

Entrega 4:

Interfaz Grafica

Juan Sebastian Avila 202320465; Nicolas Toro; Andrea Davila 202024214

Table of Contents

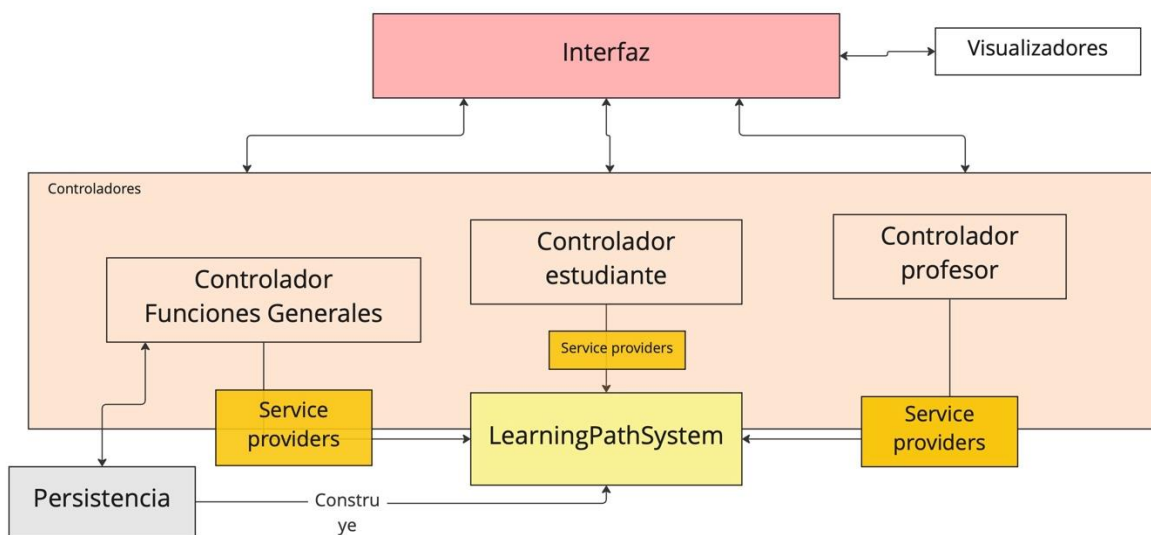
<i>Cambios en el diseño</i>	<i>2</i>
<i>Diagrama general:.....</i>	<i>2</i>
<i>Diagrama de alto nivel:</i>	<i>4</i>
<i>Diagramas de secuencias:.....</i>	<i>6</i>
Diagrama de obtención del LPS:	6
Diagrama de visualizar caminos y actividades (obtención de caminos y actividades):	6
Diagrama de creación de usuario:	6
Diagrama de autenticación de usuarios:	7
Diagrama de dejar reseñas y ratings:	7
Diagrama de creación y clonación de caminos:	8
Diagrama de creación y clonación de actividades:	8
Diagrama de edición de actividades y caminos:	9
Diagrama de inscripción en caminos:	9
Diagrama de iniciar de actividades:	9
Diagrama de calificar y marcar como exitosos o enviados envíos:	10
Diagrama de consultar avances estudiantes:	10
<i>Historias de Usuario</i>	<i>12</i>
<i>Diagrama de clases detallado:.....</i>	<i>19</i>

Cambios en el diseño

Ahora con la inclusión de las interfaces realizadas con Swing, las cuales se centran en 3 partes principales, el inicio el cual es llamado LoginPanel y es donde los usuarios de la aplicación deben ingresar y registrarse si es que no lo han hecho, también los dos paneles principales, la de profesor y la de estudiante. Se ven más adelante algunos paneles extra para el funcionamiento de estas y como fue un diseño preliminar.

Diagrama general:

(Esto no es un diagrama UML, es solamente un diagrama para entender el funcionamiento general de la aplicación. No es de los diagramas requeridos, fue uno adicional que hicimos para propósitos de aclaraciones)



Este diagrama muestra la idea principal detrás del diseño de nuestro programa. Aunque no fue implementada la interfaz en esta fase del proyecto, la idea es que esta se comuniquen con los controladores de una manera similar en la que el main se comunica en este momento. Estos controladores a su vez se comunican con el LearningPathSystem (LPS), que sirve como el information holder principal de las demás clases al contener todos los caminos y usuarios. Pusimos el LPS como atributo del controlador de funciones generales para que el LPS sea creado automáticamente cuando se cree el controlador general.

Para requisitos que involucren crear clases o editar varios atributos en distintas clases, los controladores llaman a las funciones estáticas de los service providers.

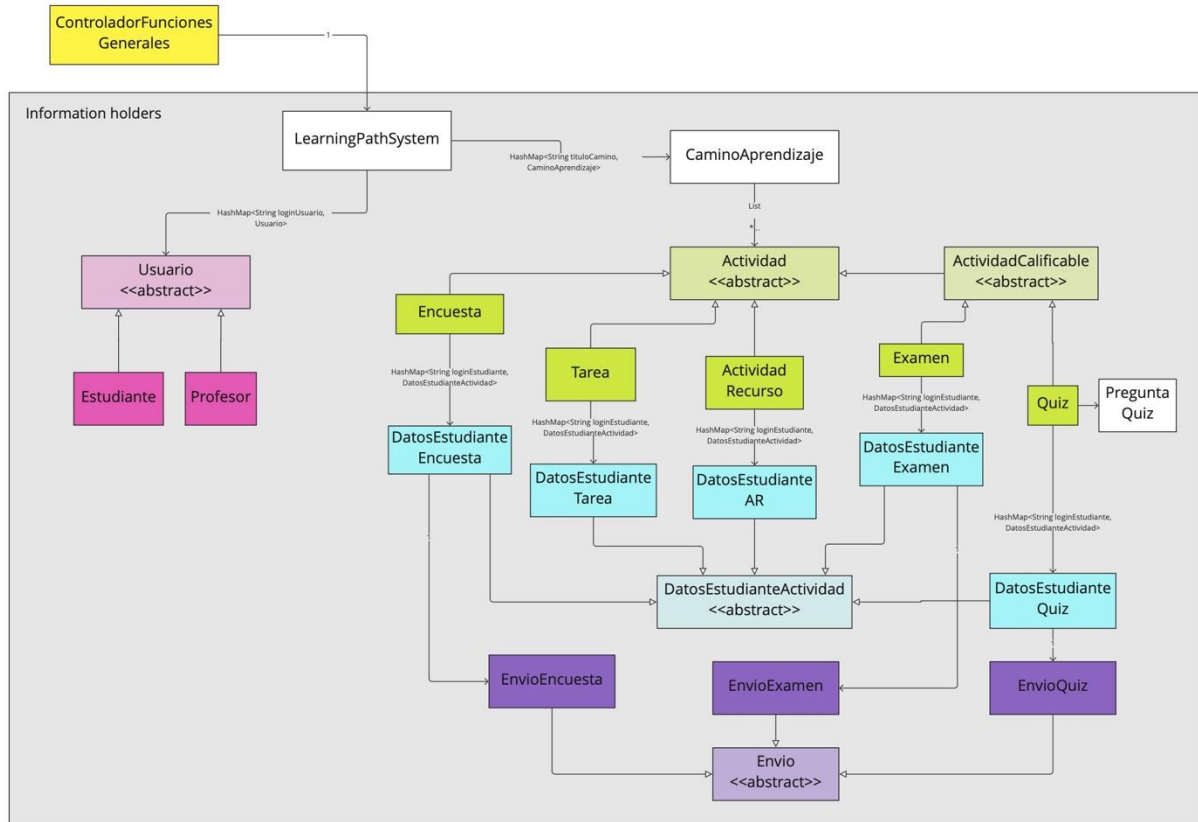
Es importante resaltar que en nuestro diseño decidimos que la transferencia de información se hiciera principalmente por medio de usar objetos en los parámetros y returns de los métodos. Por ejemplo, en vez de hacer docenas de funciones en el controlador de profesor o en los service providers para editar una actividad, la idea es que la actividad en sí sea mandada a la interfaz y desde ahí se edite directamente con los setters de la actividad (claro está que la interfaz solo va a dejar la opción para editar si se confirma previamente que el profesor es el creador de la actividad). Otro ejemplo de esta idea en práctica es la manera de conseguir un usuario individual. Si la interfaz necesita un usuario, la idea es que se comunique directamente con el método del LPS de `getUsuarioIndividual()` (por eso es que en el diagrama la mitad de la caja de LPS está en la caja de los controladores). No hicimos un método en un controlador que sirviera de intermediario para funciones como estas ya que no añadía mucho valor al programa. Somos conscientes que al seguir esta idea incrementamos las dependencias en nuestro diseño, pero consideramos que valía la pena considerando la cantidad de código que se ahorra.

