Juan Manuel Rojas 202321306
Julián Restrepo 202320177
Julián Pinto 202321207

Con base en nuestro análisis de la entrega 1, llegamos a la conclusión de que para organizar y representar adecuadamente todos los componentes y relaciones del sistema, es necesario trabajar con tres diagramas principales. Estos diagramas nos permiten visualizar las interacciones entre las distintas clases, usuarios y actividades, así como entender el flujo de información y las responsabilidades clave de cada elemento dentro del sistema. A continuación, presentamos el desglose de cada uno de estos diagramas y sus componentes, junto con las justificaciones de diseño detrás de las decisiones tomadas.

Diagrama de clases de diseño:

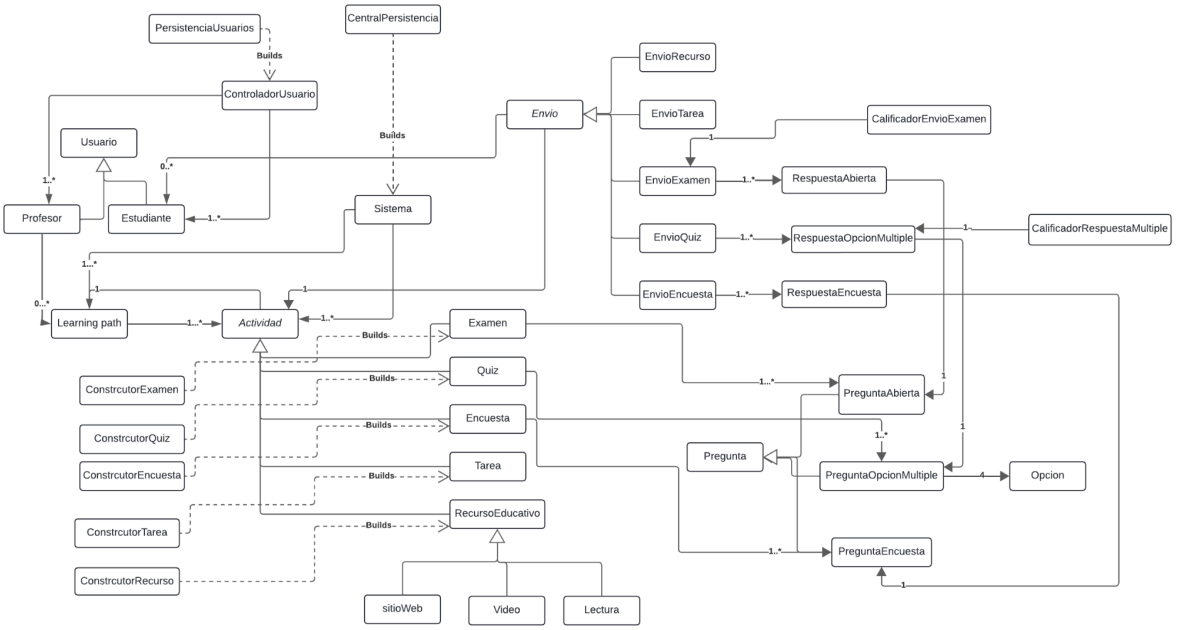
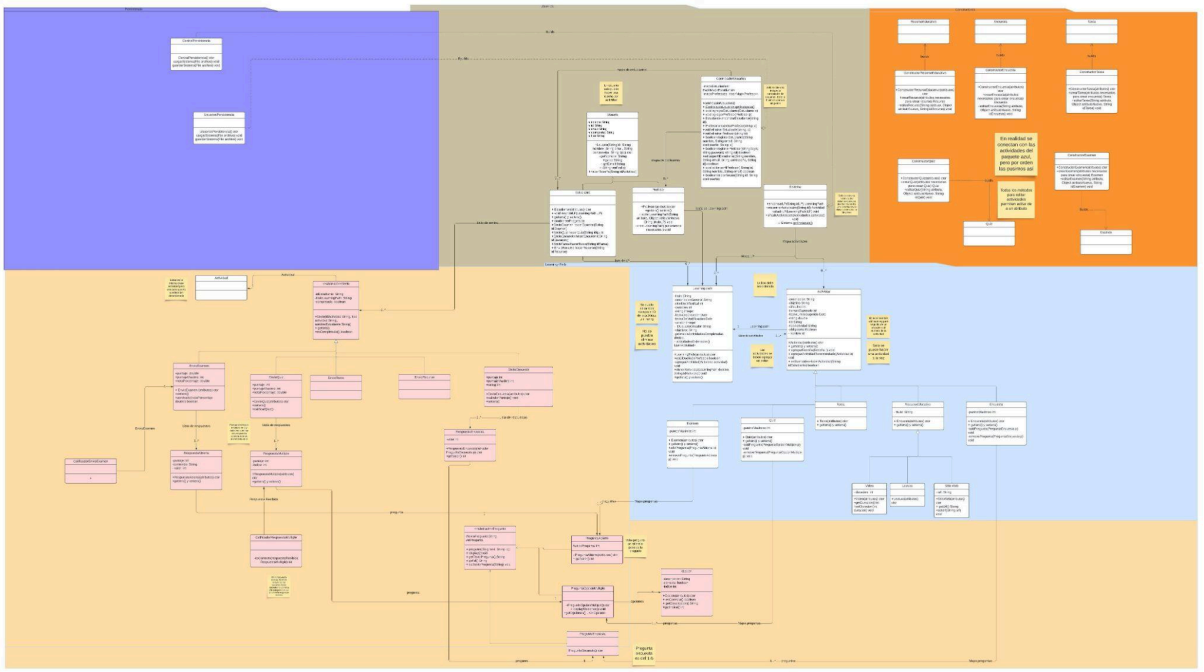


