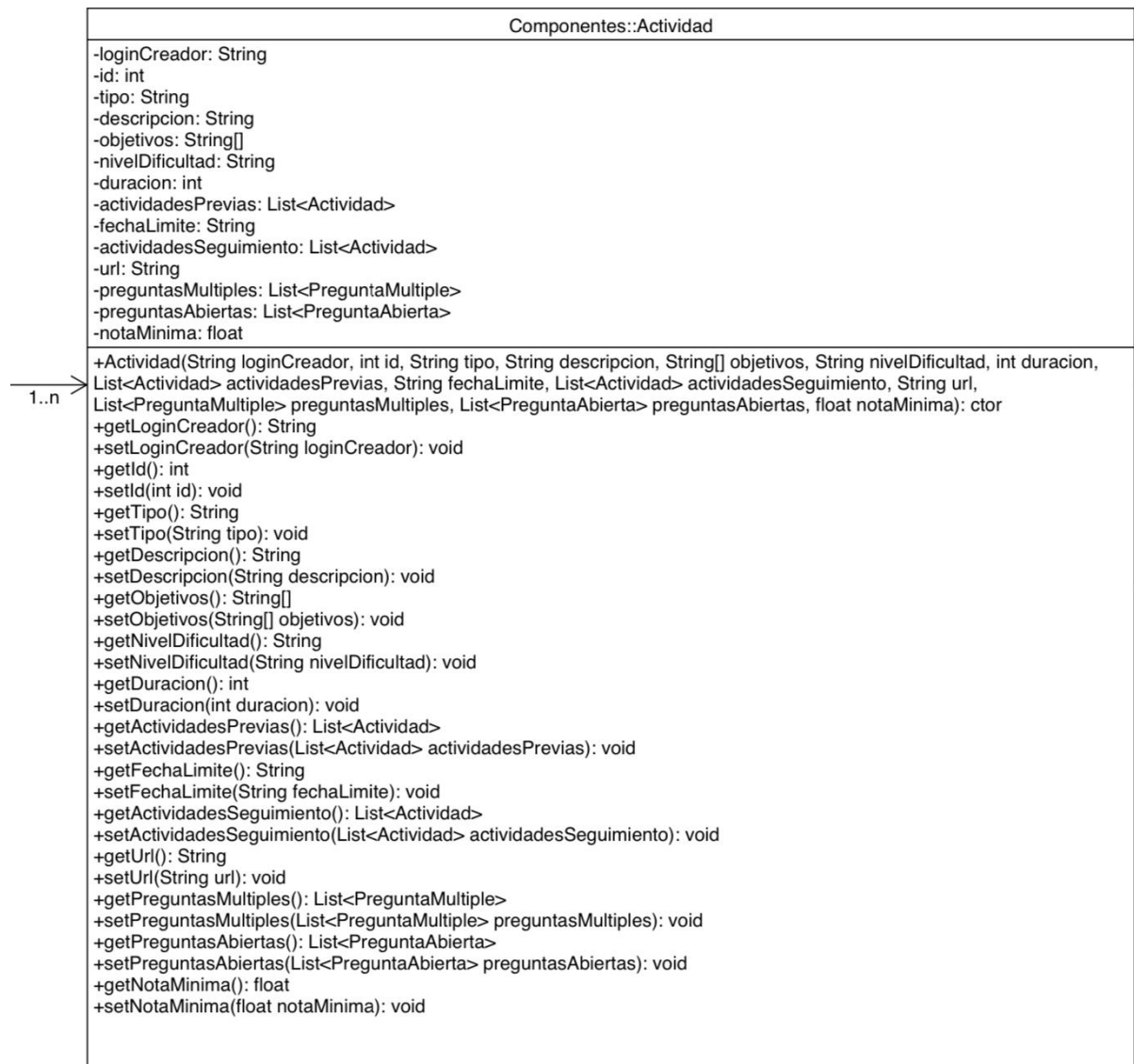
El diseño de las consolas se centra en proporcionar menús específicos para cada tipo de usuario (estudiante y profesor) tras un proceso de autenticación obligatorio. Cada menú presenta opciones claras asociadas a números, y las entradas del usuario se validan para garantizar su corrección. Los datos de la aplicación se cargan automáticamente desde archivos al inicio y se persisten en tiempo real tras cada cambio, asegurando continuidad. La estructura modular de las consolas permite una fácil extensión y mantenimiento, separando funcionalidades por clases específicas y evitando código complejo o redundante.