La persistencia construye el LPS al comienzo de cada inicio del programa, por ende se comunica con el controlador de funciones generales. Al cerrar cada programa, el LPS con los nuevos cambios es mandado a la persistencia para que esta guarde los cambios hechos.

Por último, a pesar de que los visualizadores son guardados en el paquete de service providers en el programa, estas clases se supone que son clases auxiliares para la interfaz para imprimir la información en consola.

Diagrama de alto nivel:

(Añadimos colores en las actividades, datosEstudiante, y envios para propósitos de aclaración y facilidad de lectura del diagrama)



Controladores

ControladorEstudiante

ControladorProfesor

Service Providers

Clases y funciones estáticas

VisualizadorEnviosCalificacionesEstados

VisualizadorCaminosActividades

ActualizadorCalificacionesExitoso

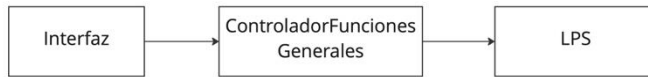
Inscriptor

CreadorCaminosActividades

CreadorUsuarios

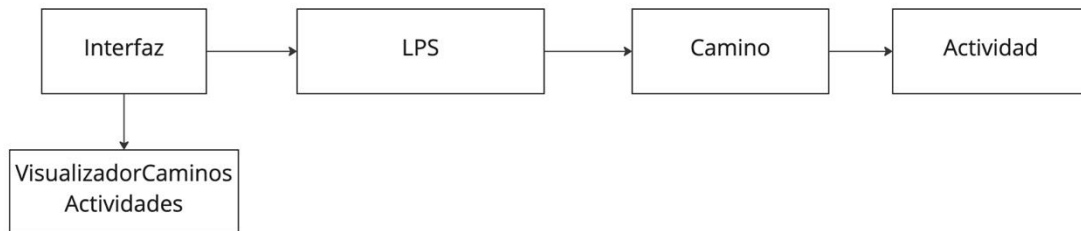
Diagramas de secuencias:

Diagrama de obtención del LPS:



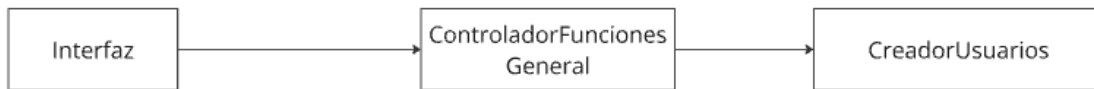
Recién se abre el programa, se crea la interfaz y el controlador de funciones generales. De este controlador la interfaz obtiene el LearningPathSystem, la cual va a tener acceso directo para facilitar la búsqueda de información.

Diagrama de visualizar caminos y actividades (obtención de caminos y actividades):



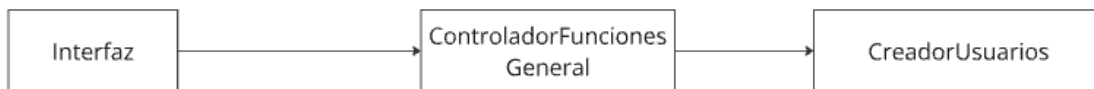
Después de haber obtenido el LearningPathSystem del controller, la interfaz se puede comunicar directamente con esta clase para acceder a un camino. El camino devuelve las actividades que tiene dentro de sí si se quiere ver una actividad en específico en vez del camino. Al ya haber obtenido los caminos o actividades necesarios, la interfaz llama a funciones del VisualizadorCaminosActividades para que sean imprimidos y mostrados en la consola del programa. Se pasa la actividad y el camino como objetos a la interfaz para facilitar la transferencia de información de los atributos de estos objetos.

Diagrama de creación de usuario:



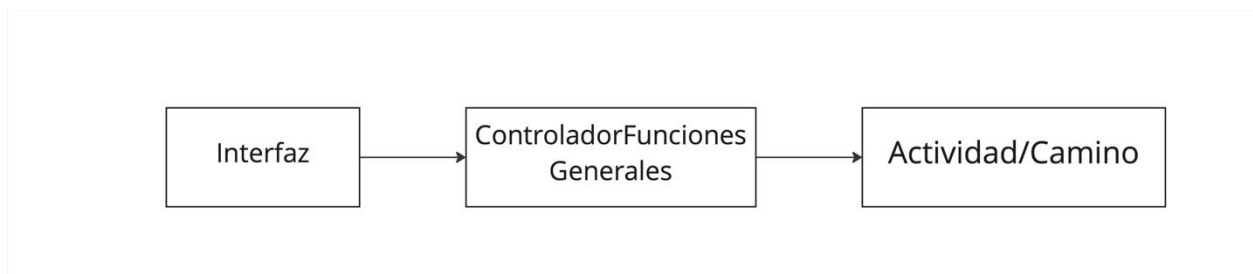
Para la creación de un usuario la interfaz obtiene un nombre de usuario, contraseña y tipo de usuario (estudiante o profesor). Luego esta información se pasa al controlador de funciones generales y este último se la pasa al creador de usuarios. Ahí se verifica que no haya un usuario con ese nombre de usuario registrado y si no lo hay, se crea un nuevo usuario con la información proporcionada. La información del usuario se guarda para ser usada después para poder ingresar como ese usuario.

