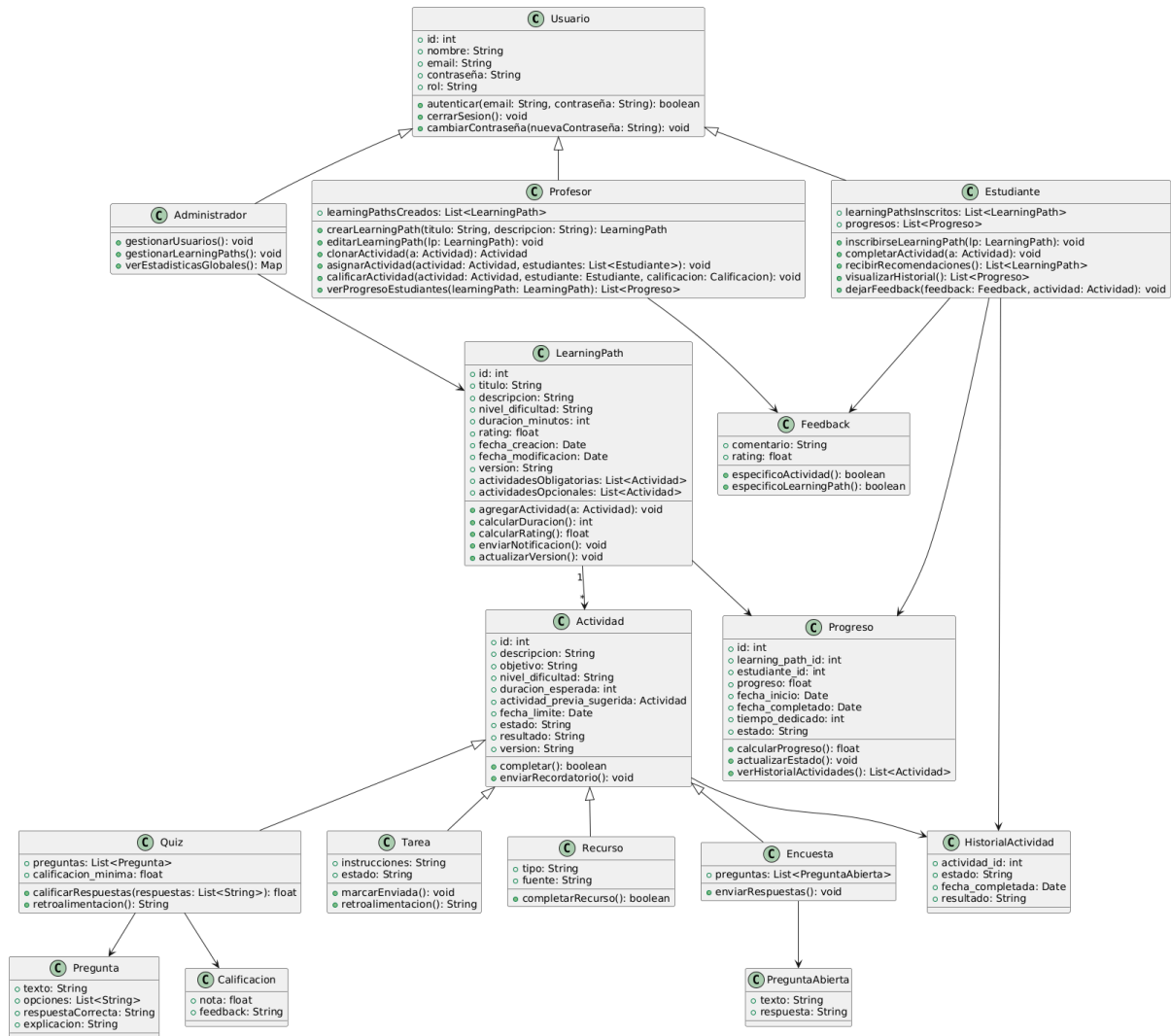
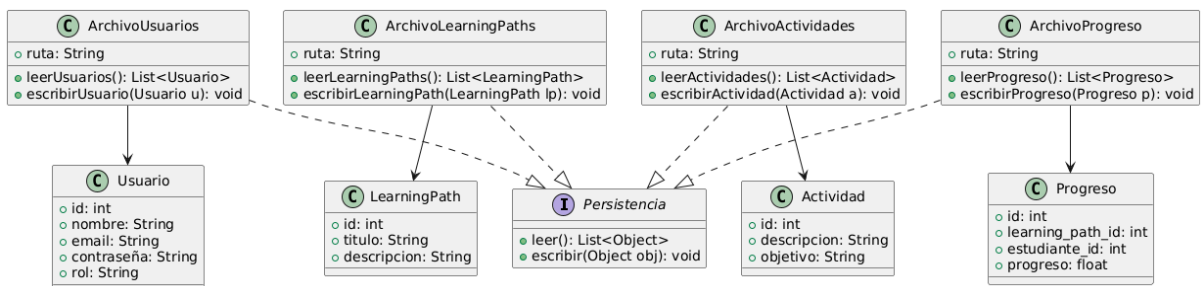


