

## Historias de usuario:

### Usuario

#### Crear un usuario

##### Historia

Como usuario del sistema, me gustaría tener la capacidad de crear una cuenta. Esto me permitirá acceder a todas las funcionalidades del sistema, personalizar mi perfil y gestionar mis interacciones dentro del entorno de aprendizaje. La creación de una cuenta debe ser un proceso sencillo, solicitando solo un usuario y contraseña para este fin.

##### Métodos usados:

- crearUsuario()

##### Entradas esperadas:

- Datos requeridos para la autenticación de la cuenta.

##### Salidas esperadas:

- Un usuario nuevo registrado en el sistema.
- Confirmación de la creación exitosa de la cuenta.

##### Criterios de aceptación:

- El proceso de creación de cuenta debe ser claro y fácil de seguir para el usuario.
- El usuario debe recibir una notificación de que su cuenta fue creada exitosamente.

### Iniciar Sesión

##### Historia

**Como usuario del sistema,** me gustaría tener la capacidad de iniciar sesión con mi correo

electrónico y contraseña, para acceder a todas las funcionalidades del sistema que me permitan interactuar con los learning paths. Iniciar sesión debe ser un proceso sencillo y seguro para garantizar que solo los usuarios registrados puedan acceder a su cuenta.

Métodos usados:

- iniciarSesion()

Entradas esperadas:

- Nombre de usuario
- Contraseña del usuario.

Salidas esperadas:

Criterios de aceptación:

- El sistema debe autenticar correctamente al usuario usando su nombre de usuario y contraseña.
- El usuario debe ser redirigido a su página de inicio o perfil personal después de iniciar sesión correctamente.
- Si las credenciales son incorrectas, el sistema debe mostrar un mensaje claro de error y permitir al usuario intentar nuevamente.

## Profesor: en rol de creador de Learning Paths

### 1. Crear un learning path

Historia:

Como profesor, me gustaría tener la capacidad de crear nuevos learning paths para proporcionar a mis estudiantes una estructura clara y organizada de aprendizaje.

Quiero que cada learning path **incluya**:

- Actividades que motiven a los estudiantes,
- El nombre del tema a estudiar
- Objetivos específicos de aprendizaje,
- La duración estimada del learning path a partir de las actividades planteadas.

También considero importante poder definir la duración estimada para completar el learning path, asignarle un nivel de dificultad para clasificar su complejidad, y

establecer un sistema de calificación mediante ratings y reseñas para medir su efectividad.

Finalmente, me gustaría agregar palabras clave a mi learning path con el objetivo de que los estudiantes sepan que esperar del learning path

#### Métodos usados

Los métodos por los que tendrá que pasar el profesor para cumplir con el requerimiento funcional son

- crearUsuario (Si no se ha creado un usuario)
- iniciarSesion
- mostrarConsolaProfesorLP
- crearLearningPath (Consola)
- crearLearningPath (Lógica)

#### Entradas esperadas

- Usuario y contraseña del profesor
- Título
- Descripción
- Objetivos del Learning Path
- Nivel de dificultad
- Palabras clave del learning path

#### Salidas esperadas

- Un Learning Path con los elementos especificados

#### Criterios de aceptación:

- El Learning Path se debe encontrar en los catálogos ubicados en el manejador de Learning Paths.
- El learning path debe encontrarse dentro de los learning paths creados por el profesor
- El learning path debe ser clonable.

## 2. Clonar learning path

#### Historia:

Como profesor, **quiero poder clonar learning paths previamente creados por otros profesores para usarlos como plantillas en la creación de nuevos.**

Esto me permitirá ahorrar tiempo, ya que podré modificar y personalizar los datos y actividades de la copia según mis necesidades específicas, como reemplazar actividades, actualizar objetivos y ajustar el contenido para adaptarlo a un nuevo grupo de estudiantes o tema.

Métodos usados:

- crearUsuario (Si no se ha creado un usuario)
- iniciarSesion
- mostrarConsolaProfesor
- clonarLearningPath

Entradas esperadas

- Un learning path previamente creado.
- El profesor que desea clonar el learning path.