Diagrama de autenticación de usuarios:



Para la autenticación de usuarios la interfaz obtiene un nombre de usuario y una contraseña. Esta información se la pasa al controlador general de funciones y este se lo pasa al creador de usuarios. Este último se asegura que la clave de un usuario con el nombre de usuario sea también la obtenida del controlador y por ende de la interfaz.

Diagrama de dejar reseñas y ratings:



Primero, la interfaz obtiene el rating/reseña que el usuario quiera añadir y la actividad. Esto se lo pasa al controlador que llama a los métodos de añadir reseñas y ratings de la actividad o camino. Decidimos de que fuera el controlador que llama al método de la actividad/camino en vez de la interfaz por si tenemos que añadir alguna comprobación o exception a esta función en el futuro. En el caso que sea así, se añadiría en el controlador en vez de la interfaz.

La decisión de pasar por un controlador fue implementada en otras funciones por la misma razón de tener facilidad de añadir exceptions en el futuro.

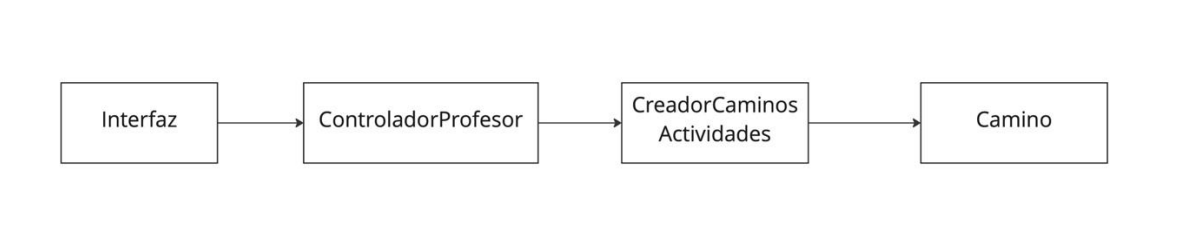
Diagrama de creación y clonación de caminos:



Para crear un camino, la interfaz primero obtiene los atributos necesarios con los cuales el usuario quiere crear el camino. Después se comunica con el controlador de profesores. Este llama a la función de creación de caminos dentro de CreadorCaminosActividades, la cual crea el camino y lo guarda en el LearningPathSystem y guarda su título en los datos del profesor.

Para el caso de la clonación se sigue la misma secuencia. Lo que cambia es que en vez de obtener todos los atributos del teclado del usuario en la interfaz, se obtiene es el camino original que se quiera clonar.

Diagrama de creación y clonación de actividades:



Para crear una actividad, la interfaz primero obtiene los atributos necesarios con los cuales el usuario quiere crear la actividad. Después se comunica con el controlador de profesores. Este llama a la función de creación de actividades dentro de CreadorCaminosActividades dependiendo del tipo de actividad que se quiera crear, la cual crea la actividad y la guarda en el camino.

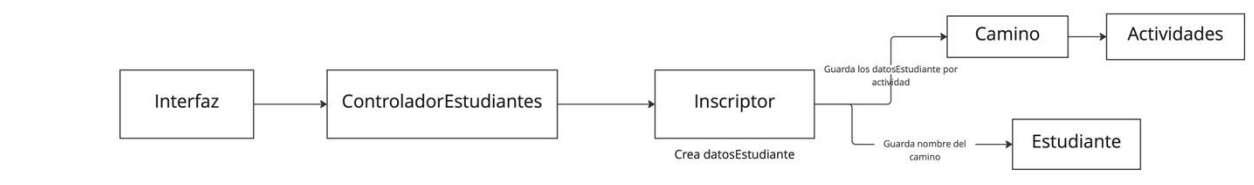
Para el caso de la clonación se sigue la misma secuencia. Lo que cambia es que en vez de obtener todos los atributos del teclado del usuario en la interfaz, se obtiene es la actividad original que se quiera clonar.

Diagrama de edición de actividades y caminos:



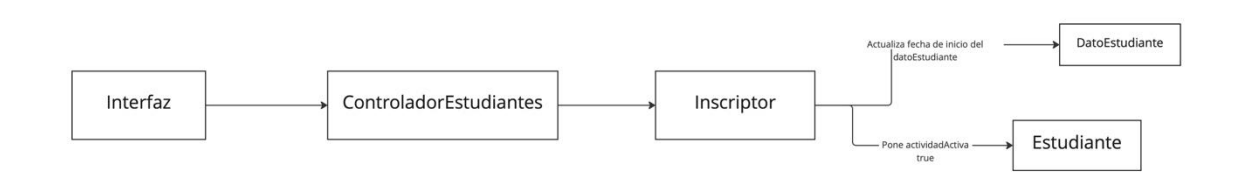
Para editar actividades o caminos, la interfaz se comunica con el LPS para obtener el camino o actividad que se desee editar. Al obtener el camino o actividad, la interfaz se comunicara directamente con los setters de estas clases para cambiar sus atributos. Sin embargo, antes de permitirle al usuario hacer esto se va a chequear que el profesor si es el creador por medio de la función estática de `isProfCreador()` que se encuentra en `CreadorCaminosActivados()`. Decidimos pasar la actividad y camino a la interfaz ya que hacer funciones en el controlador o LPS para cada setter de cada atributo hubiera sido código innecesario y desgastante.

Diagrama de inscripción en caminos:



Para inscribirse a un camino, la interfaz se comunica con el controlador de estudiantes. Este llama la función de inscripción del Inscriptor la cual crea un datosEstudiantes por cada una de las actividades del camino y lo guarda. Además, guarda el nombre del camino en el historial del estudiante.

Diagrama de iniciar de actividades:



La interfaz primero obtiene el estudiante y la actividad que este quiere empezar. Después le pasa esta información al controlador de estudiantes que chequea que el estudiante no tenga una actividad activa (si la tiene, imprime que no se puede iniciar la actividad). Si no tiene una actividad empezada, llama al inscriptor y a su función de iniciar actividad. Al iniciarla, se guarda la fecha de inicio en datoEstudiante que se encuentra en la actividad y corresponde al estudiante. Además, se pone el booleano de actividadActiva true para que no se vaya a empezar otra actividad sin haber acabado la que acaba de iniciar.

Diagrama de calificar y marcar como exitosos o enviados envíos:

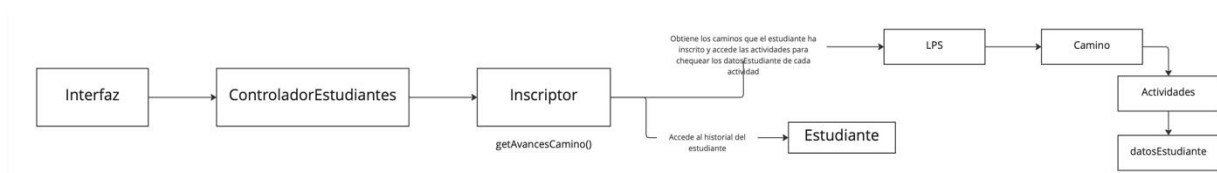


Para actividades como exámenes y tareas, en la que un profesor tiene que calificarlas, se sigue el siguiente procedimiento. Primero, la interfaz obtiene la información del profesor, actividad se quiere calificar, a que estudiante quiere calificar y que calificación y/o el estado quiere darle al estudiante en esa actividad. Luego, esta información se pasa al controlador del profesor quien se la pasa al actualizador de calificaciones. Este último cambia el estado, en el caso de tarea, o la calificación y el estado, en el caso de examen en los datos del estudiante de la actividad correspondiente.