Documento de análisis

I. Modelo De Conceptos



Modelo de Conceptos (Persistencia)



a. Entidades

1. **Usuario**
 2. **Administrador**
 3. **Profesor**
 4. **Estudiante**
 5. **Learning Path**
 6. **Actividad**
 - Quiz
 - Tarea
 - Recurso
 - Encuesta
 7. **Progreso**
 8. **HistorialActividad**
 9. **Pregunta**
 10. **PreguntaAbierta**
 11. **Calificacion**
 12. **Feedback**
-

b. Características (atributos y métodos) de las entidades

1. **Usuario**
 - **Atributos:**
 - **id**: int - Identificador único del usuario.
 - **nombre**: String - Nombre completo del usuario.
 - **email**: String - Correo electrónico.
 - **contraseña**: String - Contraseña encriptada.
 - **rol**: String - Rol del usuario (puede ser "profesor", "estudiante" o "administrador").
 - **Métodos:**
 - **autenticar(email: String, contraseña: String): boolean** - Valida si las credenciales del usuario son correctas.
 - **cerrarSesion(): void** - Finaliza la sesión del usuario.
 - **cambiarContraseña(nuevaContraseña: String): void** - Permite al usuario cambiar su contraseña.
2. **Administrador (hereda de Usuario)**
 - **Métodos:**
 - **gestionarUsuarios(): void** - Gestiona la creación, modificación y eliminación de usuarios.
 - **gestionarLearningPaths(): void** - Gestiona la creación y eliminación de Learning Paths globales.
 - **verEstadisticasGlobales(): Map** - Ver estadísticas generales del sistema, como el número de usuarios, cantidad de Learning Paths, etc.

3. Profesor (hereda de Usuario)

- **Atributos:**

- `learningPathsCreados`: List<LearningPath> - Lista de Learning Paths creados por el profesor.

- **Métodos:**

- `crearLearningPath(titulo: String, descripcion: String): LearningPath` - Crea un nuevo Learning Path.
- `editarLearningPath(lp: LearningPath): void` - Permite editar un Learning Path existente.
- `clonarActividad(a: Actividad): Actividad` - Clona una actividad de otro profesor y permite modificarla.
- `asignarActividad(actividad: Actividad, estudiantes: List<Estudiante>): void` - Asigna una actividad a estudiantes.
- `calificarActividad(actividad: Actividad, estudiante: Estudiante, calificacion: Calificacion): void` - Califica una actividad completada por un estudiante.
- `verProgresoEstudiantes(learningPath: LearningPath): List<Progreso>` - Visualiza el progreso de todos los estudiantes inscritos en un Learning Path.

4. Estudiante (hereda de Usuario)

- **Atributos:**

- `learningPathsInscritos`: List<LearningPath> - Lista de Learning Paths en los que está inscrito.
- `progresos`: List<Progreso> - Progreso del estudiante en cada Learning Path.

- **Métodos:**

- `inscribirseLearningPath(lp: LearningPath): void` - Inscribe al estudiante en un Learning Path.
- `completarActividad(a: Actividad): void` - Marca una actividad como completada.
- `recibirRecomendaciones(): List<LearningPath>` - Recibe recomendaciones de caminos de aprendizaje basados en su progreso e intereses.
- `visualizarHistorial(): List<Progreso>` - Visualiza el historial de actividades completadas en todos los Learning Paths.
- `dejarFeedback(feedback: Feedback, actividad: Actividad): void` - Deja un comentario o calificación sobre una actividad.

5. Learning Path

- **Atributos:**

- `id`: int - Identificador único del Learning Path.
- `titulo`: String - Título del camino de aprendizaje.
- `descripcion`: String - Breve descripción del contenido y objetivos.