Salidas esperadas

- Un nuevo learning path, el cual tendrá como profesor creador a aquel que deseó clonarlo.

Criterios de aceptación

- El nuevo learning path debe contener todos los elementos del original (actividades, objetivos, fechas etc.), pero debe ser editable para que el profesor pueda personalizarlo según sus necesidades.
- El learning path del profesor anterior debe quedar intacto a la hora de que el nuevo profesor haga cambios.

### 3. Crear actividad

Historia:

Como profesor, deseo crear actividades nuevas y dinámicas para un learning path que haya creado previamente. Cada actividad debe incluir información específica como el tipo (tarea, quiz, examen, encuesta, etc).

Toda actividad tendrá:

- Título
- Descripción
- Objetivo
- Duración esperada

- Obligatoriedad
- Fecha límite

La información que se tendrá dependerá del tipo de actividad:

- Quiz verdadero/falso:
  - Enunciado de la pregunta
  - Opción verdadera
  - Opción falsa
- Quiz Multiple:
  - Enunciado de la pregunta
  - 4 opciones
  - Opción correcta.
- Encuesta/Examen
  - Enunciado de la pregunta
  - Campo de respuestas
- Tarea
  - Secciones de la tarea.
- Recurso educativo
  - Tipo de recurso
  - Link al recurso

Esto me permitirá diseñar actividades que se alineen con los objetivos de aprendizaje del curso, proporcionando experiencias personalizadas para los estudiantes.

Métodos usados:

`iniciarSesion()`

`consolaProfesor()`

`crearEncuesta/crearTarea/crearQuiz/crearExamen/crearRecurso (Consola)`

`crearEncuesta/crearTarea/crearQuiz/crearExamen/crearRecurso (Lógica)`

Entradas esperadas:

El learning path al que se va a agregar la nueva actividad.

Los datos de la actividad: tipo, título, descripción, objetivo, nivel de dificultad, duración esperada, obligatoriedad, fecha límite, entre otros.

Salidas esperadas:

La actividad agregada correctamente a la lista de actividades del learning path seleccionado.

Criterios de aceptación:

- La nueva actividad debe incluir todos los campos requeridos: tipo, título, descripción, objetivo, nivel de dificultad, duración esperada, obligatoriedad, fecha límite, materiales didácticos y porcentaje de contribución.
- La actividad debe ser visible dentro del learning path, con la posibilidad de ser editada posteriormente si es necesario.

#### 4. Edición de un learning path

Historia

Como profesor, me gustaría tener la flexibilidad de editar completamente un learning path que haya creado anteriormente. Esto incluye la posibilidad de modificar su título, descripción, objetivos de aprendizaje, nivel de dificultad, duración estimada, actividades asociadas y cualquier otro atributo.

Me interesa que los cambios realizados no afecten negativamente a los estudiantes que ya están inscritos en el learning path.

De esta manera, puedo ajustar el contenido y la estructura del learning path sin interferir con el progreso de los estudiantes actuales.

Métodos usados:

- `iniciarSesion()`
- `consolaProfesor()`
- `modificarLearningPath()`
- `modificarLearningPath(ID: String, -título: String, -descripcion: String, -nivelDificultad: String, -objetivos: List<String>, -duracion: double, -rating: double, -numCalificaciones: int)`

Entradas esperadas:

- El ID del learning path a editar.

- Nuevos valores para los atributos del learning path: título, descripción, nivel de dificultad, objetivos, duración, entre otros.

Salidas esperadas:

- Los datos del learning path actualizados con los cambios realizados.

Criterios de aceptación:

- Los cambios deben reflejarse correctamente en el learning path editado.
- Los estudiantes previamente inscritos deben seguir accediendo al learning path sin problemas.
- El profesor debe recibir una confirmación de que los cambios se han guardado correctamente.

