

Proyecto 2

**Shaiel Mateo Jimenez Posada 202323846; Daniela González
Ovalle – 202320856**

Tabla de contenido:

<i>Diagrama general:</i>	3
<i>Diagrama alto nivel:</i>	3
<i>Diagramas de secuencia:</i>	5
Diagrama de creacion de caminos:	5
Diagrama de iniciar actividad:	5
Diagrama para ver caminos:	5
Diagrama para ver actividades:	5
Diagrama para marcar como exitoso o calificar:	6
Diagrama de creacion de usuarios:.....	6
Diagrama de clonacion actividad:	6
Diagrama de creacion actividades:.....	7
Diagrama de inscripcion en caminos:	7
Diagrama de edición de actividades y caminos:	7
Diagrama de ver avances estudiante:	7
Diagrama de ver envios:	8
Diagrama de dejar reseñas y ratings:	8
<i>Diagrama de clases detallado:</i>	9
<i>Historias de usuario:</i>	16
Profesor como creador parte 1:	16
Profesor como creador parte 2:	19
Profesor como creador parte 3:	19
Estudiante:	19
Profesor como calificador:	20

Diagrama general:

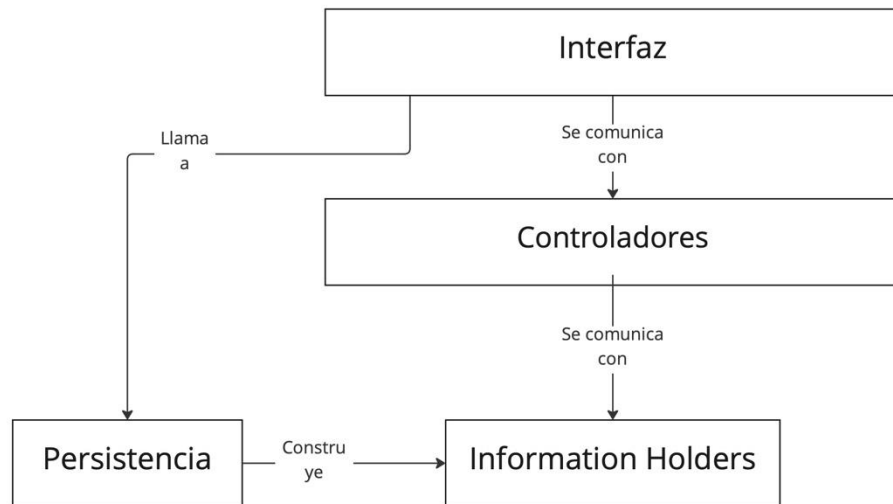
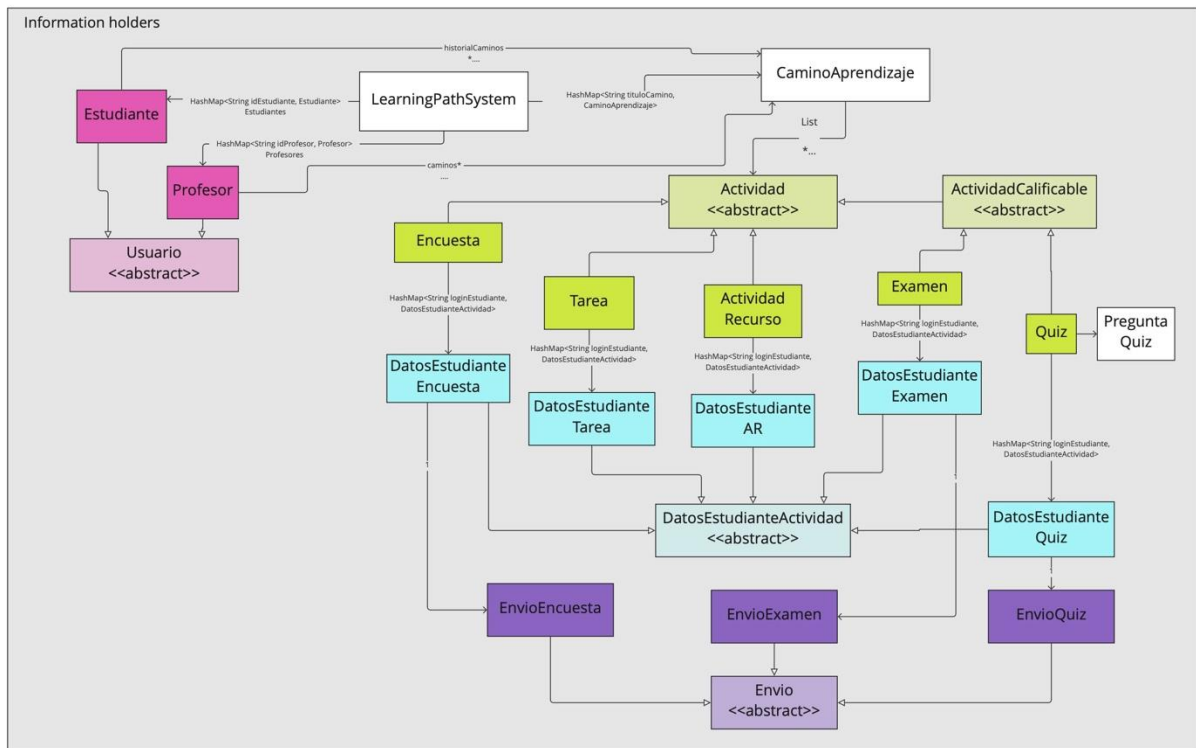
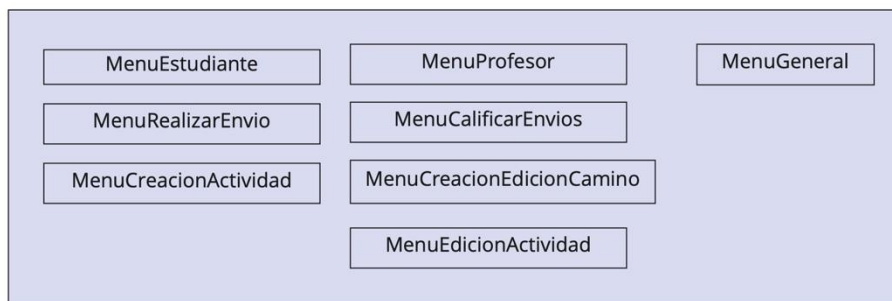
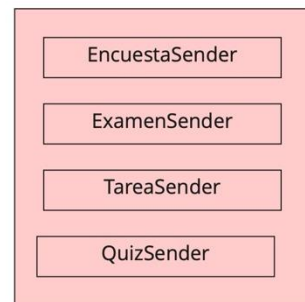
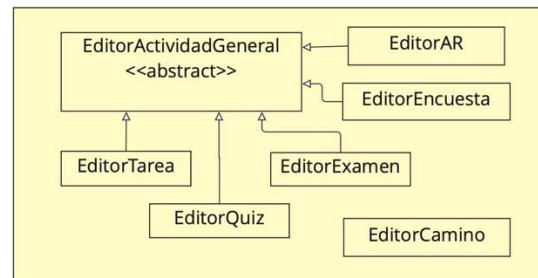
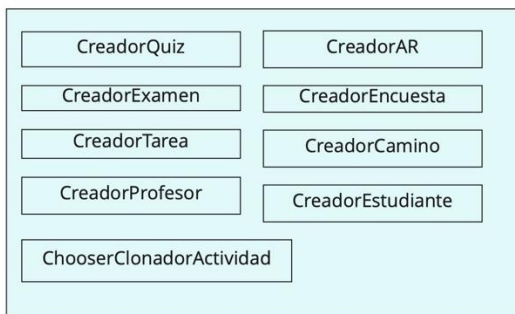
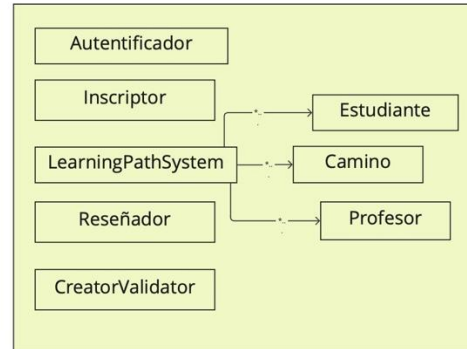
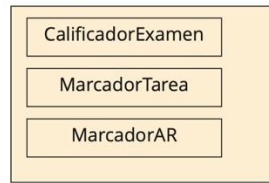
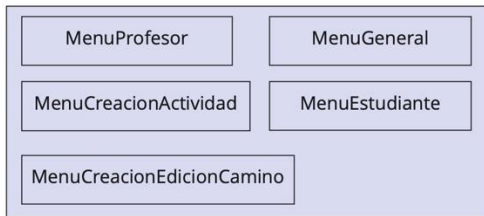