A. Un diagrama de clases de diseño que incluya todas las clases, incluyendo sus relaciones, atributos y métodos.

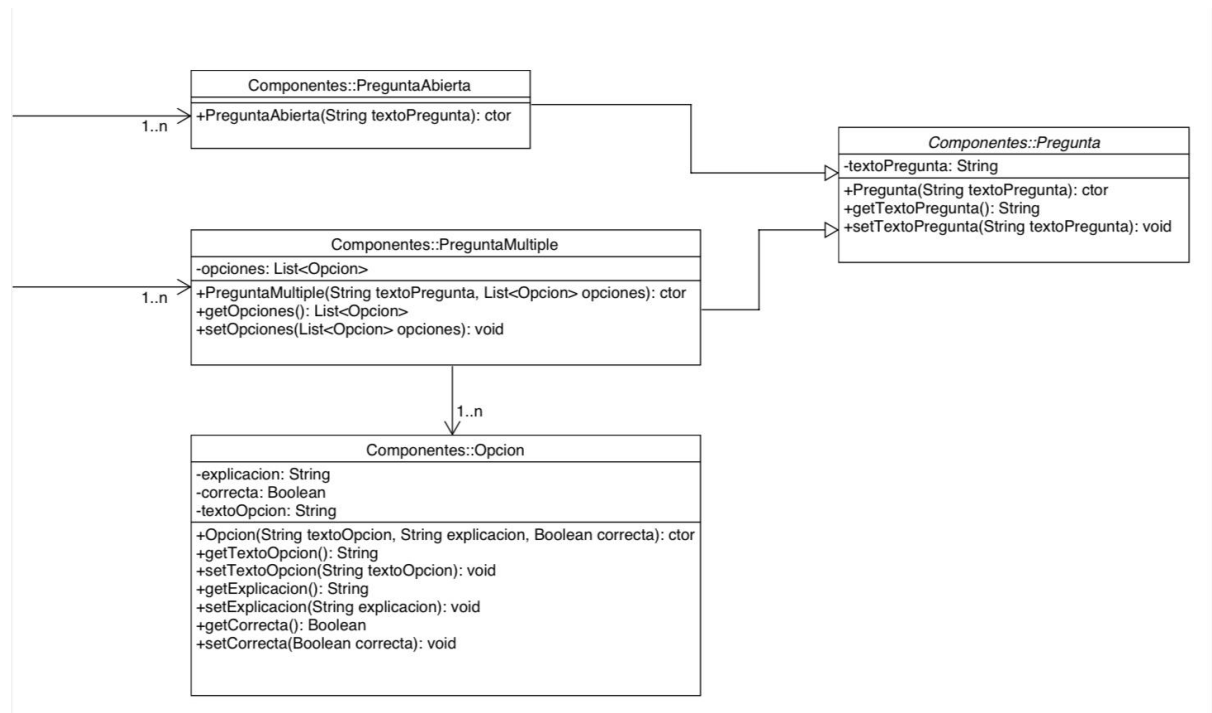
A continuación vamos a descomponer por partes el diagrama anterior, con el fin de entenderlo mejor y poder visualizar de manera clara cada clase con sus métodos y atributos.