## Profesor: en rol de seguimiento.

### 1. Calificar actividad

Historia

Como profesor que ha creado un learning path, me gustaría poder calificar las actividades realizadas por los estudiantes inscritos. Quiero que esta funcionalidad sea versátil, permitiéndome asignar una calificación a cualquier tipo de actividad (ya sea tarea, quiz, examen o encuesta). También necesito poder agregar comentarios que ayuden al estudiante a comprender su desempeño y brindar retroalimentación constructiva para mejorar su aprendizaje. Esto me permitirá monitorear el progreso de los estudiantes de manera efectiva y ayudarlos a mejorar su rendimiento.

Métodos usados:

- `iniciarSesion()`
- `consolaPo()`
- `calificarActividadConsola()`
- `calificarActividad(Actividad actividad)`

Entradas esperadas:

- La actividad realizada por el estudiante (tarea, quiz, examen, encuesta).
- La calificación asignada.
- Comentarios de retroalimentación (opcional).

Salidas esperadas:

- El puntaje de la actividad del estudiante editado con el nuevo valor proporcionado.
- Los comentarios de retroalimentación guardados correctamente.

Criterios de aceptación:

- La calificación debe actualizarse correctamente en la actividad del estudiante.
- Los comentarios deben ser asociados correctamente con la actividad para que el estudiante pueda ver la retroalimentación.
- El profesor debe recibir una confirmación de que los cambios se han guardado correctamente.

## 2. Reseñar y calificar una actividad

Historia:

Como profesor, me gustaría tener la opción de reseñar y asignar una puntuación a las actividades propuestas en cualquier learning path.

Esta funcionalidad me permite compartir comentarios específicos sobre la calidad y utilidad de la actividad desde mi perspectiva, así como asignar un porcentaje que refleje mi nivel de satisfacción con dicha actividad. Considero importante que estas reseñas sean visibles para otros profesores, ayudándolos a evaluar si las actividades son adecuadas para sus propios learning paths.

Métodos usados:

- `iniciarSesion()`
- `consolaProfesor()`
- `reseñarActividadConsola()`
- `reseñarActividad(String comentario, Float calificacion)`

Entradas esperadas:

- El comentario de reseña sobre la actividad.
- La calificación (1-5) asignada a la actividad.

Salidas esperadas:

- La reseña y la calificación agregadas correctamente a la actividad.



#### Criterios de aceptación:

- La reseña debe ser almacenada correctamente junto con la calificación.
- La calificación debe ser en formato adecuado (porcentaje).
- La reseña debe ser visible para otros profesores para su evaluación.
- El profesor debe recibir una confirmación de que la reseña se ha guardado correctamente.

### 3. Consultar el avance de un estudiante

#### Historia:

Como profesor, me gustaría tener acceso al avance individual de cada estudiante inscrito en uno de mis learning paths. Esto incluye un porcentaje de completitud que refleje cuántas actividades ha completado, además de un desglose detallado de las actividades realizadas, calificadas y pendientes.

Esta funcionalidad me ayudará a identificar a los estudiantes que necesitan apoyo adicional y asegurarme de que estén cumpliendo con los objetivos del learning path.

#### Métodos usados:

- `iniciarSesion()`
- `consolaProfesor()`
- `consultarAvanceConsola()`
- `consultarAvance(Estudiante estudiante)`

#### Entradas esperadas:

- El estudiante cuyo avance se va a consultar.

#### Salidas esperadas:

- El avance del estudiante, con el porcentaje de completitud y el desglose de las actividades realizadas, calificadas y pendientes.

#### Criterios de aceptación:

- El porcentaje de completitud debe reflejar con precisión el progreso del estudiante en el learning path.

- El desglose de las actividades debe mostrar correctamente las actividades realizadas, las que han sido calificadas y las pendientes.

## Los Estudiantes

### Inscripción a un learning path

#### Historia:

Como estudiante, quiero poder inscribirme a cualquier learning path publicado en el catálogo general. Esto me permitirá acceder a sus actividades, recursos educativos y al sistema de calificaciones que evalúa mi desempeño. También necesito que el sistema registre automáticamente mi inscripción en una lista personalizada para facilitar el acceso a los learning paths que estoy cursando.