- `nivel_dificultad`: String - Nivel de dificultad (básico, intermedio, avanzado).
- `duracion_minutos`: int - Duración total en minutos (calculada a partir de la duración de las actividades).
- `rating`: float - Promedio de calificaciones recibidas por los estudiantes.
- `fecha_creacion`: Date - Fecha de creación del Learning Path.
- `fecha_modificacion`: Date - Fecha de la última modificación.
- `version`: String - Número de versión.
- `actividadesObligatorias`: List<Actividad> - Lista de actividades obligatorias.
- `actividadesOpcionales`: List<Actividad> - Lista de actividades opcionales.
- **Métodos:**
 - `agregarActividad(a: Actividad): void` - Agrega una actividad al Learning Path.
 - `calcularDuracion(): int` - Calcula la duración total del Learning Path.
 - `calcularRating(): float` - Calcula el promedio de las calificaciones dadas por los estudiantes.
 - `enviarNotificacion(): void` - Envía una notificación a los estudiantes inscritos sobre actividades nuevas o pendientes.
 - `actualizarVersion(): void` - Actualiza la versión del Learning Path cuando se realizan cambios importantes.

6. Actividad

- **Atributos:**
 - `id`: int - Identificador único de la actividad.
 - `descripcion`: String - Descripción detallada de la actividad.
 - `objetivo`: String - Objetivo de aprendizaje de la actividad.
 - `nivel_dificultad`: String - Dificultad de la actividad.
 - `duracion_esperada`: int - Tiempo estimado para completar la actividad.
 - `actividad_previa_sugerida`: Actividad - Actividad recomendada como pre-requisito.
 - `fecha_limite`: Date - Fecha límite sugerida para completar la actividad.
 - `estado`: String - Estado de la actividad (completada, pendiente, en progreso).
 - `resultado`: String - Resultado de la actividad (éxito, fracaso, etc.).
 - `version`: String - Número de versión de la actividad.
- **Métodos:**
 - `completar(): boolean` - Marca la actividad como completada.
 - `enviarRecordatorio(): void` - Envía un recordatorio al estudiante sobre la actividad pendiente.

7. Quiz (hereda de Actividad)

- **Atributos:**
 - `preguntas`: List<Pregunta> - Lista de preguntas del quiz.
 - `calificacion_minima`: float - Calificación mínima para aprobar el quiz.
- **Métodos:**
 - `calificarRespuestas(respuestas: List<String>): float` - Calcula la calificación del estudiante según sus respuestas.
 - `retroalimentacion(): String` - Devuelve una retroalimentación sobre las respuestas correctas e incorrectas.

8. Tarea (hereda de Actividad)

- **Atributos:**
 - `instrucciones`: String - Instrucciones para completar la tarea.
 - `estado`: String - Estado de la tarea (enviada, pendiente, etc.).
- **Métodos:**
 - `marcarEnviada(): void` - Marca la tarea como enviada.
 - `retroalimentacion(): String` - Proporciona feedback sobre la tarea entregada.

9. Recurso (hereda de Actividad)

- **Atributos:**
 - `tipo`: String - Tipo de recurso (video, libro, sitio web, etc.).
 - `fuelle`: String - Fuente o URL del recurso.
- **Métodos:**
 - `completarRecurso(): boolean` - Marca el recurso como completado por el estudiante.

10. Encuesta (hereda de Actividad)

- **Atributos:**
 - `preguntas`: List<PreguntaAbierta> - Lista de preguntas abiertas.
- **Métodos:**
 - `enviarRespuestas(): void` - Envía las respuestas de la encuesta.

11. Progreso

- **Atributos:**
 - `id`: int - Identificador único del progreso.
 - `learning_path_id`: int - Referencia al Learning Path en el que está inscrito el estudiante.
 - `estudiante_id`: int - Referencia al estudiante.
 - `progreso`: float - Porcentaje completado del Learning Path.
 - `fecha_inicio`: Date - Fecha en la que el estudiante comenzó el Learning Path.
 - `fecha_completado`: Date - Fecha en la que el estudiante completó el Learning Path (si corresponde).
 - `tiempo_dedicado`: int - Tiempo total dedicado al Learning Path.
 - `estado`: String - Estado del progreso (en progreso, completado, etc.).
- **Métodos:**

- `calcularProgreso(): float` - Calcula el porcentaje de actividades completadas.
- `actualizarEstado(): void` - Actualiza el estado del progreso (completado, en progreso, etc.).
- `verHistorialActividades(): List<Actividad>` - Muestra el historial de actividades completadas.