Controladores::ControladorActividad	
~actividades: HashMap<Integer, Actividad>	
+ControladorActividad(): ctor	
+ImprimirActividades(): void	
+ImprimirActividades(ArrayList<Integer> ids): void	
+GetActividad(int id): Actividad	
+GetPreguntasAbiertas(Actividad a): List<PreguntaAbierta>	
+GetPreguntasMultiples(Actividad a): List<PreguntaMultiple>	
+VerificarCreador(int id, String loginActual): boolean	
+GetActividades(List<Integer> ids): List<Actividad>	
+CrearActividadExamen(String loginCreador, String tipo, String descripcion, String stringObjetivos, String nivelDificultad, int duracion, List<Integer> idActividadesPrevias, String fechaLimite, List<Integer> idActividadesSeguimiento, List<String> preguntas, float notaMinima): void	
-CrearPreguntas(List<String> preguntas): List<PreguntaAbierta>	
+CrearActividadQuiz(String loginCreador, String tipo, String descripcion, String stringObjetivos, String nivelDificultad, int duracion, List<Integer> idActividadesPrevias, String fechaLimite, List<Integer> idActividadesSeguimiento, HashMap<String, HashMap<String, String> preguntas, List<Integer> correctas, float notaMinima): void	
+CrearPreguntasMultiples(HashMap<String, HashMap<String, String> preguntas, List<Integer> correctas): List<PreguntaMultiple>	
+CrearActividadEncuesta(String loginCreador, String tipo, String descripcion, String stringObjetivos, String nivelDificultad, int duracion, List<Integer> idActividadesPrevias, String fechaLimite, List<Integer> idActividadesSeguimiento, List<String> preguntas): void	
+CrearActividadRecurso(String loginCreador, String tipo, String descripcion, String stringObjetivos, String nivelDificultad, int duracion, List<Integer> idActividadesPrevias, String fechaLimite, List<Integer> idActividadesSeguimiento, String url): void	
+CrearActividadTarea(String loginCreador, String tipo, String descripcion, String stringObjetivos, String nivelDificultad, int duracion, List<Integer> idActividadesPrevias, String fechaLimite, List<Integer> idActividadesSeguimiento): void	
+CrearActividadExamen(String loginCreador, String tipo, String descripcion, String stringObjetivos, String nivelDificultad, int duracion, List<Integer> idActividadesPrevias, String fechaLimite, List<Integer> idActividadesSeguimiento, List<String> preguntas, float notaMinima, int idActividad): void	
+CrearActividadQuiz(String loginCreador, String tipo, String descripcion, String stringObjetivos, String nivelDificultad, int duracion, List<Integer> idActividadesPrevias, String fechaLimite, List<Integer> idActividadesSeguimiento, HashMap<String, HashMap<String, String> preguntas, List<Integer> correctas, float notaMinima, int idActividad): void	
+CrearActividadEncuesta(String loginCreador, String tipo, String descripcion, String stringObjetivos, String nivelDificultad, int duracion, List<Integer> idActividadesPrevias, String fechaLimite, List<Integer> idActividadesSeguimiento, List<String> preguntas, int idActividad): void	
+CrearActividadRecurso(String loginCreador, String tipo, String descripcion, String stringObjetivos, String nivelDificultad, int duracion, List<Integer> idActividadesPrevias, String fechaLimite, List<Integer> idActividadesSeguimiento, String url, int idActividad): void	
+CrearActividadTarea(String loginCreador, String tipo, String descripcion, String stringObjetivos, String nivelDificultad, int duracion, List<Integer> idActividadesPrevias, String fechaLimite, List<Integer> idActividadesSeguimiento, int idActividad): void	

Controlador actividades; tiene como responsabilidad principal la creación , administración y verificar las diferentes actividades del sistema.