#### Métodos usados:

- `iniciarSesion()`
- `consolaEstudiante()`
- `inscribirLearningPathConsola()`
- `inscribirLearningPath(LearningPath learningPath)`

#### Entradas esperadas:

- La palabra clave que le interesa al estudiante
- El learning path al que el estudiante desea inscribirse.

#### Salidas esperadas:

El learning path se agrega al catálogo de learning paths inscritos del estudiante.

#### Criterios de aceptación:

- El estudiante debe ser capaz de inscribirse correctamente en el learning path deseado.
- El learning path debe ser agregado a la lista personalizada de learning paths en los que el estudiante está inscrito.
- El sistema debe mostrar un mensaje de confirmación de la inscripción exitosa.

### Visualización de avances por parte de estudiante

#### Historia:

Como estudiante, quiero poder consultar mi avance en cualquier learning path en el que esté inscrito. Esto incluye un porcentaje que indique mi progreso total, así como detalles de las actividades completadas, pendientes y calificadas.

Esta funcionalidad me ayuda a planificar mejor mi tiempo y mantenerme motivado para alcanzar los objetivos del learning path.

Métodos usados:

- `iniciarSesion()`
- `consolaEstudiante()`
- `consultarAvanceConsola()`
- `consultarAvance(LearningPath learningPath)`

Entradas esperadas:

- El learning path en el que el estudiante desea consultar su avance.

Salidas esperadas:

- El avance del estudiante en el learning path seleccionado, incluyendo el porcentaje de progreso total y detalles de las actividades completadas, pendientes y calificadas.

Criterios de aceptación:

- El porcentaje de avance debe reflejar correctamente el progreso del estudiante en el learning path.
- El desglose de actividades debe mostrar correctamente las actividades completadas, calificadas y pendientes.

## Realizar una Actividad

### Historia

Como estudiante quiero poder realizar las actividades propuestas por el profesor encargado del learning path. Por lo tanto, me gustaría poder decidir qué actividad quiero hacer y que se me pueda calificar. Los tipos de actividades están entre tareas, quizzes, exámenes y encuestas. Esto me permitirá completar los requisitos del learning path y recibir retroalimentación sobre mi desempeño.

Métodos usados:

- `iniciarSesion()`
- `consolaEstudiante()`
- `seleccionarLearningPath()`

- seleccionarActividad()
- realizarActividad(Actividad actividadBase) (Depende de la actividad seleccionada, pero siempre se realiza la actividad mediante la clase abstracta realizar(Actividad actividad))

#### Entradas esperadas:

- La actividad que el estudiante desea realizar (puede ser tarea, quiz, examen o encuesta).
- El tipo de actividad seleccionada (esto influye en cómo se registra la actividad).

#### Salidas esperadas:

- La actividad seleccionada se marca como realizada y entregada. Dependiendo del tipo de actividad, se marca como calificada o pendiente de calificación.

#### Criterios de aceptación:

- El estudiante debe poder seleccionar y realizar la actividad adecuada para el learning path.
- La actividad debe quedar registrada como entregada.
- La actividad debe ser calificada, si corresponde, o estar marcada para calificación.
- El sistema debe notificar que la actividad ha sido realizada y entregada correctamente.

## Diseño:

#### Cambios para la implementación de Quiz Verdadero/Falso

Para implementar el cambio propuesto en el learning path, que consiste en separar el quiz de verdadero/falso de un quiz de selección múltiple, se modificó la clase “Quiz” previamente definida. Esta clase fue redefinida como abstracta, lo que permitió la creación de dos clases heredadas: una para el quiz de selección múltiple y otra para el quiz de verdadero/falso, cambiando sus métodos constructores a un arreglo de 2 posiciones en el caso del verdadero/falso, y un arreglo de 4 opciones para aquel de selección múltiple.