Para actividades que requieren ser marcadas como exitosas/enviadas por el estudiante el procedimiento es el siguiente: Primero, la interfaz obtiene los datos del usuario y de la actividad que se desea marcar como exitosa o enviada. Luego esta información se le pasa al controlador del estudiante, el cual luego se la pasa al actualizador de calificaciones o exitoso. Este cambia en los datos del estudiante para la actividad correspondiente el estado pertinente (exitoso, enviado). Por ejemplo, una actividad de recurso siempre se marca como exitosa, pero una actividad tarea se marca como enviada.

Diagrama de consultar avances estudiantes:



La interfaz se comunica con el controlador de estudiantes que llama al método de `getAvancesCamino` del inscriptor. Este método accede al historial del estudiante para conseguir los títulos de los caminos que ha iniciado. Con esta información, accede a los caminos en el `LearningPathSystem` que ha inscrito el estudiante para conseguir las actividades y los `datosEstudiante` del estudiante. Al acceder esta información puede calcular el porcentaje de actividades completadas del camino, lo cual es guardado en un `hashMap` con el título del camino como llave que es devuelto a la interfaz.

Historias de Usuario

1. Historias de Usuario Generales: Profesor y Estudiante
 - a. Registro al sistema de caminos de aprendizaje

Como usuario interesado en acceder al sistema de los caminos de aprendizaje. Quiero registrarme en el sistema para poder acceder a las funcionalidades de este mismo según mis necesidades (administrar o aprender de los caminos de aprendizaje).

Entradas: Nombre de usuario, contraseña, tipo de usuario (estudiante/profesor).

Salidas: Registrarse en el sistema para poder iniciar sesión.

- b. Ingresar al sistema de caminos de aprendizaje

Como usuario interesado en acceder al sistema de los caminos de aprendizaje, quiero autenticarme en el sistema para acceder a las funcionalidades de este mismo según mis necesidades (administrar o aprender de los caminos de aprendizaje).

Entradas: Nombre de usuario, contraseña.

Salidas: Iniciar sesión para poder realizar operaciones dentro del sistema.

2. Historias de Usuario del Estudiante
 - a. Inscripción a un camino de aprendizaje.

Como estudiante interesado en aprender. Quiero inscribirme a un camino de aprendizaje para tener una guía de cómo aprender el tema de mi interés, completando diferentes actividades.

Entradas: Camino a inscribirse, estudiante a inscribir.

Salidas: Se registra al estudiante en el camino para poder hacer las actividades.

- b. Iniciar una actividad de un camino de aprendizaje en el que se está inscrito

Como estudiante, quiero iniciar una actividad de un camino de aprendizaje en el que estoy inscrito para poder entender mejor los temas que me interesan y poder avanzar en el camino.

Entradas: Usuario del estudiante, camino de aprendizaje, actividad a iniciar.

Salidas: El estudiante puede acceder para luego marcarla como enviada, exitosa.

- c. Obtener recomendaciones de caminos de aprendizaje según mis intereses

Como estudiante interesado en cursar algún camino de aprendizaje que me guste, quiero poder ver los caminos que se recomiendan, para que se me haga más fácil encontrar un camino afín a mis intereses.

Entradas: Usuario del estudiante (con sus intereses), sistema de caminos (cada camino tiene etiquetas).

Salidas: El estudiante recibe la lista de caminos que se le recomiendan.

- d. Marcar actividades como enviadas o completadas

Como estudiante, quiero marcar actividades como enviadas o completadas (exitosas), para poder iniciar otra actividad y avanzar en el progreso de mi camino de aprendizaje.

Entradas: Usuario del estudiante, camino de aprendizaje, actividad a marcar

Salidas: Se cambia el estado de la actividad y el estudiante puede acceder a la siguiente.

e. Dejar reseñas o calificaciones en actividades

Como estudiante, quiero dejar reseñas o calificaciones, para que otras personas puedan saber mi opinión sobre la actividad en cuestión.

Entradas: Usuario del estudiante, actividad.

Salidas: Se registra la reseña o calificación del estudiante en la actividad.

f. Ver el progreso del estudiante en sus caminos

Como estudiante, quiero saber cual es mi progreso en los caminos en los que estoy inscrito, para poder saber que porcentaje me falta aprobar para completar dichos caminos.

Entradas: Usuario del estudiante, sistema de caminos de aprendizaje

Salidas: Se le muestra al estudiante los caminos a los que está inscrito y sus respectivos porcentajes de cuanto a completado (según actividades exitosas).

3. Historias de Usuario del Profesor creador de caminos de aprendizaje

a. Creación de un camino de aprendizaje

Como profesor creador de caminos, quiero crear un camino de aprendizaje, para poder modelarlo según las actividades que crea pertinentes para que un estudiante entienda el tema a fondo.

Entradas: Usuario del profesor, nombre del camino, descripción, objetivos, nivel de dificultad, actividades del camino.

Salidas: Se registra el camino en el sistema de caminos de aprendizaje.

b. Creación de actividades

Como profesor creador de caminos, quiero crear actividades, para que mis estudiantes puedan entender más los temas o para evaluar su aprendizaje.

Entradas: nombre de la actividad, descripción, objetivos, nivel de dificultad, duración estimada, fecha límite (opcional), tipo de actividad (recurso, tarea, encuesta, examen o quiz) y si es obligatoria o no. Parámetros según el tipo: preguntas (con respuesta correcta, explicaciones y opciones) en el caso de quizzes, en el caso de exámenes o encuestas, preguntas abiertas, recursos para leer o ver en el caso de actividades de recurso o instrucciones para tareas.

Salidas: Se registra la actividad para luego asociarla a un camino

c. Añadir actividades a un camino de aprendizaje

Como profesor creador de caminos, quiero añadir actividades a uno de mis caminos de aprendizaje para que mis estudiantes sean capaces de acceder al contenido que les quiero proporcionar.

Entradas: Actividad, camino

Salidas: Se asocia la actividad a un camino específico.

d. Duplicar caminos de aprendizaje de otro profesor

Como profesor creador de caminos, quiero duplicar el camino de aprendizaje de otro profesor, para poder editarlo a mi gusto y no tener que empezar un camino desde cero.

Entradas: Camino a duplicar, nombre del nuevo camino

Salidas: Se duplica el camino bajo la autoría del nuevo profesor.

e. Editar caminos o actividades

Como profesor creador de caminos, quiero editar mis caminos de aprendizaje para poder tener mayor control sobre este mismo y que refleje mis nuevos deseos.

Entradas: Camino o actividad a editar, modificaciones necesarias.

Salidas: Se edita el camino o actividad y se guarda.