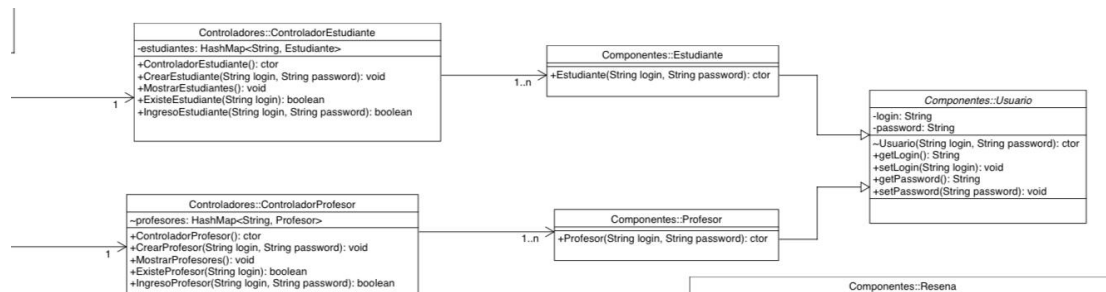


La clase Actividad modela una actividad educativa dentro del sistema, con atributos clave como el creador (loginCreador), identificador (id), tipo, descripción, objetivos, nivel de dificultad, duración, y URLs asociadas. Además, incluye colecciones de actividades previas y de seguimiento (List<Actividad>), así como preguntas abiertas y de opción múltiple asociadas a la actividad. Los métodos de la clase son principalmente getters y setters que permiten modificar y acceder a los atributos. También gestiona atributos como la fecha límite (fechaLimite), el nivel de dificultad, y la nota mínima requerida (notaMinima), lo que sugiere que está diseñada para actividades evaluativas.



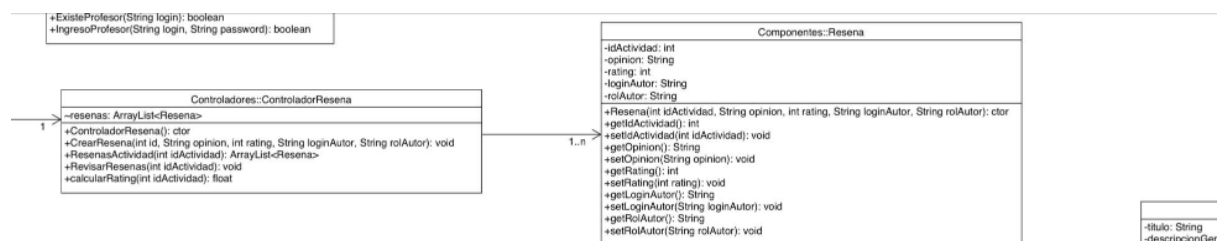
Las clases componentes heredan los atributos y métodos de la clase componentes pregunta, de la cual se desprenden las tres opciones de pregunta existentes en el sistema y que varían en su forma de calificar y en la forma de ser respondidas.

Usuarios:



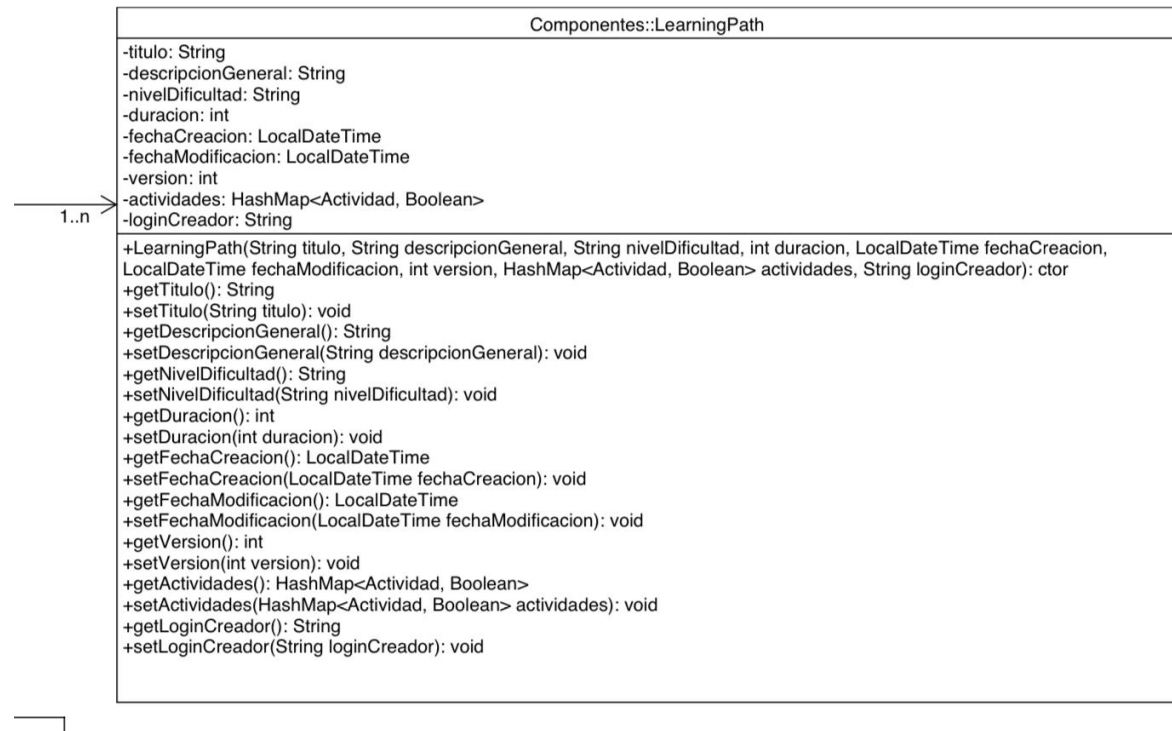
La clase usuario es una clase abstracta de la ua heredan las clases profesor y estudiante; esto debido a que ambas clases comparte atributos, sin embargo sus métodos, permisos y su uso del sistema es totalmente diferente, puesto que los profesores cuenta con más permisos como por ejemplo para crear y editar los learning paths. A pesar de esto tanto profesor como estudiante, son usuarios que necesitan registrarse e ingresar con un login y una contraseña.