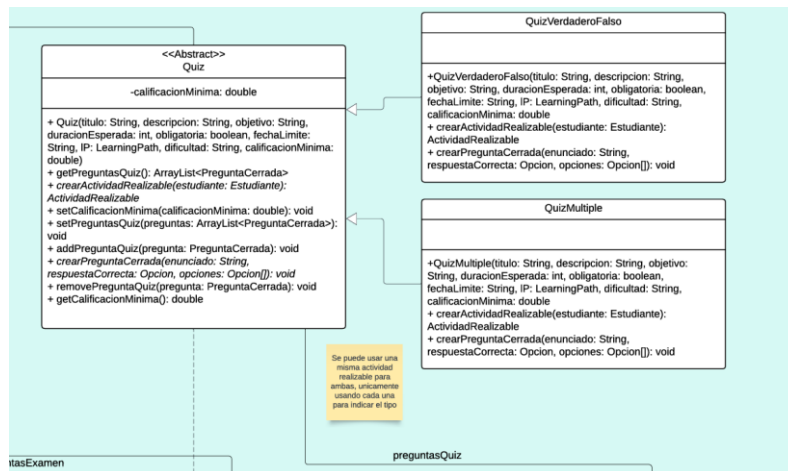


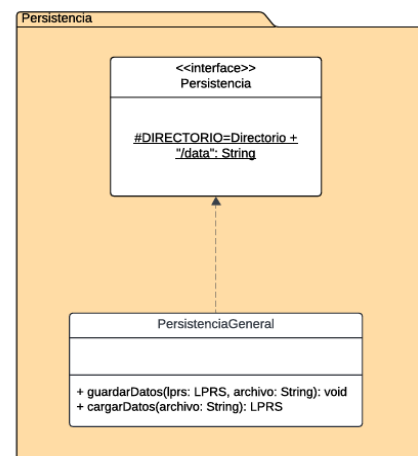
Diagrama UML: Separación de tipos de quiz.

Posteriormente, no hubo otro cambio en las clases del paquete realizable, puesto que se mantuvo la misma clase de QuizRealizable para la realización tanto del Quiz verdadero/falso como el Quiz de selección múltiple, puesto que no hubo otros cambios en las clases de PreguntaCerrada.

Finalmente, en el caso de la consola, se encontró la necesidad de realizar un único método para la creación de quiz verdadero o falso o la creación de quiz de selección múltiple, instanciando la creación de un nuevo objeto de cada tipo dependiendo de la preferencia del usuario.

### Cambios en la persistencia:

Inicialmente, se había planteado una separación en la persistencia entre los learning paths creados y los usuarios creados para tener en cuenta las necesidades de ambos manejadores, pero dado a limitaciones en la implementación, se tuvo que cambiar a una persistencia general, serializando el objeto de LPRS, siendo aquel que se guarda dentro de un archivo binario denominado “Persistencia.bin” con unicamente dos metodos, uno para guardar los datos, dando un nombre de archivo y otro para lograr cargar los datos y obtener un objeto LPRS a partir de estos.



Implementación de consola:

Para la consola, se realizaron tres métodos main para cada tipo de usuario, separados por paquetes.

#### *Consola Estudiante:*

En primer lugar, se tiene a la consola del estudiante, la cual tiene 3 subclases para dividir las responsabilidades. La primera clase de ConsolaSesionEstudiante hace las consultas al manejador de sesión dados un usuario y una contraseña.

Luego, en la clase ConsolaEstudiante, se le pide al usuario lo que desea hacer, Si esta actividad es relacionada con una actividad, se realiza dentro de la ConsolaActividadEstudiante, ya sea la realización de un Quiz, Examen, Tarea, etc.

Mientras que, si es relacionada con el avance de un estudiante dentro del learning path, se realiza en ConsolaAvanceEstudiante, ya sea ver el porcentaje de avance, tasa de éxito, entre otros aspectos.