4. Historias de Usuario del Profesor calificador de caminos de aprendizaje

a. Asignar notas (estados) al envío de una actividad de un estudiante

Como profesor calificador de un camino, quiero acceder a los envíos de un estudiante para poder revisarlos y asignarle una nota.

Entradas: Camino, actividad, usuario del estudiante a evaluar, calificación (estado) correspondiente.

Salidas: Se registra la calificación (estado) de la actividad del estudiante.

b. Ver el progreso de los estudiantes que cursan un camino

Como profesor calificador de un camino, quiero ver el progreso de mis estudiantes en Un camino para poder evaluar su desempeño y entendimiento del material.

Entradas: Camino

Salidas: Se le muestra al profesor el desempeño de las actividades de todos los estudiantes que cursen el camino en cuestión.

Interfaz diseñada con Swing


La interfaz gráfica está diseñada para ofrecer una ex y estructurada, dividiendo claramente las diferentes funcionalidades del sistema en secciones fáciles de acceder. El flujo de navegación se organiza en paneles, cada uno enfocado en una tarea específica, y cada uno se activa mediante botones situados en un panel de botones vertical. Esta organización asegura que el usuario siempre esté consciente de dónde se encuentra dentro de la aplicación y de qué opciones tiene disponibles en cada momento. La estructura comienza con un panel de inicio que da la bienvenida al usuario, seguido de opciones claras como visualizar los Learning Paths (LPs), las actividades, agregar reseñas o inscribirse en un LP.

El panel de login es fundamental en esta interfaz, sirviendo como la puerta de entrada al sistema. A través de él, los usuarios pueden ingresar sus credenciales para autenticar su identidad. Existen dos tipos de usuarios en el sistema: estudiantes y profesores, y la interfaz se adapta a cada uno de ellos, mostrando diferentes funcionalidades según el rol. Los estudiantes pueden ver sus Learning Paths, actividades y progreso, así como agregar reseñas, mientras que los profesores tienen acceso a funciones de administración y evaluación. Además, el visualizador de actividades se organiza de forma que el estudiante pueda seleccionar el Learning Path de su interés y navegar entre las actividades de manera sencilla. Al incorporar estas distintas funcionalidades dentro de un diseño basado en paneles y botones, la interfaz proporciona un acceso claro y rápido a todas las opciones disponibles, mejorando la eficiencia y la usabilidad del sistema.

También permite fácilmente agregar nuevos botones a la lista de opciones de ambos ya sea creando nuevas clases o colocando más métodos a sus respectivas clases. En el panel de Login también es posible registrar a un usuario nuevo, con uno contraseña y login nuevos además de seleccionar un tipo de usuario, sin embargo para que sea posible tener un usuario como profesor es necesario adicionar al final de la contraseña la palabra clave “PROFESOR” la cual permite crearlo como profesor y no como estudiante, sin embargo esta palabra clave no se guarda como parte de la contraseña por lo que el profesor va a seguir siendo capaz de ingresar con la contraseña que tenía antes de colocar esta palabra clave.

A continuación, están los diferentes paneles colocados en la aplicación.




 Sistema de Inicio de Sesión — □ ×

Bienvenido

Login:

Contraseña:

 Sistema de Inicio de Sesión — □ ×

Crear Nuevo Usuario

Nuevo Usuario:

Nueva Contraseña:

Selecciona uno: ☐ Profesor ☐ Estudiante

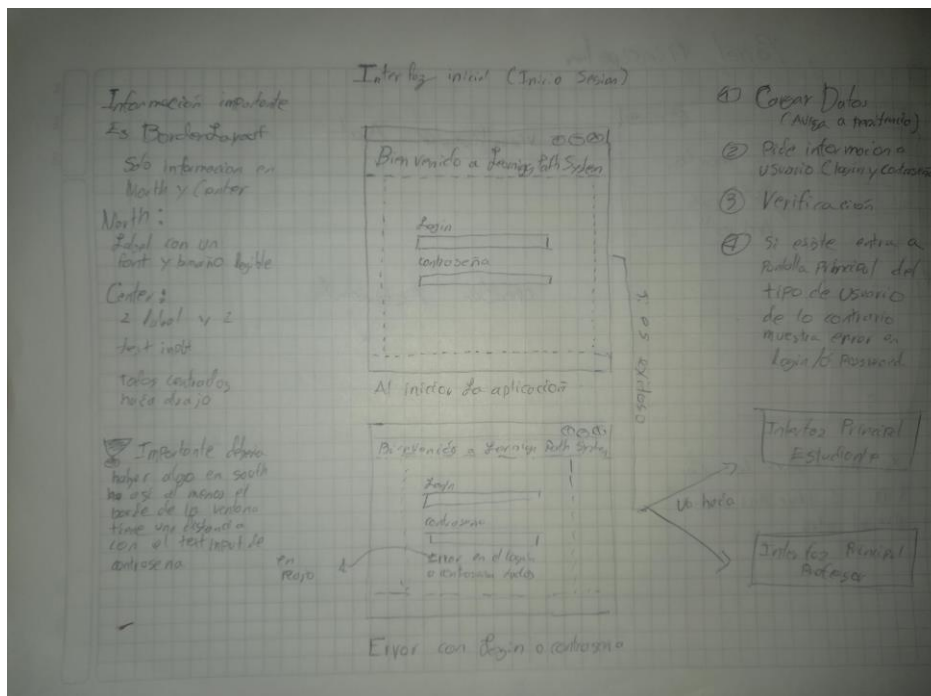
Papel Principales

Boton

Información
Relevante
"Acciones"