Reseñas:



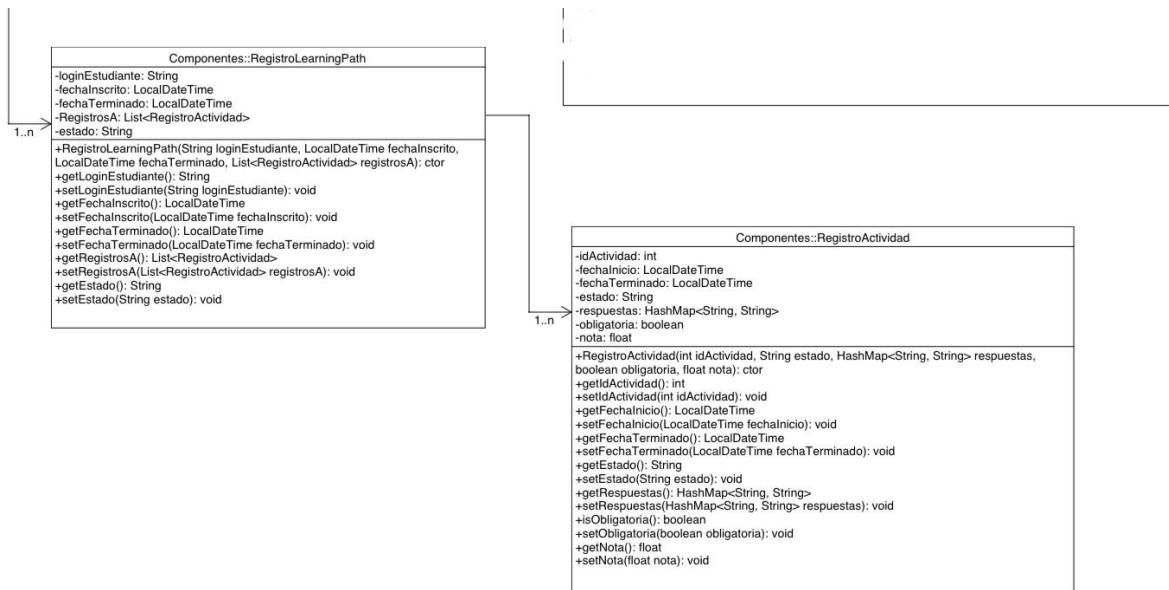
Reseñas son aquellas opiniones que pueden crear los estudiantes acerca de las actividades y los Learning paths, el controlador se encarga de permitir al estudiante crear, revisar y leer las reseñas que tiene que cumplir con los criterios estipulados de la clase reseña.