#### *Consola Profesor – Learning Path*

En el caso del rol de profesor, se tienen dos métodos principales según sus responsabilidades. Una de las responsabilidades clave es el **rol de creador de learning paths**. En este rol, el profesor puede gestionar, lo cual incluye crear, modificar y asignar actividades o recursos a los estudiantes. Esto se gestiona principalmente desde las clases:

- **ConsolaProfesorLP:** Aquí, el profesor interactúa con el sistema para tareas generales relacionadas con el learning path, como obtener estadísticas de los estudiantes.
- **ConsolaActividadProfesor:** Esta clase permite al profesor añadir y gestionar actividades específicas, como quizzes, exámenes, tareas y recursos educativos.
- **ConsolaRecursoProfesor:** Desde aquí, el profesor puede añadir o visualizar recursos específicos asociados al learning path.

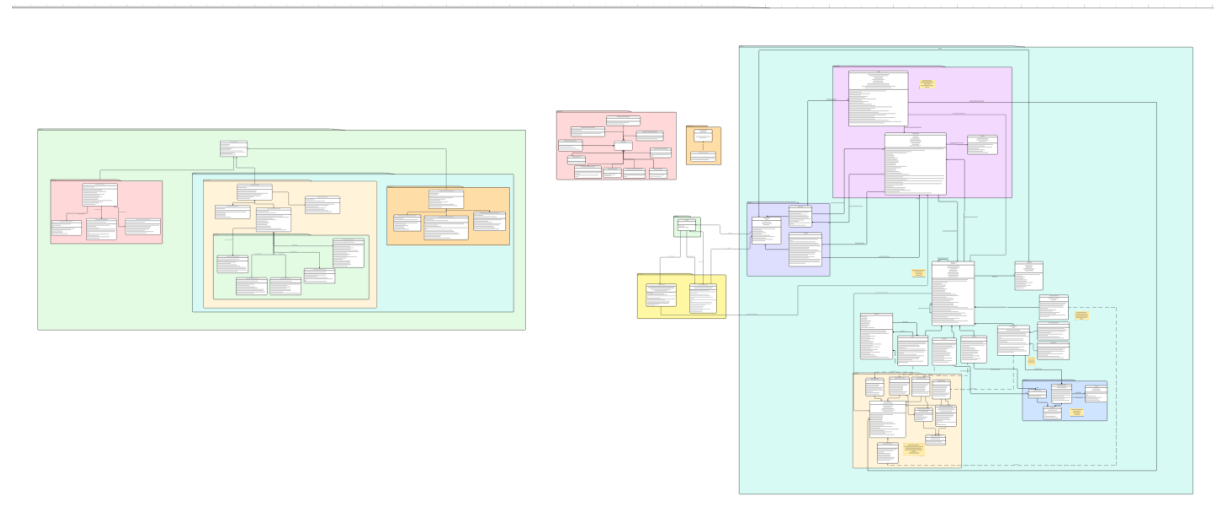
#### *Consola Profesor – Seguimiento*

El otro método principal para el profesor se enfoca en **realizar seguimiento al progreso del estudiante dentro del learning path**. Para esto, se tiene un conjunto de clases que permiten al profesor observar métricas relacionadas con el avance y éxito del estudiante:

- **ConsolaProfesorSeguimiento:** Permite al profesor acceder al seguimiento global del estudiante.
- **ConsolaActividadSeguimientoProfesor:** Focaliza las métricas relacionadas con actividades puntuales, como el tiempo invertido o los resultados obtenidos.

- **ConsolaSesionProfesorSeguimiento:** Maneja la interacción de sesión, facilitando el inicio y cierre de sesión del profesor mientras realiza seguimiento.

Diagrama UML Actualizado:



[https://lucid.app/lucidchart/75483c69-8760-4e05-9047-152f4f226fc2/edit?invitationId=inv\\_057b56bc-cc95-4d01-bd3b-a89e1e242916&page=0\\_0#](https://lucid.app/lucidchart/75483c69-8760-4e05-9047-152f4f226fc2/edit?invitationId=inv_057b56bc-cc95-4d01-bd3b-a89e1e242916&page=0_0#)