Diagrama alto nivel:





Diagramas de secuencia:

Diagrama de creacion de caminos:

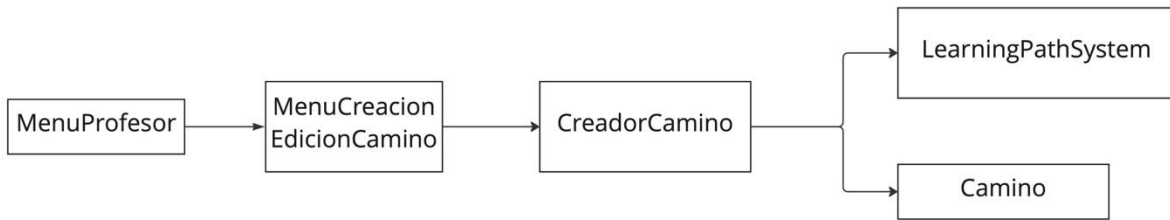


Diagrama de iniciar actividad:

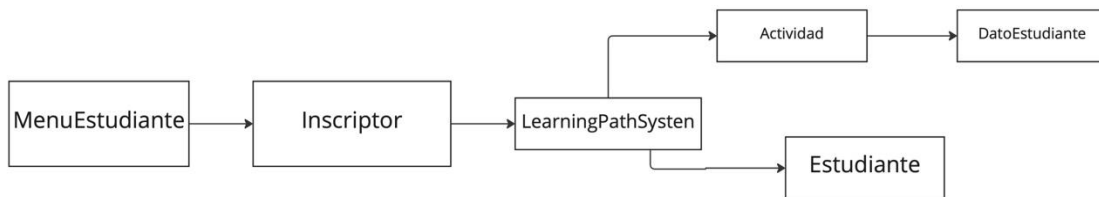


Diagrama para ver caminos:

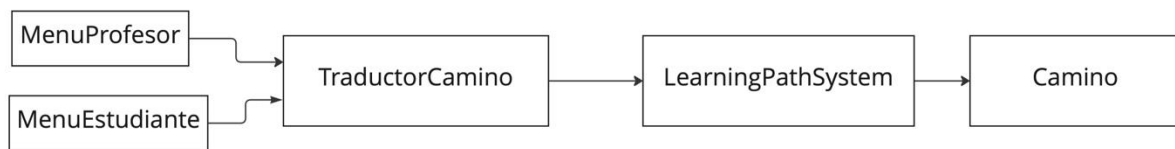


Diagrama para ver actividades:

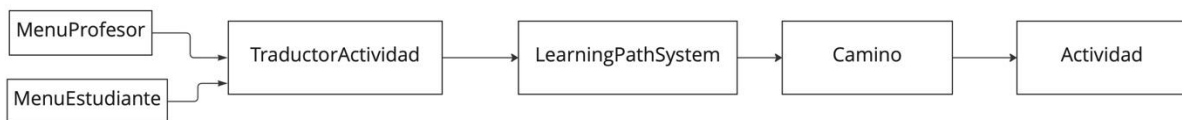


Diagrama para marcar como exitoso o calificar:

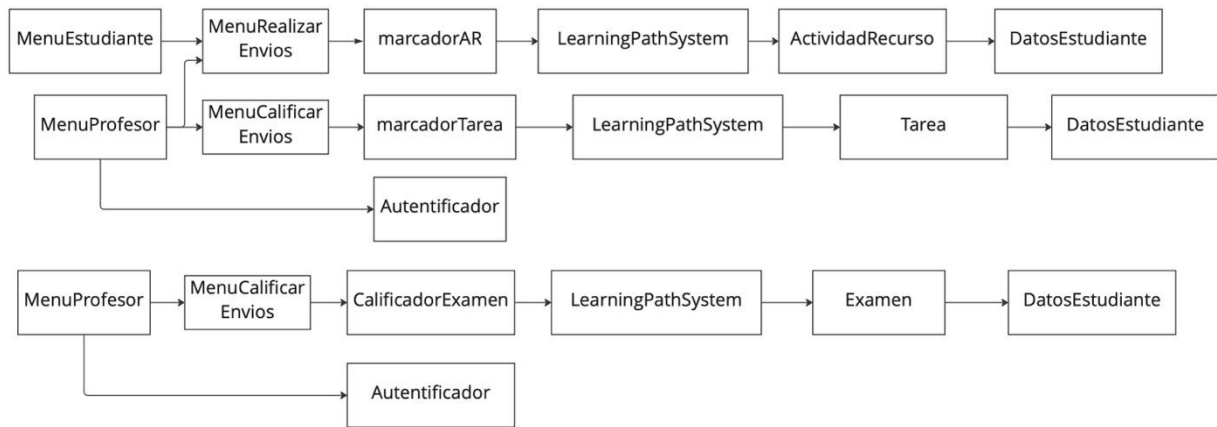


Diagrama de creacion de usuarios:

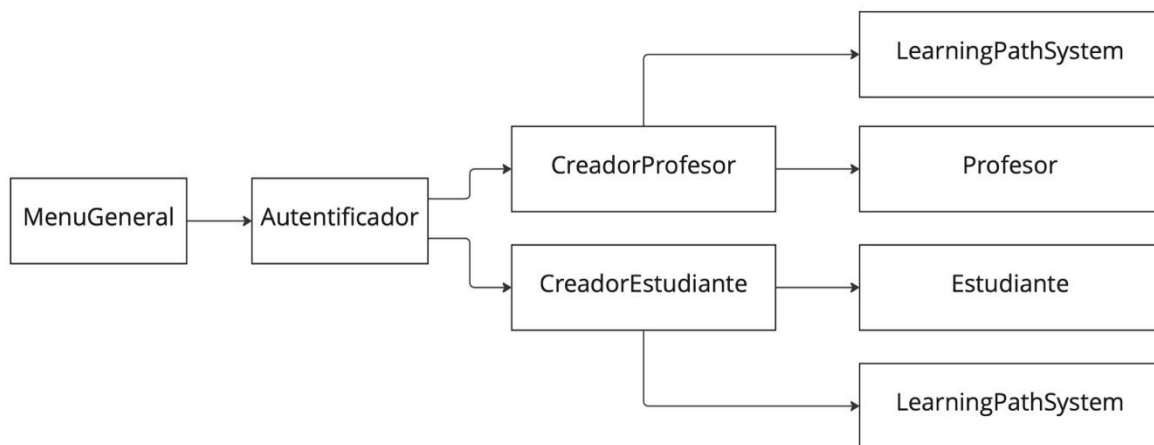


Diagrama de clonacion actividad:

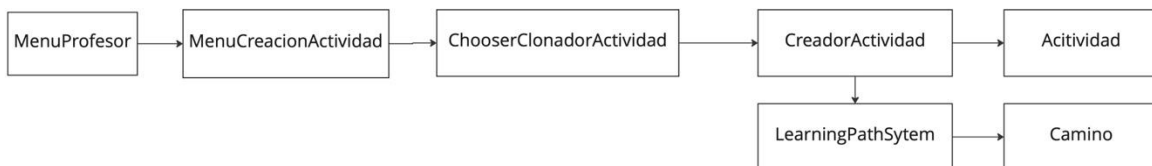


Diagrama de creacion actividades:

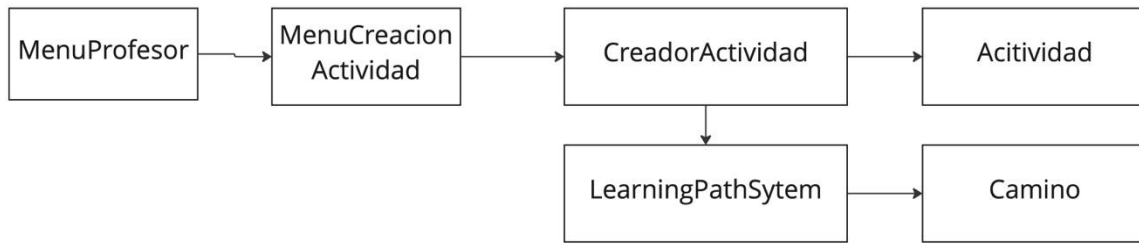


Diagrama de inscripcion en caminos:

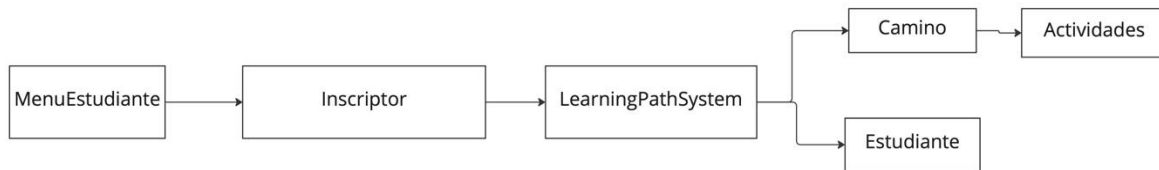


Diagrama de edición de actividades y caminos:

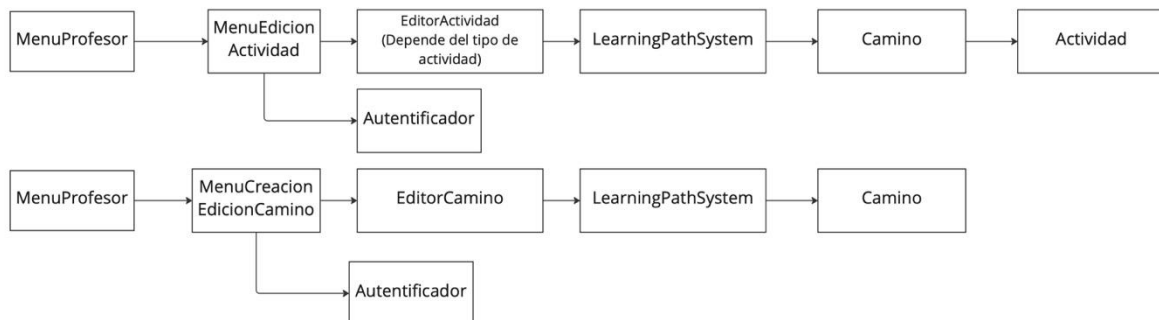


Diagrama de ver avances estudiante:

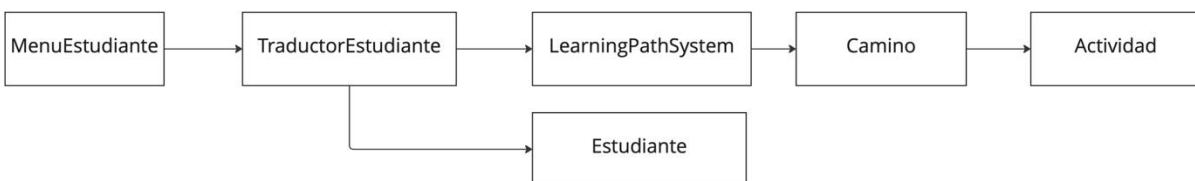


Diagrama de ver envíos:

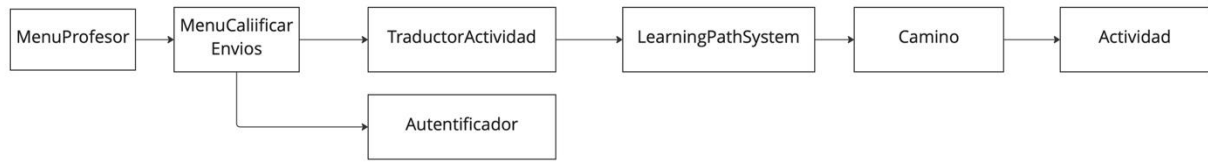


Diagrama de dejar reseñas y ratings:

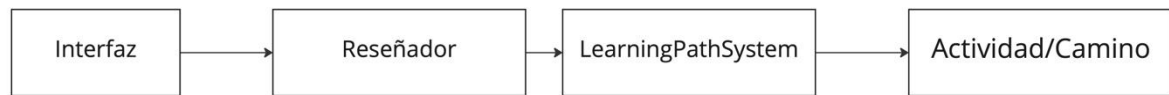
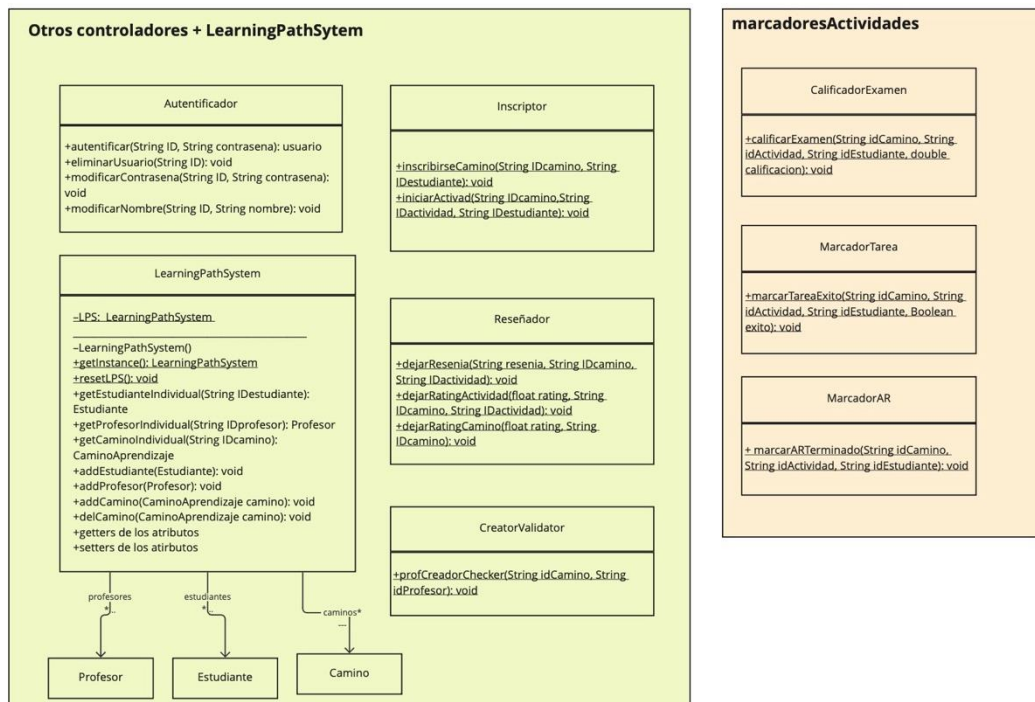
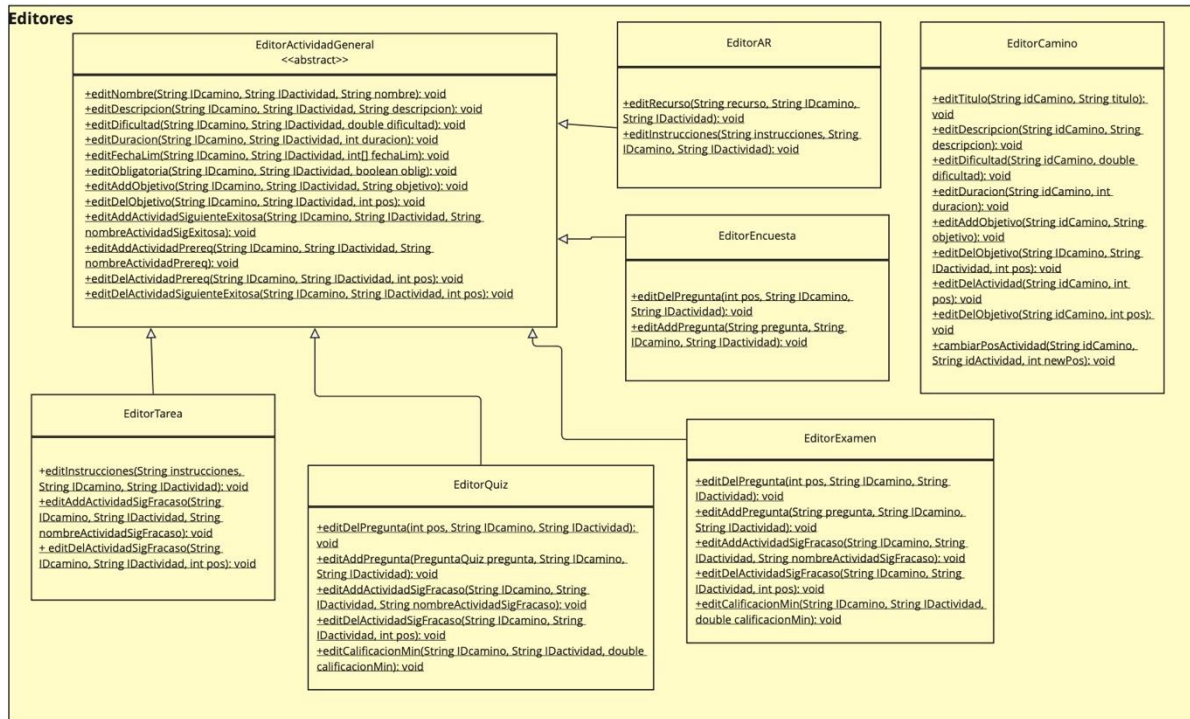


Diagrama de clases detallado:



Creadores

CreadorQuiz <pre> +crearQuizCero(String IDcamino, String nombre, String descripcion, List<String> objetivos, double dificultad, int duracion, int[] fechaLim, boolean obligatoria, double calificacionMin, List<PreguntaQuiz> preguntas, String IDprofesor, boolean verdaderoFalso, int pos): void +clonarQuiz(Actividad quizOG, String IDprofesor, String IDcaminoNuevo, int pos): void </pre>	CreadorAR <pre> +crearARCero(String IDcamino, String nombre, String descripcion, List<String> objetivos, double dificultad, int duracion, int[] fechaLim, boolean obligatoria, String recurso, String instrucciones, String IDprofesor, int pos): void +clonarAR(Actividad AROG, String IDprofesor, String IDcaminoNuevo, int pos): void </pre>	CreadorExamen <pre> +crearExamenCero(String IDcamino, String nombre, String descripcion, List<String> objetivos, double dificultad, int duracion, int[] fechaLim, boolean obligatoria, double calificacionMin, List<String> preguntas, String IDprofesor, int pos): void +clonarExamen(Actividad examenOG, String IDprofesor, String IDcaminoNuevo, int pos): void </pre>
CreadorEncuesta <pre> +crearEncuestaCero(String IDcamino, String nombre, String descripcion, List<String> objetivos, double dificultad, int duracion, int[] fechaLim, boolean obligatoria, List<String> preguntas, String IDprofesor, int pos): void +clonarEncuesta(Actividad encuestaOG, String IDprofesor, String IDcaminoNuevo, int pos): void </pre>	CreadorTarea <pre> +crearTareaCero(String IDcamino, String nombre, String descripcion, List<String> objetivos, double dificultad, int duracion, int[] fechaLim, boolean obligatoria, String instrucciones, String IDprofesor, int pos): void +clonarTarea(Actividad tareaOG, String IDprofesor, String IDcaminoNuevo, int pos): void </pre>	CreadorCamino <pre> +crearCaminoCero(String titulo, String descripcion, List<String> objetivos, double dificultad, int duracion, String IDprofesor): void +clonarCamino(String IDcaminoOG, String titulo, String IDprofesor): void </pre>
CreadorProfesor <pre> +crearProfeso(String login, String password, String nombre): void +eliminarProfesor(String IDprofesor): void </pre>	CreadorEstudiante <pre> +crearEstudiante(String login, String password, String nombre): void +eliminarEstudiante(String ID): void </pre>	ChooserClonadorActividad <pre> +ClonarActividad(String IDcaminoOG, String IDActividadOG, String IDprofesor, String IDcaminoNuevo, int pos): void </pre>