Learning Paths:



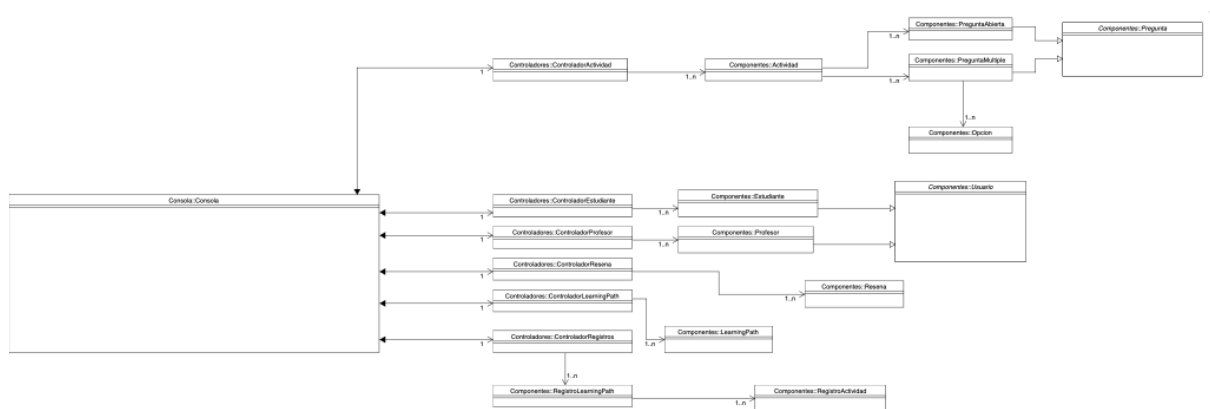
Los learning paths, fundamentales en este sistema vienen ligados con actividades, dado que se componen directamente de ellas, un learning path está conformado por varias actividades p, pueden ser creados únicamente por profesores y deben tener información básica que se estipula en los atributos de la clase.

Registros:



Por último los componentes de registro permiten almacenar la información de cada learning path, que guarda la información de cada actividad que lo compone, es decir en estas clases, se implementa el almacenamiento del avance de cada estudiante en cada una de las actividades, así como su calificación, la fecha en la que lo realizó, el tiempo que le tomó y demás información relevante para el profesor y para el estudiante. Esta clase permite guardar los avances de cada usuario para así facilitar sacar promedios o información tanto genérica como individual de cada uno de los learning paths.

B. Un diagrama de clases de alto nivel, que incluya todas las clases y sus relaciones, pero no todos los métodos ni atributos. Este diagrama facilitará entender las relaciones entre clases.



Las relaciones entre clases, son básicamente asociaciones que nos indican la existencia de una conexión lógica e incluso funcional entre ellas, es decir nos muestra cómo los objetos de una clase se relacionan con la otra. En nuestro diagrama;

Consola:

La Consola actúa como la interfaz donde los usuarios interactúan con el sistema, mientras que los controladores gestionan la lógica y los datos.

- la consola es unidireccional es decir; solo necesita interactuar con los controladores para realizar tareas específicas
- la consola depende de los controladores para poder ejecutar las funcionalidades del sistema

Controladores:

Cada controlador gestiona un conjunto de datos específicos, como ControladorEstudiante gestiona un HashMap de Estudiante y ControladorProfesor un HashMap de Profesor.

- La relación entre cada controlador y su conjunto de datos (estudiantes, profesores, etc.) es de composición, ya que el controlador crea, modifica y borra estos datos, y si se destruye el controlador, también lo hacen sus datos.

Controlador Learnig Path y Actividades:

Para construir y administrar los "Learning Paths", el ControladorLearningPath requiere interactuar con el ControladorActividad, dado que un Learning Path se compone de diversas actividades.

- LearningPath Controlador incluye un HashMap de actividades. Esto nos muestra que se compone de diversas actividades que se vinculan de forma lógica a través del controlador.