000

Lo mismo pero lo, Co,
tipo. 1 tiene mas acciones
que el otro



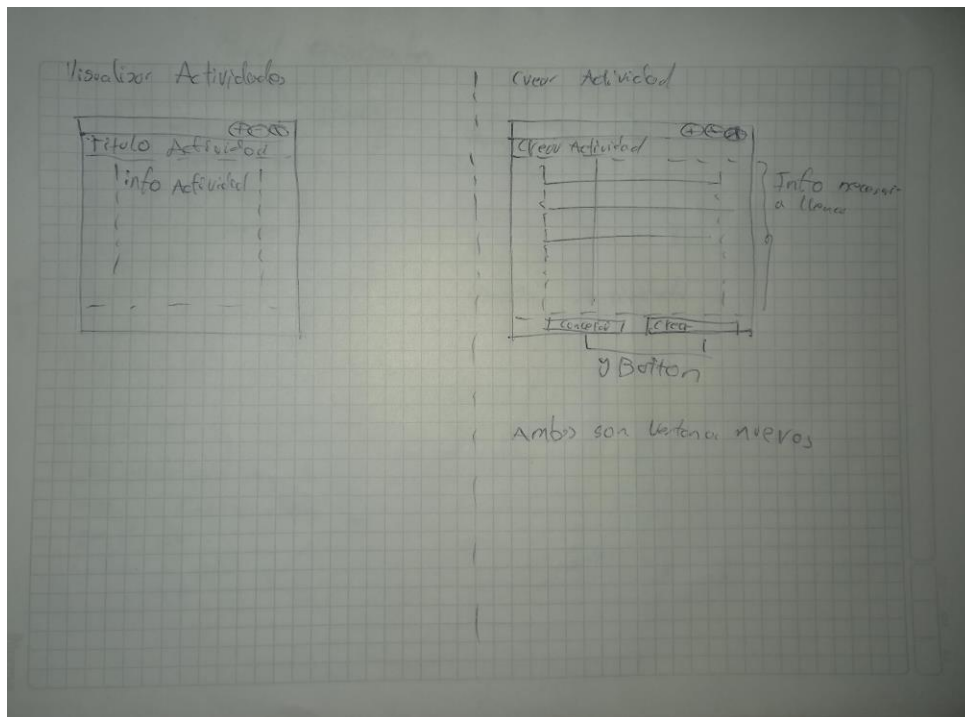
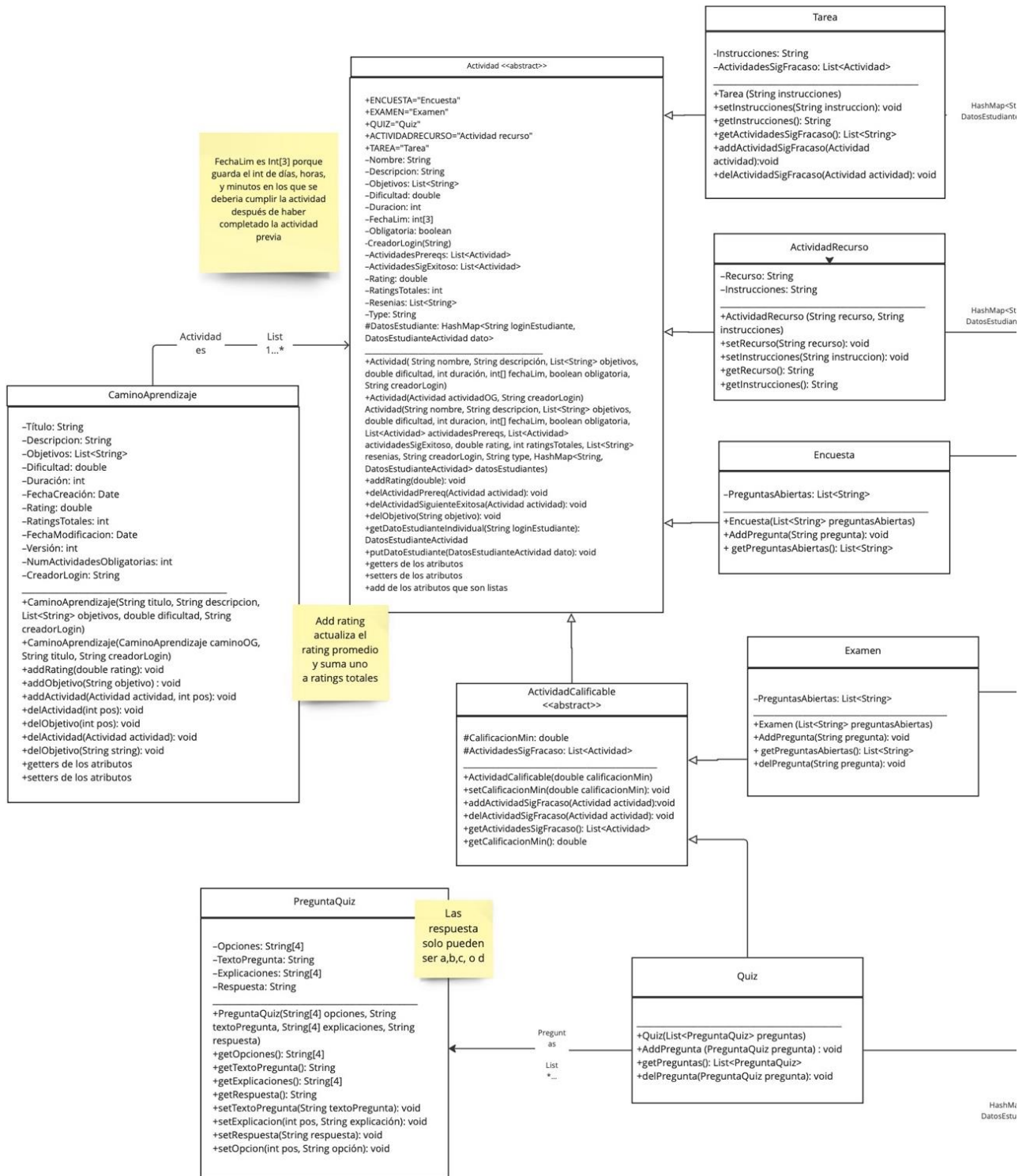
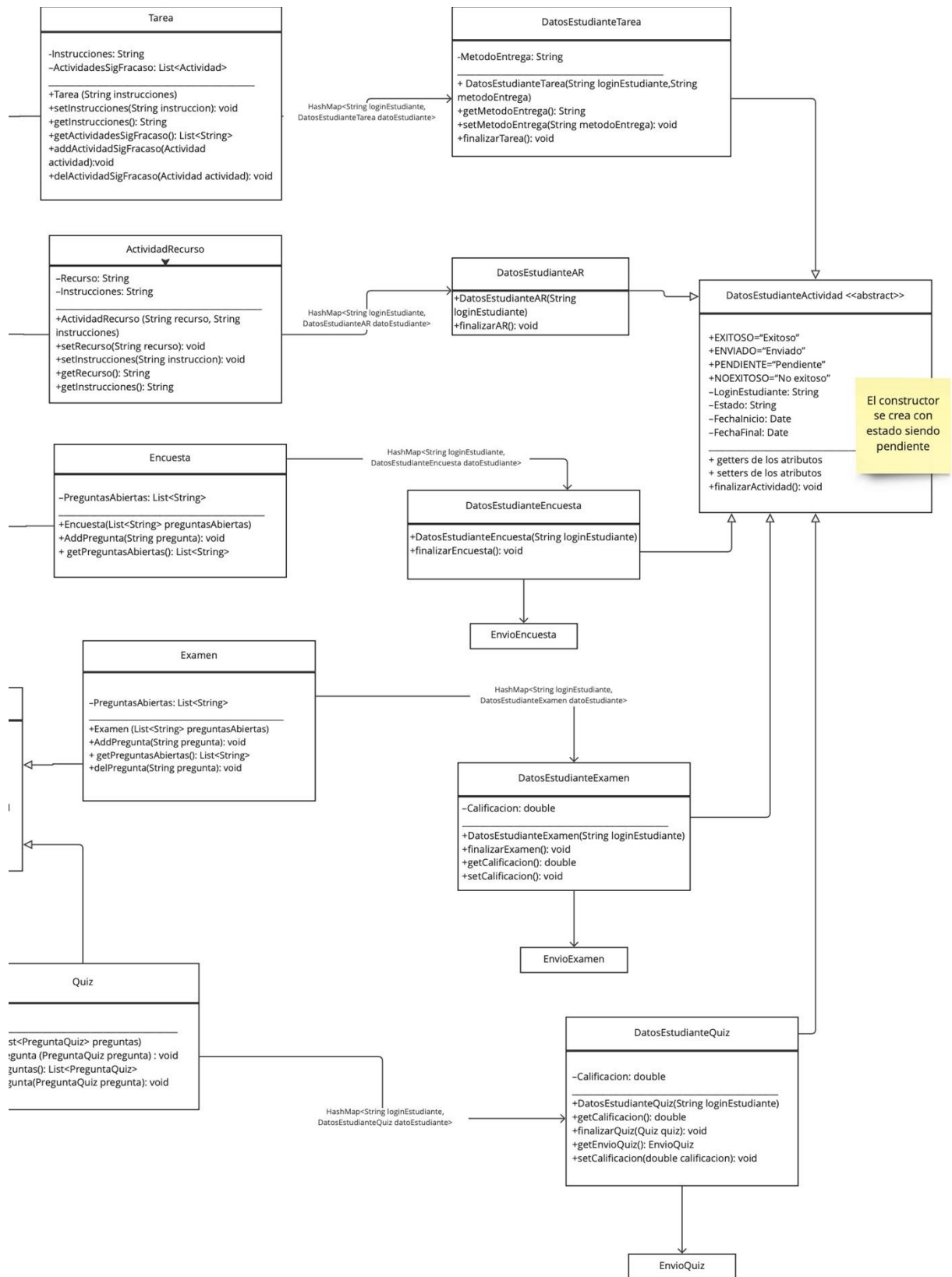
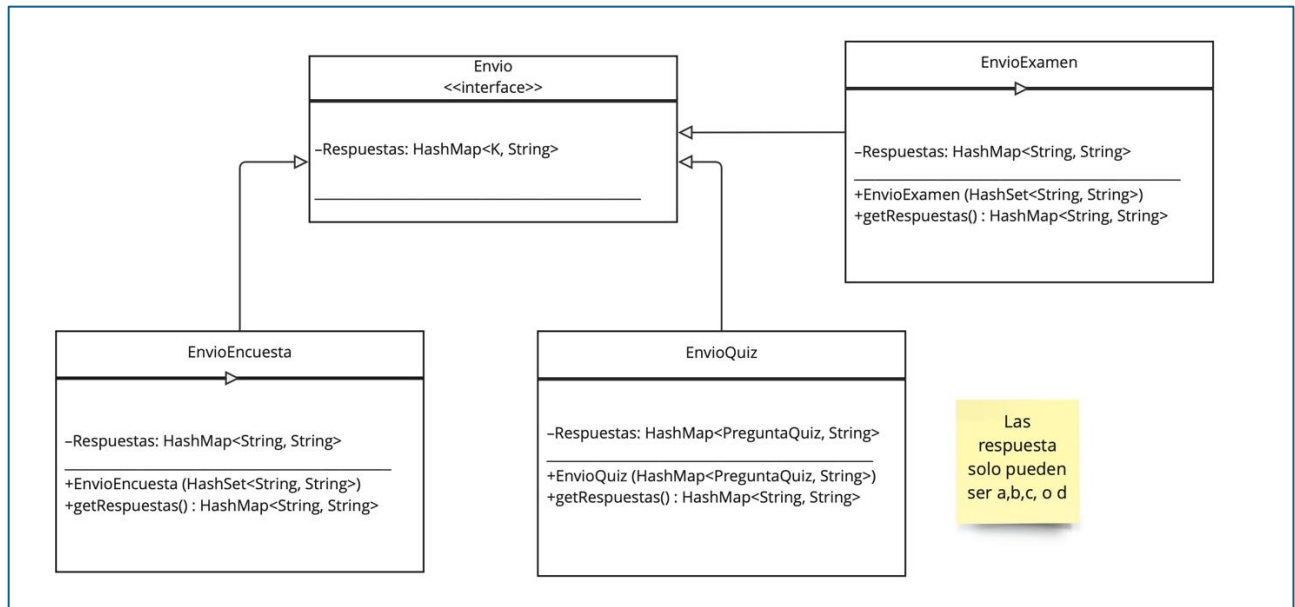