traductores

TraductorExamen <pre> +retornarPreguntas(String idCamino, String idActividad): List<String> +retornarCalificacinMin (String idCamino, String idActividad): double +retornarListaEstudiantesEnvios(String idCamino, String idActividad): HashMap<String, String[]> +retornarEnvioIndividual(String idCamino, String idActividad, String idEstudiante): List<String> </pre>	TraductorEncuesta <pre> +retornarPreguntas(String idCamino, String idActividad): List<String> +retornarEnvioIndividual(String idCamino, String idActividad, String idEstudiante): List<String> +retornarListaEstudiantesEnvios(String idCamino, String idActividad): HashMap<String, String> </pre>	TraductorProfesor <pre> + getIDfromLogin(String login): String +getNombrefromID(String idEstudiante): String +verCaminosCreados(String idProfesor): HashMap<String, String> </pre>
TraductorActividad <pre> +getIDfromNombre(String idCamino, String nombre): String +getTipoQuiz(String idCamino, String idActividad): boolean +verInfoGeneralActividad(String idCamino, String idActividad): HashMap<String, String> </pre>	TraudctorCamino <pre> +verTodosCaminos (): HashMap<String, String> +verInfoGeneralCamino(String IDCamino): HashMap<String, String> +verActividadesCamino(String IDCamino): List<String[]> +getIDfromNombre(String titulo): String </pre>	TraductorQuiz <pre> +retornarPreguntasSinRespuesta(String idCamino, String idActividad): List<String[]> +retornarPreguntasConRespuesta(String idCamino, String idActividad): List<String[]> +retornarCalificacinMin (String idCamino, String idActividad): double </pre>
TraductorAR <pre> +retornarInstruccionesRecurso(String idCamino, String idActividad): String[] </pre>	TraductorTarea <pre> +retornarInstrucciones (String idCamino, String idActividad): String +retornarListaEstudiantesEntrega(String idCamino, String idActividad): HashMap<String, String[]> + retornarEntregaIndividual(String idCamino, String idActividad, String idEstudiante): String </pre>	TraductorEstudiante <pre> +getAvancesCaminos(String IDestudiante): HashMap<String, String> +getAvancesCaminoIndividual(String IDestudiante, String idCamino): HashMap<String, String> + getIDfromLogin(String login): String +getNombrefromID(String idEstudiante): String +verActividadActiva(String idEstudiante): String </pre>

```

classDiagram
    class ActividadesPersistencia {
        +guardarActividad(Actividad actividad, String caminoID, String pathCaminos): void
        +cargarActividad(JSONObject jActividad, String pathActividad): Actividad
    }
    class CaminosPersistencia {
        +GuardarCaminoSingular(CaminoAprendizaje camino, String pathCaminos, File fileCaminosDirectorio): void
        +cargarCamino(JSONObject jcamino, String pathCarpetas): CaminoAprendizaje
    }
    class DatosEstudiantesPersistencia {
        +guardarDatosEstudiante(DatosEstudianteActividad datosEst, String caminoID, String actividadID, String pathCaminos): void
        +cargarDatosEstudiante(JSONObject jDatosEst): DatosEstudianteActividad
    }
    class CentralPersistencia {
        +cargarTodo(boolean test): void
        +guardarTodo(boolean test): void
        +guardarCaminosActividadesDatosEstudiante(boolean test): void
        +cargarCaminosActividadesDatosEstudiante(boolean test): void
        +cargarEstudiantes(boolean test): void
        +guardarEstudiantes(boolean test): void
        +guardarProfesores(boolean test): void
        +cargarProfesores(boolean test): void
    }
    class metodosAuxPersistencia {
        +deleteDirectory(File directoryToBeDeleted): boolean
        +cleanDatos(boolean test): void
        +contadorLineasCaminosDirectorio(boolean test): int
        +contadorFoldersCaminos(boolean test): int
    }
    class EstudiantesPersistencia {
        +guardarEstudianteSingular(Estudiante estudiante, String pathCarpetasEstudiante, String pathDirectorio): void
        +cargarListaCaminosEstudiantes(JSONObject jEstudiante): List<CaminoAprendizaje>
        +cargarEstudiante(JSONObject jEstudiante): Estudiante
    }
    class ProfesoresPersistencia {
        +guardarProfesorSingular(Profesor profesor, String pathProfesor, String pathDirectorio): void
        +cargarListaCaminosProfesor(JSONObject jProfesor): List<CaminoAprendizaje>
        +cargarProfesor(JSONObject jProfesor): Profesor
    }
  
```

ActividadesPersistencia

- +guardarActividad(Actividad actividad, String caminoID, String pathCaminos): void
- +cargarActividad(JSONObject jActividad, String pathActividad): Actividad

CaminosPersistencia

- +GuardarCaminoSingular(CaminoAprendizaje camino, String pathCaminos, File fileCaminosDirectorio): void
- +cargarCamino(JSONObject jcamino, String pathCarpetas): CaminoAprendizaje

DatosEstudiantesPersistencia

- +guardarDatosEstudiante(DatosEstudianteActividad datosEst, String caminoID, String actividadID, String pathCaminos): void
- +cargarDatosEstudiante(JSONObject jDatosEst): DatosEstudianteActividad

CentralPersistencia

- +cargarTodo(boolean test): void
- +guardarTodo(boolean test): void
- +guardarCaminosActividadesDatosEstudiante(boolean test): void
- +cargarCaminosActividadesDatosEstudiante(boolean test): void
- +cargarEstudiantes(boolean test): void
- +guardarEstudiantes(boolean test): void
- +guardarProfesores(boolean test): void
- +cargarProfesores(boolean test): void

metodosAuxPersistencia

- +deleteDirectory(File directoryToBeDeleted): boolean
- +cleanDatos(boolean test): void
- +contadorLineasCaminosDirectorio(boolean test): int
- +contadorFoldersCaminos(boolean test): int

EstudiantesPersistencia

- +guardarEstudianteSingular(Estudiante estudiante, String pathCarpetasEstudiante, String pathDirectorio): void
- +cargarListaCaminosEstudiantes(JSONObject jEstudiante): List<CaminoAprendizaje>
- +cargarEstudiante(JSONObject jEstudiante): Estudiante

ProfesoresPersistencia

- +guardarProfesorSingular(Profesor profesor, String pathProfesor, String pathDirectorio): void
- +cargarListaCaminosProfesor(JSONObject jProfesor): List<CaminoAprendizaje>
- +cargarProfesor(JSONObject jProfesor): Profesor

senders

```
classDiagram
    class EncuestaSender {
        +sendEnvioEncuesta(String idCamino, String idActividad, String idEstudiante, HashMap<String, String> respuestas): void
    }
    class ExamenSender {
        +sendEnvioExamen(String idCamino, String idActividad, String idEstudiante, HashMap<String, String> respuestas): void
    }
    class TareaSender {
        +addMetodoEntregaTarea(String idCamino, String idActividad, String idEstudiante, String metodoEntrega): void
    }
    class QuizSender {
        +sendEnvioQuiz(String idCamino, String idActividad, String idEstudiante, HashMap<String, Integer> respuestas): void
    }
```

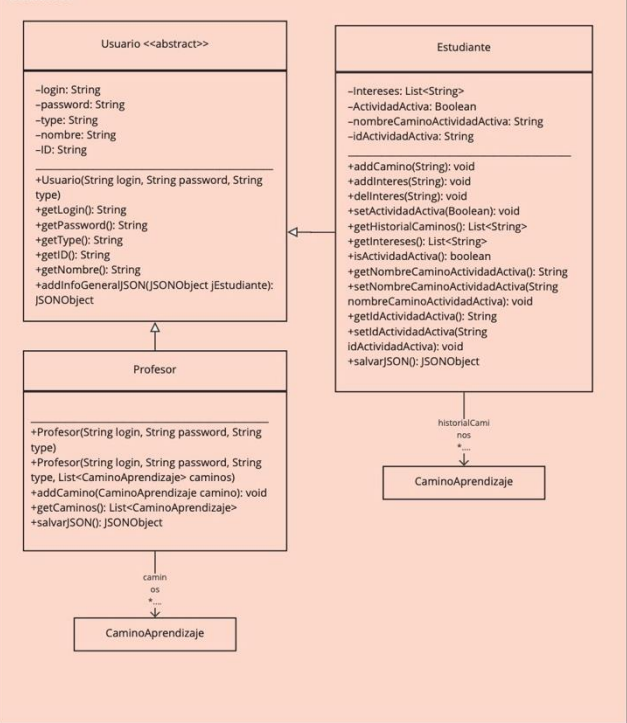
```
classDiagram
    class EncuestaSender {
        +sendEnvioEncuesta(String idCamino, String idActividad, String idEstudiante, HashMap<String, String> respuestas): void
    }
    class ExamenSender {
        +sendEnvioExamen(String idCamino, String idActividad, String idEstudiante, HashMap<String, String> respuestas): void
    }
    class TareaSender {
        +addMetodoEntregaTarea(String idCamino, String idActividad, String idEstudiante, String metodoEntrega): void
    }
    class QuizSender {
        +sendEnvioQuiz(String idCamino, String idActividad, String idEstudiante, HashMap<String, Integer> respuestas): void
    }
```