Controlador Reseñas y las reseñas:

ControladorResena tiene una agregación con la clase Resena, debido que contiene una lista de Resena (ArrayList<Resena>). Esto implica que el controlador gestiona las reseñas, pero las reseñas pueden existir independientemente del controlador.

- Un ControladorResena puede tener muchas reseñas, como se indica en el uso de una ArrayList.

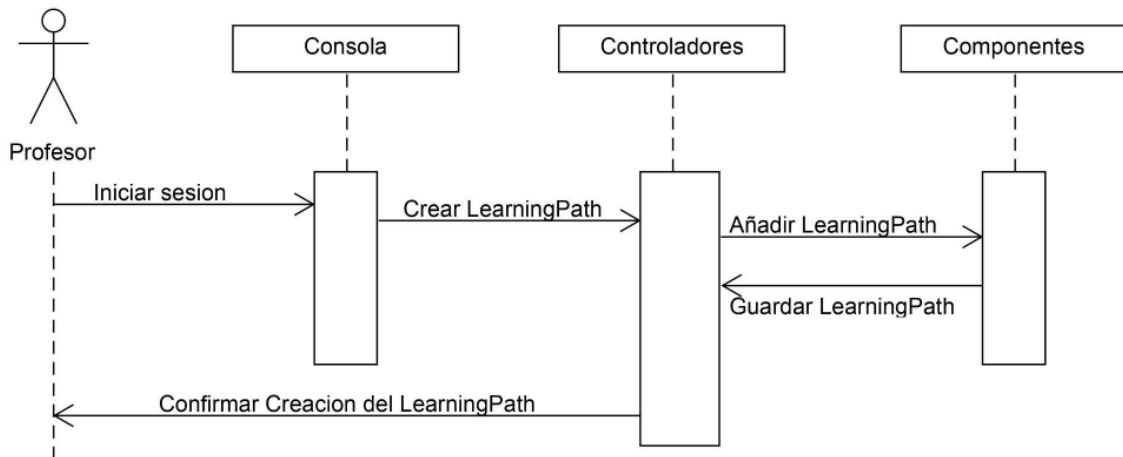
ControladorRegistros y RegistroLearningPath:

- ControladorRegistros gestiona registros de estudiantes inscritos en Learning Paths a través de un HashMap<Integer, ArrayList<RegistroLearningPath>>. Esto indica una relación fuerte en la cual múltiples registros pueden existir para cada Learning Path.

C. Diagramas de secuencia para las funcionalidades que usted considere críticas:

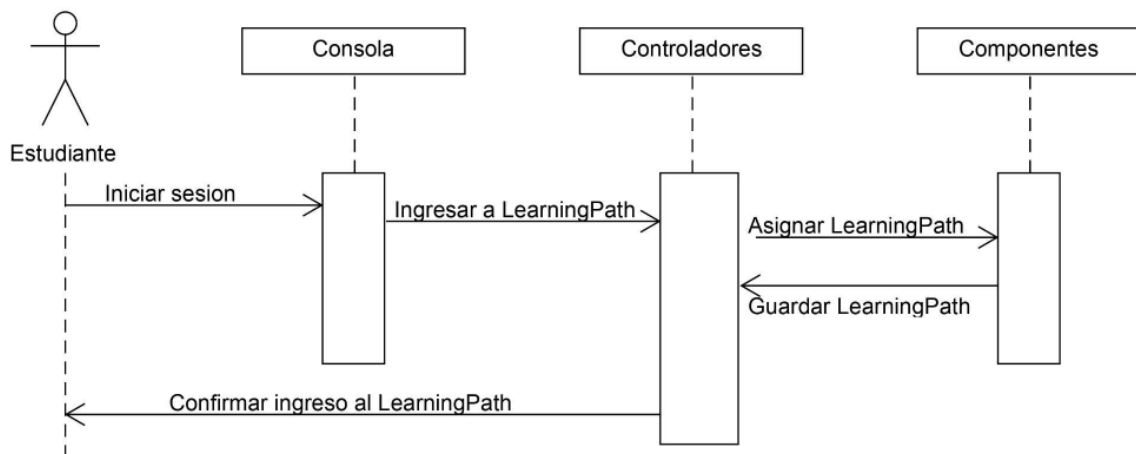
1.