12. HistorialActividad

○ Atributos:

- `actividad_id: int` - Identificador de la actividad.
- `estado: String` - Estado de la actividad (completada, en progreso, etc.).
- `fecha_completada: Date` - Fecha en la que se completó la actividad.
- `resultado: String` - Resultado de la actividad.

13. Pregunta

○ Atributos:

- `texto: String` - Texto de la pregunta.
- `opciones: List<String>` - Opciones de respuesta.
- `respuestaCorrecta: String` - Respuesta correcta.
- `explicacion: String` - Explicación de la respuesta correcta.

14. PreguntaAbierta

○ Atributos:

- `texto: String` - Texto de la pregunta.
- `respuesta: String` - Respuesta proporcionada por el estudiante.

15. Calificacion

○ Atributos:

- `nota: float` - Calificación obtenida.
- `feedback: String` - Comentarios sobre la calificación.

16. Feedback

○ Atributos:

- `comentario: String` - Comentario dejado por el estudiante.
- `rating: float` - Calificación en forma de estrellas o número.

○ Métodos:

- `especificoActividad(): boolean` - Verifica si el feedback está relacionado con una actividad específica.
- `especificoLearningPath(): boolean` - Verifica si el feedback está relacionado con todo el Learning Path.

c. Relaciones entre las entidades, incluyendo asociaciones y herencia

1. Relaciones de herencia:

- `Administrador`, `Profesor`, y `Estudiante` **heredan** de `Usuario`.
- `Quiz`, `Tarea`, `Recurso`, y `Encuesta` **heredan** de `Actividad`.

2. Asociaciones:

- **Usuario - Learning Path:**
 - Un profesor puede crear y gestionar múltiples **LearningPath**.
 - Un estudiante puede inscribirse en varios **LearningPath**.
- **Learning Path - Actividades:**
 - Un **LearningPath** está compuesto por una secuencia de **Actividad**. Estas actividades se dividen en obligatorias y opcionales.
- **Estudiante - Progreso:**
 - Un estudiante tiene un **Progreso** asociado a cada **LearningPath** en el que está inscrito.
- **Actividad - HistorialActividad:**
 - Cada **Actividad** completada por un estudiante genera un registro en **HistorialActividad**, donde se almacena el estado, fecha y resultado.
- **Quiz - Pregunta:**
 - Un **Quiz** tiene una lista de preguntas de opción múltiple (**Pregunta**).
- **Encuesta - PreguntaAbierta:**
 - Una **Encuesta** tiene una lista de preguntas abiertas (**PreguntaAbierta**).
- **Estudiante - Feedback:**
 - Los estudiantes pueden dejar comentarios y calificaciones (**Feedback**) sobre las actividades o los Learning Paths.
- **Profesor - Feedback:**
 - Los profesores pueden ver y gestionar el **Feedback** recibido.

II. Descripción con las restricciones del proyecto

1. Persistencia de datos:

- Toda la información debe almacenarse en archivos (planos o binarios) dentro de una carpeta externa al código fuente. No se permite utilizar bases de datos, y los archivos deben estructurarse de forma organizada para facilitar la lectura y escritura.
- Los archivos pueden estar separados por entidad, como usuarios, actividades, caminos de aprendizaje, y progreso de los estudiantes. Se debe garantizar que solo la aplicación pueda leer y modificar estos archivos.

2. Autenticación obligatoria:

- Todos los usuarios deben tener un login y una contraseña. Solo los usuarios autenticados pueden acceder a las funcionalidades del sistema.
- Los usuarios se dividen en tres roles: administrador, profesor y estudiante. Cada rol tiene acceso a diferentes funcionalidades:
 - **Administrador:** Gestión de usuarios y caminos de aprendizaje.
 - **Profesor:** Creación y gestión de caminos de aprendizaje y actividades.
 - **Estudiante:** Inscripción y participación en los caminos de aprendizaje, con posibilidad de dejar feedback.

3. Gestión de versiones:

- Tanto los Learning Paths como las actividades deben manejar versiones, de forma que cuando se edite un camino de aprendizaje o actividad, se actualice su número de versión.
- Las actividades pueden ser clonadas por otros profesores, pero siempre deben mantener la versión original.