Envios

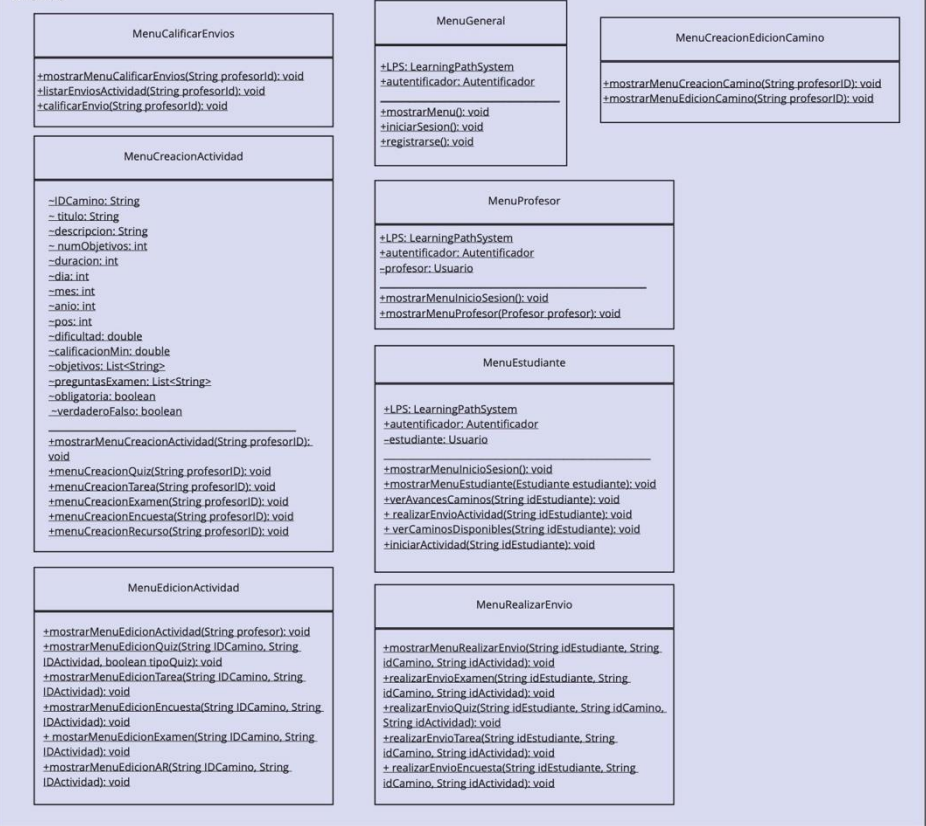
```
classDiagram
    class Envio {
        <<interface>>
        -Respuestas: HashMap<K, String>
    }
    class EnvioExamen {
        -Respuestas: HashMap<String, String>
        -calificacion: double
        +EnvioExamen(HashSet<String, String>)
        +EnvioExamen()
        +getRespuestas(): HashMap<String, String>
        +getCalificacion(): double
        +setCalificaicon(double calificacion): void
    }
    class EnvioEncuesta {
        -Respuestas: HashMap<String, String>
        +EnvioEncuesta(HashSet<String, String>)
        +EnvioEncuesta()
        +getRespuestas(): HashMap<String, String>
    }
    class EnvioQuiz {
        -Respuestas: HashMap<PreguntaQuiz, Integer>
        -calificacion: double
        +EnvioQuiz(HashMap<PreguntaQuiz, Integer>)
        +EnvioQuiz()
        +getRespuestas(): HashMap<PreguntaQuiz, Integer>
        +calcularCalificacionQuiz(): double
        +getCalificacion(): double
        +setCalificacion(double calificacion): void
    }
    Envio <|-- EnvioEncuesta
    Envio <|-- EnvioQuiz
    EnvioExamen <|-- Envio
    EnvioExamen --> Envio
```

The diagram illustrates the structure of the Envios package. It features four classes: Envio, EnvioExamen, EnvioEncuesta, and EnvioQuiz. Envio is an interface with a single method -Respuestas: HashMap<K, String>. EnvioExamen is a class that implements Envio, as indicated by the solid line with a hollow triangle pointing to Envio. EnvioExamen also has a directed association to Envio, shown by a solid line with an open arrowhead. EnvioEncuesta and EnvioQuiz are subclasses of Envio, indicated by solid lines with hollow triangle heads pointing to Envio. EnvioEncuesta and EnvioQuiz both inherit the -Respuestas: HashMap<String, String> attribute from Envio. EnvioExamen has its own attributes: -Respuestas: HashMap<String, String> and -calificacion: double. It also has methods: +EnvioExamen(HashSet<String, String>), +EnvioExamen(), +getRespuestas(): HashMap<String, String>, +getCalificacion(): double, and +setCalificaicon(double calificacion): void. EnvioEncuesta has methods: +EnvioEncuesta(HashSet<String, String>), +EnvioEncuesta(), and +getRespuestas(): HashMap<String, String>. EnvioQuiz has methods: +EnvioQuiz(HashMap<PreguntaQuiz, Integer>), +EnvioQuiz(), +getRespuestas(): HashMap<PreguntaQuiz, Integer>, +calcularCalificacionQuiz(): double, +getCalificacion(): double, and +setCalificacion(double calificacion): void.