Este diagrama muestra cómo debe interactuar un usuario de tipo profesor con el sistema para poder crear un learning Path, esto se hace de la siguiente manera, el profesor inicia sesión en el sistema para que pueda crear el learning Path, luego da toda la información necesaria para crear un learning Path, esta información se le pasa a los controladores y finalmente a los componentes, donde finalmente le envía la instrucción de guardar el Learning Path a los controladores, y estos se encargan de enviar una confirmación directa al usuario de tipo profesor que el Learning Path fue creado.



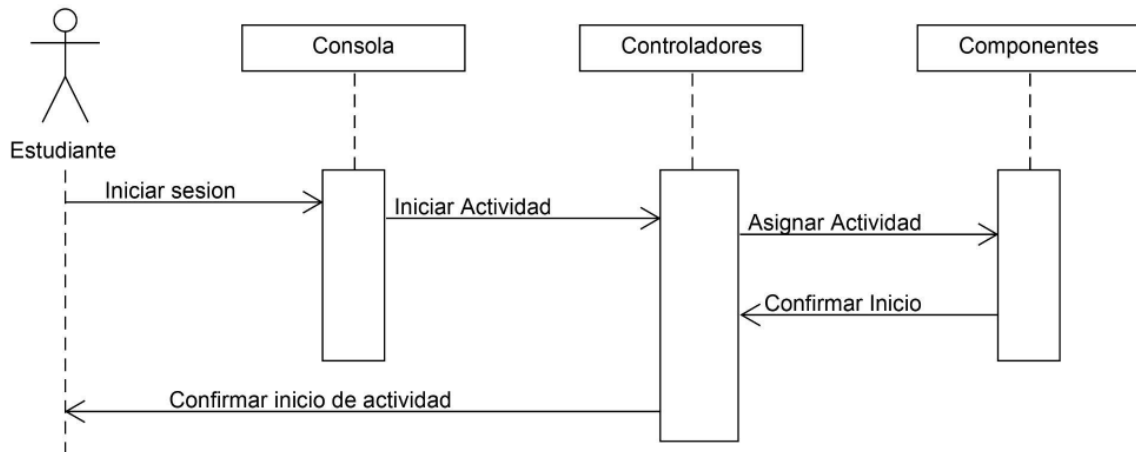
2.

Este diagrama de secuencia ilustra como un usuario de tipo Estudiante interactúa con el sistema para ingresar a un learning Path que le interesa, el estudiante inicia sesión en el sistema para que pueda ingresar al learning Path, esta instrucción se le pasa a los controladores y finalmente a los componentes, donde finalmente le envía la instrucción de guardar el Learning Path a los controladores, y estos se encargan de enviar una confirmación directa al usuario de tipo profesor que el Learning Path fue creado.



3.

En este diagrama se ilustra cómo el usuario de tipo estudiante interactúa con el sistema al momento de iniciar una actividad, para iniciar el estudiante debe iniciar sesión para que pueda dar la instrucción al controlador de iniciar una actividad, este controlador envía una instrucción a los componentes que da inicio a la actividad, el componente envía una confirmación de inicio de actividad al controlador, y este controlador se encarga de informarle al estudiante que la actividad ha sido iniciada.



4.

En este diagrama se evidencia como un usuario de tipo Profesor debe interactuar con el sistema para poder agregar una actividad nueva, esto se hace de la siguiente manera, el profesor inicia sesión en el sistema para que pueda agregar la actividad, luego da toda la información necesaria para agregarla, esta información se le pasa a los controladores y finalmente a los componentes, donde finalmente le envía la instrucción de agregar la actividad a los controladores, y estos se encargan de enviar una confirmación directa al usuario de tipo profesor que la actividad fue creada.