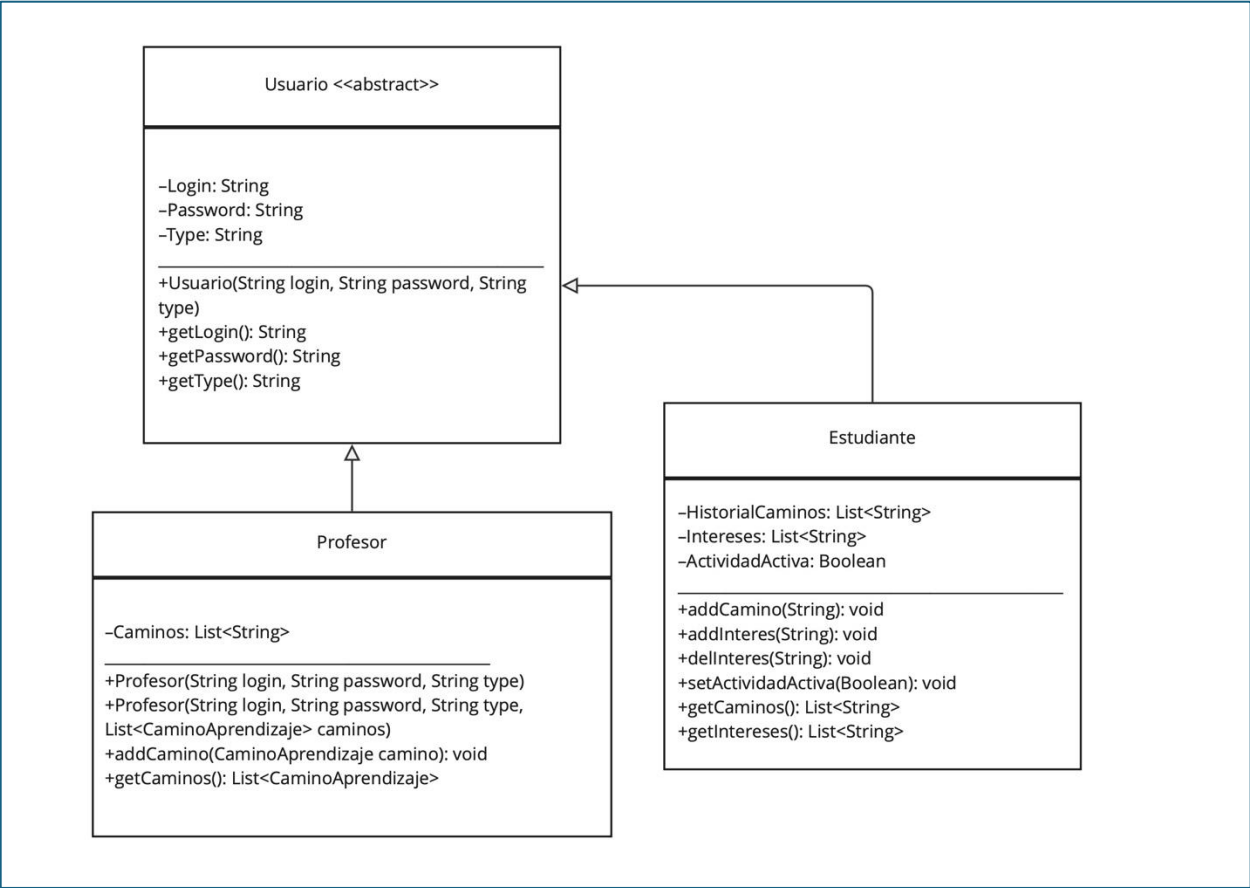
Diagrama de clases detallado:

Para que el diagrama fuer más legible, resumimos los getters y setters de ciertas clases como getters y setters de los atributos.