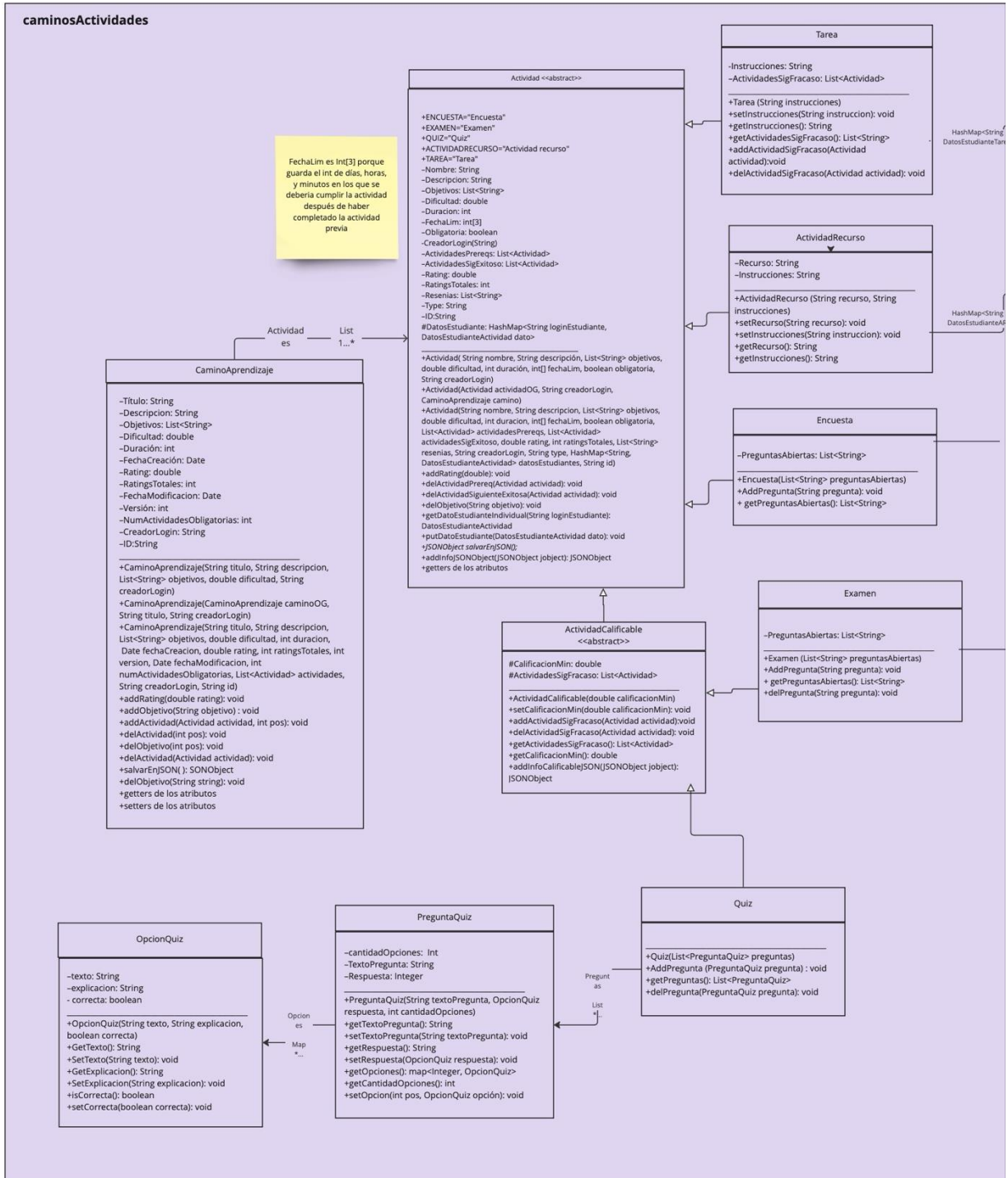
Usuarios

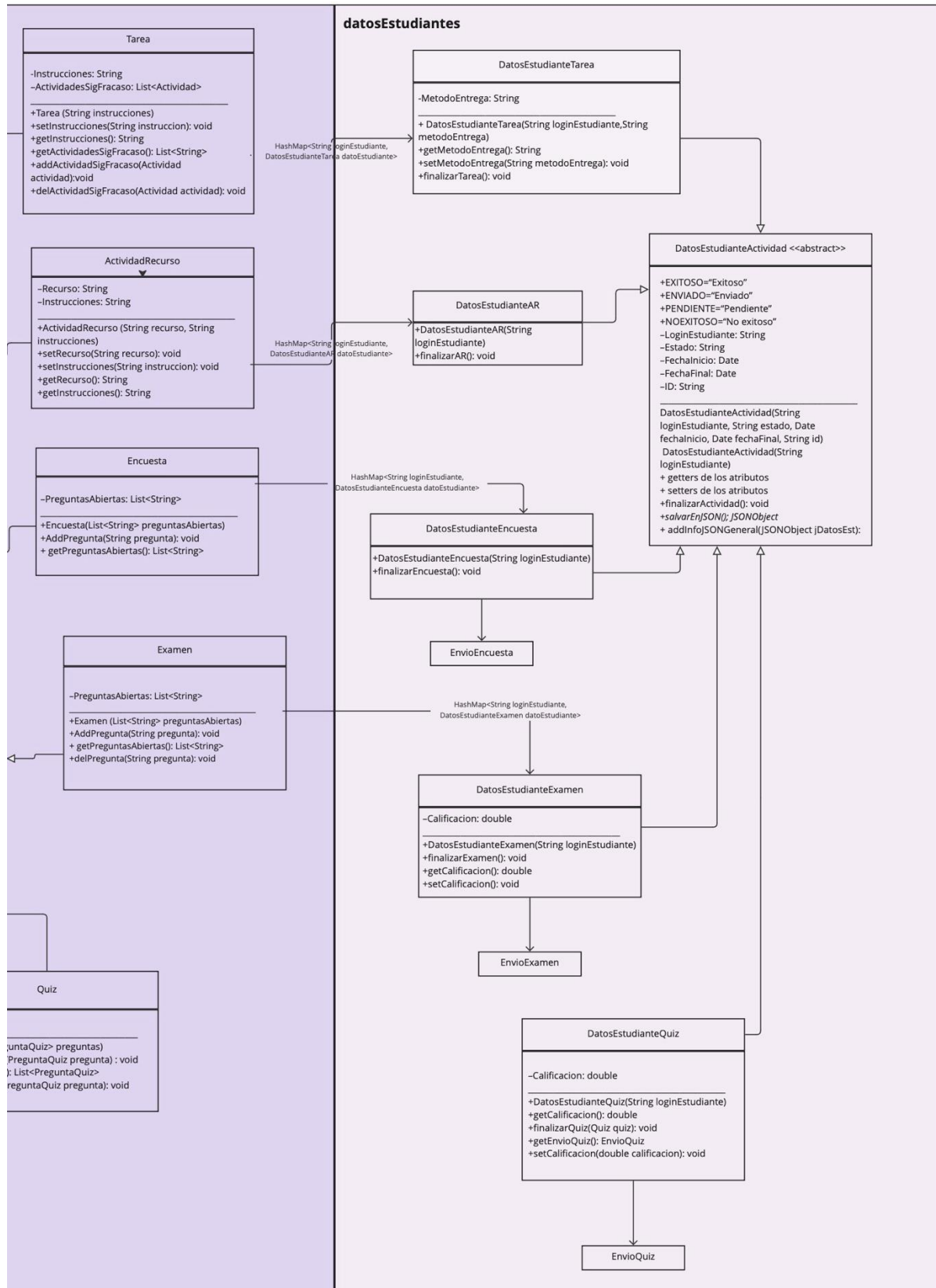


menus



caminosActividades





Historias de usuario:

Profesor como creador parte 1:

Gojo Satoru es un profesor que busca usar nuestra aplicacion, creando un perfil de profesor con el login "Gojo999", contraseña "Gojo123" y su nombre "Gojo Satoru".

Luego crea un camino llamado "El maravilloso mundo de los cuervos", con la siguiente información:

- Objetivos: "Saber la diferencia entre distintos tipos de cuervos."; "Poder sustentar porque los cuervos son tan increíbles."; "Volverse fan de los cuervos".
- Descripción: "Esto es un curso que te enseña lo increíble que son los cuervos"
- Dificultad: 1.5

A este camino le quiere añadir seis actividades con las siguientes características en el siguiente orden:

1. Quiz verdadero o falso:

- Nombre: "Quiz de verdadero o falso de cuervos"
- Descripción: "Una actividad para aprender más de los cuervos."
- Dificultad: 1.5
- Duración: 20
- Sí es obligatoria
- Calificación mínima: 2.5
- Fecha límite: 1 día, 0 horas y 0 minutos
- Mismos objetivos que el camino.
- Preguntas:
 - Texto: "El cuervo es el mejor animal del mundo"
 - Verdadero
 - Opción correcta: Si
 - Explicación: "Porque lo son. Punto"
 - Falso
 - Opción correcta: No
 - Explicación: "Ningún animal se compara a los cuervos"
 - Texto: "El siguiente presidente debería ser un cuervo"
 - Verdadero
 - Opción correcta: Si
 - Explicación: "Porque si, all hail the mighty crow presidente"
 - Falso
 - Opción correcta: No
 - Explicación: "Los humanos son más malvados que los cuervos"