4. Actividades opcionales y obligatorias:

- Los caminos de aprendizaje deben tener actividades obligatorias y opcionales. El progreso del estudiante se calculará con base en la finalización de actividades obligatorias, aunque las actividades opcionales pueden proporcionar puntos adicionales.

5. Sistema de notificaciones:

- El sistema debe enviar notificaciones y recordatorios sobre actividades pendientes y fechas límite, de manera que los estudiantes estén informados sobre las próximas tareas que deben completar.

6. Escalabilidad futura:

- El sistema debe permitir en el futuro la integración con plataformas de gestión de aprendizaje (LMS) para automatizar procesos como la entrega de tareas y quizzes.
- Las actividades pueden ampliarse en el futuro para incluir nuevos tipos.

7. Ejecución de programas de prueba:

- Se debe implementar un conjunto de pruebas unitarias para garantizar que las funcionalidades de cada clase funcionan correctamente.

III. Descripción de lo que demostrará cada uno de los programas de prueba

Los programas de prueba verificarán la correcta funcionalidad de los componentes clave del sistema. A continuación, se detallan las pruebas que se realizarán para cada parte del sistema:

1. Pruebas de autenticación (Usuario):

- **Propósito:** Verificar que el sistema permite el inicio de sesión solo para usuarios autenticados y que el cierre de sesión funciona correctamente.
- **Prueba:** Crear varios usuarios (profesor, estudiante, administrador) e intentar iniciar sesión con credenciales correctas e incorrectas. Comprobar que solo los usuarios con credenciales válidas pueden acceder al sistema.

2. Pruebas de gestión de caminos de aprendizaje (LearningPath):

- **Propósito:** Verificar que los profesores pueden crear, editar y clonar caminos de aprendizaje, y que los estudiantes pueden inscribirse y visualizar sus caminos de aprendizaje.
- **Prueba:** Crear un nuevo camino de aprendizaje, agregar actividades obligatorias y opcionales, editar el título y la descripción, y verificar que se actualiza correctamente. Los estudiantes se inscribirán en el camino de aprendizaje y el sistema debe calcular correctamente su progreso basado en actividades completadas.

3. Pruebas de actividades (Actividad):

- **Propósito:** Asegurar que las actividades (quiz, tarea, recurso, encuesta) funcionan correctamente, calculando el resultado y estado de las actividades.
 - **Prueba:** Crear un quiz con preguntas, permitir que un estudiante lo complete y verificar que la calificación se calcula correctamente. Lo mismo se hará con tareas y encuestas, comprobando que los profesores pueden calificar y proporcionar retroalimentación.
4. **Pruebas de progreso de estudiantes (Progreso):**
- **Propósito:** Comprobar que el sistema rastrea el progreso del estudiante en base a actividades obligatorias y opcionales, y que puede mostrar el historial de actividades completadas.
 - **Prueba:** Registrar el progreso de un estudiante en un camino de aprendizaje, completar varias actividades y verificar que el progreso se actualiza correctamente. Revisar el historial de actividades completadas.
5. **Pruebas de retroalimentación (Feedback):**
- **Propósito:** Verificar que los estudiantes pueden dejar comentarios y calificaciones sobre las actividades o los caminos de aprendizaje.
 - **Prueba:** Completar un camino de aprendizaje y permitir que el estudiante deje comentarios sobre una actividad específica y sobre el camino de aprendizaje completo. Verificar que los comentarios y calificaciones se almacenan y son visibles para los profesores.
6. **Pruebas de persistencia (Archivos de datos):**
- **Propósito:** Verificar que los datos de usuarios, caminos de aprendizaje, actividades, y progreso se almacenan correctamente en archivos externos.
 - **Prueba:** Crear archivos planos o binarios para cada entidad, ejecutar varias operaciones (crear usuarios, actividades, caminos de aprendizaje, etc.), y luego verificar que los datos se guardan y pueden leerse correctamente al reiniciar la aplicación.
7. **Pruebas de notificaciones y recordatorios:**
- **Propósito:** Asegurar que el sistema envíe notificaciones sobre actividades pendientes y fechas límite.
 - **Prueba:** Crear actividades con fechas límite y verificar que el sistema envía recordatorios y notificaciones a los estudiantes inscritos.