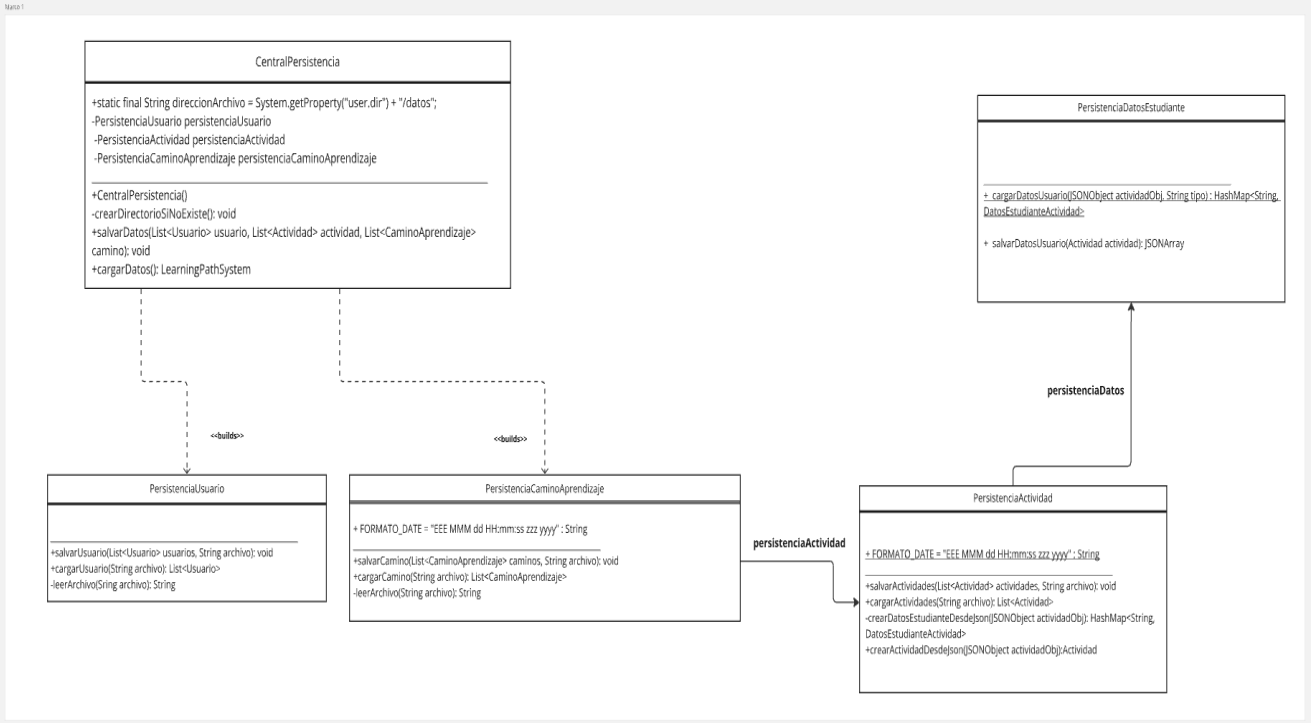


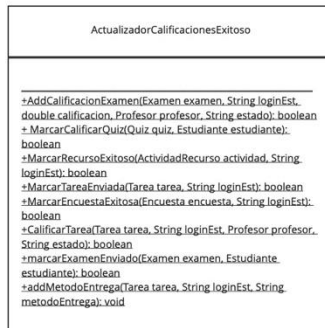
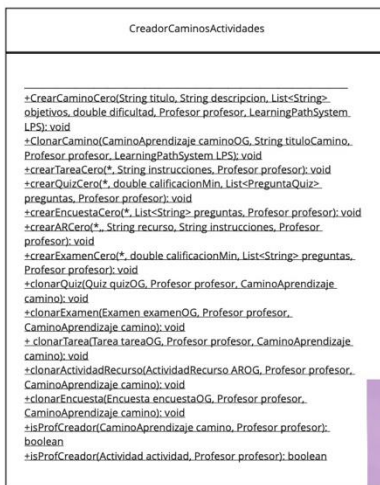
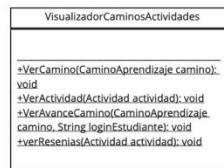
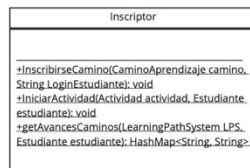
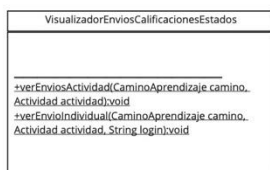
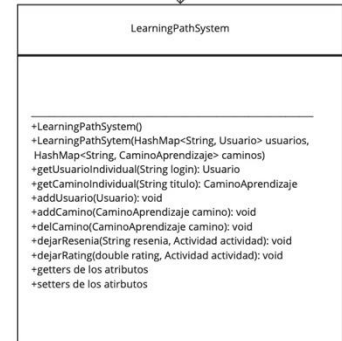
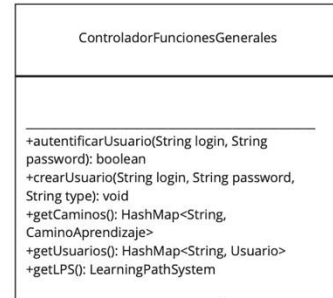
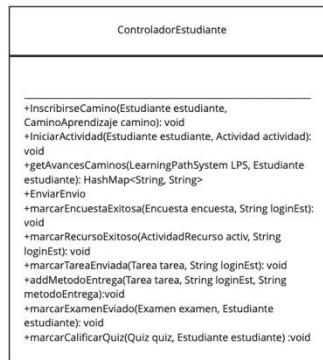
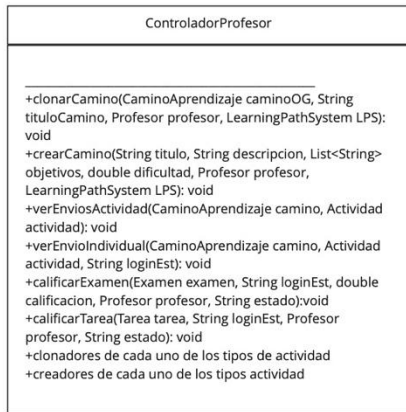






Nota 1





*=CaminoAprendizaje camino, String nombre, String descripcion, List<String> objetivos, double dificultad, int duration, int[] fechaLim, boolean obligatoria