2. Quiz opción múltiple:

- Nombre: "Quiz de 4 opciones de cuervos"
- Dificultad: 1.5
- Duración: 20 minutos
- No es obligatoria
- Calificación mínima: 3
- Descripción: "Una actividad para aprender más de los cuervos."
- Fecha límite: 1 día, 0 horas y 0 minutos
- Mismos objetivos que el camino.
- Preguntas:
 - Texto: "Si quiero representar el número de cuervos que tengo, que tipo de variable debería usar?"
 - Opción: "int"
 - Correcta: true
 - Explicacion: "Porque es un entero lo que necesitas representar"
 - Opción: "double"
 - false
 - "Porque no puedes tener decimales de cuervos"
 - Opción: "float"
 - False
 - "Porque no puedes tener decimales de cuervos"
 - Opción: "string"
 - False
 - "Porque es un numero lo que necesitas"
 - Texto: "Cual es el mejor pájaro?"
 - Opción: "Paloma"
 - false
 - "Porque son sucias"
 - Opción: "Cuervo"
 - True
 - "Porque son hermosos e inteligentes"
 - Opción: "Pechirrojo"
 - False
 - "Tierno pero es muy pequeño"
 - Opción: "Vaca"
 - False
 - "La vaca no es un pájaro"

3. Examen:

- Nombre: "Examen de cuervos"
- Descripción: "Una actividad para aprender más de los cuervos."
- Dificultad: 2

- Duración: 20 minutos
 - Fecha límite: 1 día, 0 horas y 0 minutos
 - Si es obligatoria
 - Calificación mínima: 3
 - Mismos objetivos que el camino.
 - Preguntas:
 - "¿Cuales son las características principales de los cuervos?"
 - "¿Que otro animal es parecido e igual de increíble que los cuervos?"
 - "¿Por que son mejores los cuervos que otros pajaros?"
4. Tarea:
- Nombre: "Tarea de cuervos"
 - Descripción: "Una actividad para aprender más de los cuervos."
 - Dificultad: 1.5
 - Duración: 120 minutos
 - Fecha límite: 1 día, 0 horas y 0 minutos
 - No es obligatoria
 - Mismos objetivos que el camino.
 - Instrucciones: "Tomar fotos de cuervos"
5. Actividad de recurso:
- Nombre: "Lectura sobre cuervos"
 - Descripción: "Una actividad para aprender más de los cuervos."
 - Fecha límite: 1 día, 0 horas y 0 minutos
 - Mismos objetivos que el camino.
 - Recurso: <https://www.audubon.org/es/guia-de-aves/ave/cuervo-comun>
 - Instrucciones: "Leer el artículo"
 - Si es obligatoria
6. Encuesta:
- Nombre: "Encuesta de satisfacción del curso"
 - Descripción: "Una actividad para aprender más de los cuervos."
 - Dificultad: 3
 - Duración: 60 minutos
 - Fecha límite: 1 día, 0 horas y 0 minutos
 - Mismos objetivos que el camino.
 - No es obligatoria
 - Preguntas:
 - "¿Qué opinaste del camino?"
 - "¿Qué podría mejorar?"
 - "¿Cuál fue tu actividad favorita?"

El profesor después puede ver su camino y actividades, y podrá visualizar la información de cada uno de ellos:

- Información del camino: Descripción, títulos, versiones, fechas de creación, actividades, rating.
- Información de cada una de las actividades: Nombre, descripción, objetivos, duración, fecha límite, dificultad, tipo de actividad, actividades prerequisites, actividades siguientes exitosas, reseñas y rating.
- Información del examen: Preguntas y calificación mínima.
- Información de la tarea: Instrucciones.
- Información de los quizzes: Preguntas y calificación mínima.
- Información de la encuesta: Preguntas
- Información de la actividad de recurso: Recurso y las instrucciones.

Profesor como creador parte 2:

El profesor Gojo Satoru quiere clonar el camino “El maravilloso mundo de los cuervos” para crear un nuevo camino llamado “El maravilloso mundo de los zorros”. Quiere que se copie todas las actividades, por ende al mostrar la información del “El maravilloso mundo de los zorros” y sus actividades, deberían ser igual que las de “El maravilloso mundo de los cuervos”.

Profesor como creador parte 3:

El profesor Gojo Satoru quiere editar el camino “El maravilloso mundo de los zorros”. Primero se autentifica con su contraseña “Gojo123” y login “Gojo999”. Después selecciona el camino “El maravilloso mundo de los zorros” y accede al menú de edición. La nueva información que quiere poner y cambiar es:

- Objetivos: Borrar todos los objetivos y añadirle “Ser un experto en zorros”.
- Descripción: “Esto es un curso que te enseña lo increíble que son los zorros”
- Dificultad: 2
- Borrar el quiz de verdadero o falso.
- Cambiar el nombre de la actividad de recurso a “Primera lectura de zorros” y ponerla en el primer lugar.
- Añadir de último lugar una nueva actividad de recurso de lectura clonada de la actividad de recurso que antes existía, solo que con el nombre “Lectura adicional de zorros”.

Cuando se pida la información de este camino, debe aparecer la información actualizada y cambiada del camino.

Estudiante:

Trey Clover es un estudiante que quiere usar nuestra aplicación. Él se quiere inscribir al curso “El maravilloso mundo de los cuervos”. Quiere hacer las actividades en el orden indicado por el camino.

- Hace el quiz verdadero o falso con las siguientes respuestas:
 - Texto: “El cuervo es el mejor animal del mundo”
 - Respuesta: Verdadero

- Texto: "El siguiente presidente debería ser un cuervo"
 - Respuesta: Verdadero
- Saco todas las respuestas bien, entonces Trey aprueba el quiz.
- Hace el quiz de opcion multiple con las siguientes respuestas:
 - Pregunta: "Si quiero representar el número de cuervos que tengo, que tipo de variable debería usar?"
 - Opción: int
 - Pregunta: "Cual es el mejor pájaro?"
 - Opción: "Vaca"
- Ya que saco solo una respuesta correcta de dos, y la calificacion mínima para pasar el tres, entonces Trey reprueba el quiz.
- Hace el examen respondiendo:
 - "¿Cuales son las características principales de los cuervos?"
 - "Que son inteligentes hermosos y maravillosos en todos los aspectos"
 - "¿Que otro animal es parecido e igual de increíble que los cuervos?"
 - "Los zorros tambien son increíbles, pero los cuervos aún así son mejores"
 - "¿Por que son mejores los cuervos que otros pajaros?"
 - "Porque son los más inteligentes"
- Hace la tarea diciendo que entrego las fotos por correo.
- Lee el recurso de la actividad de recurso y lo marca como terminado.
- Hace la encuesta respondiendo:
 - "¿Qué opinaste del camino?"
 - "Maravilloso camino se merece un 5"
 - "¿Qué podría mejorar?"
 - "Nada fue increíble"
 - "¿Cuál fue tu actividad favorita?"
 - "La tarea de tomar fotos"

Profesor como calificador:

Gojo Satoru quiere calificar el examen de Trey Clover del camino "El maravilloso mundo de los cuervos". Primero se autentifica, mira la lista de envios, selecciona el envio de Trey Clover para mirarlo, y le pone calificacion "4". Esto debería quedar guardado de tal manera que Trey pueda ver el avance

Gojo luego quiere marcar como exitosa la entrega en la tarea de Trey Clover del camino "El maravilloso mundo de los cuervos". Mira la lista de envios, selecciona el envio de Trey Clover para mirarlo, y dice que si fue exitoso.

Por último, Gojo quiere ver las respuestas de Trey en la encuesta. Mira la lista de envios y selecciona el envio de Trey Clover para mirarlo.