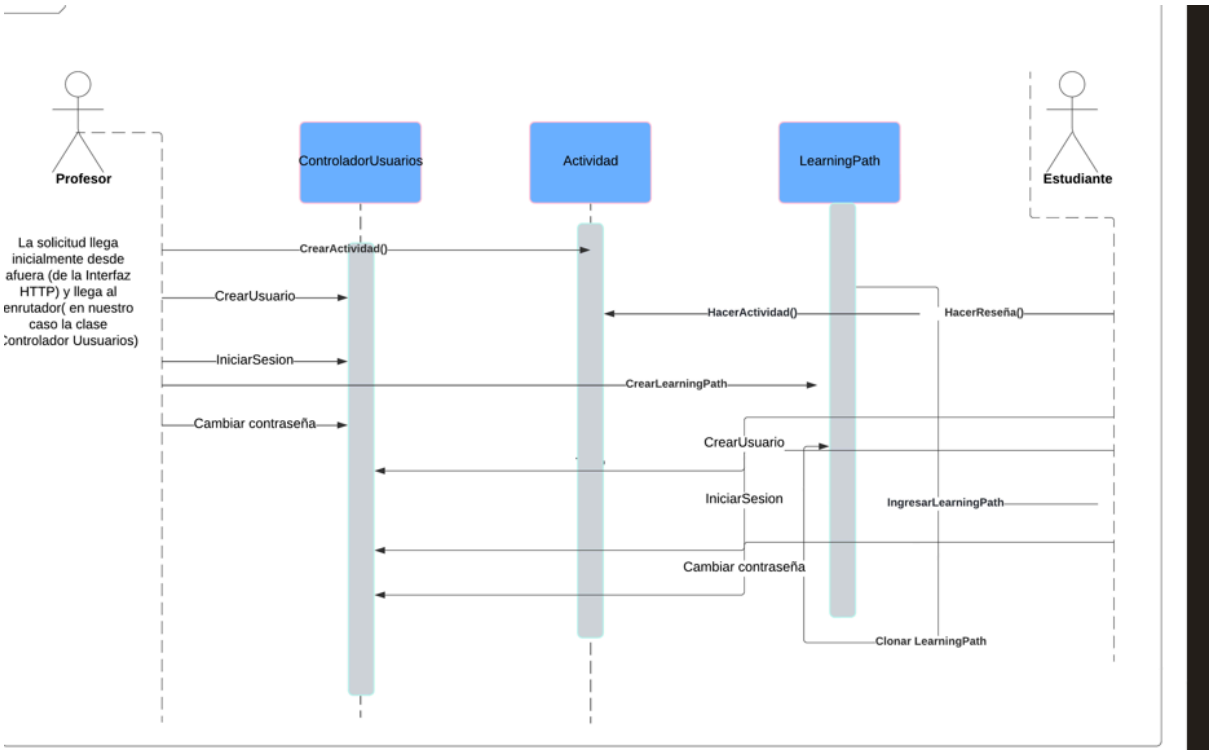
Diagrama de clases de alto nivel:



Ambos diagramas se pueden observar mejor en el PDF encontrado en la carpeta de la entrega

Diagrama de secuencia de las principales funcionalidades(las mismas que están en el main de la aplicación):

Crear LP y Actividad:
Editar y Clonar LP:
ControlDeUsuarios:



Responsabilidades de cada componente en el diseño final:

1. **Usuario**: Representa la abstracción general de un usuario en el sistema. Se especializa en dos tipos de usuarios: **Profesor** y **Estudiante**. Esta clase se encarga de mantener la información común de cualquier usuario (como identificadores o datos personales) y sirve de base para que las subclases definan funcionalidades específicas.
2. **Profesor**: Es un tipo de usuario que puede crear y gestionar **Learning Paths** y actividades, además de calificar exámenes, quizzes y tareas. La responsabilidad principal es permitir la interacción y creación de contenido educativo dentro del sistema.
3. **Estudiante**: Es un usuario que puede inscribirse en **Learning Paths**, realizar las actividades disponibles, y recibir feedback sobre su progreso. Se encarga del seguimiento del avance a través de las actividades y rutas de aprendizaje.
4. **Learning Path**: Estructura que contiene una secuencia de **Actividades** que los estudiantes deben completar. La clase define la lógica de manejo de los caminos de aprendizaje y permite a los profesores organizar las actividades en secuencias personalizadas.
5. **Actividad**: Clase abstracta que representa cualquier tipo de actividad que pueda formar parte de un **Learning Path**. Las actividades pueden ser exámenes, quizzes, encuestas, tareas o recursos educativos, y heredan de esta clase. Su responsabilidad es proporcionar los atributos y métodos comunes para las actividades.
6. **Examen, Quiz, Encuesta, Tarea, Recurso Educativo**: Estas son subclases especializadas de **Actividad**. Cada una representa un tipo específico de actividad con su propia lógica y estructura.
    - **Examen**: Contiene preguntas abiertas que requieren ser calificadas por un profesor.
    - **Quiz**: Consiste en preguntas de opción múltiple que se califican automáticamente.
    - **Encuesta**: Requiere respuestas abiertas del estudiante pero no implica una calificación estricta.
    - **Tarea**: Requiere una entrega externa (fuera del sistema), la cual debe ser calificada por un profesor.
    - **Recurso Educativo**: Es una actividad pasiva que se completa al consumir un recurso como un video o lectura.
7. **Pregunta**: Es la clase base para los diferentes tipos de preguntas en un examen, quiz o encuesta. Define los elementos fundamentales de una pregunta, como su enunciado y opciones de respuesta.
8. **PreguntaAbierta, PreguntaOpcionMultiple, PreguntaEncuesta**: Son subclases de **Pregunta** que representan preguntas específicas:
    - **PreguntaAbierta**: Requiere una respuesta textual y es utilizada en exámenes y encuestas.
    - **PreguntaOpcionMultiple**: Es usada en quizzes y contiene varias opciones, con una correcta.

    - **PreguntaEncuesta**: Similar a la pregunta abierta pero usada exclusivamente en encuestas.
9. **Opción**: Se utiliza exclusivamente en **PreguntaOpcionMultiple**. Define las opciones de respuesta para una pregunta de opción múltiple.
10. **Envio**: Representa una instancia en la que un estudiante ha enviado una actividad. Se especializa en tipos de envío como **EnvioExamen**, **EnvioQuiz**, **EnvioTarea**, **EnvioEncuesta**, etc. El propósito de esta clase es gestionar el seguimiento y registro de actividades completadas.
11. **CalificadorEnvioExamen, CalificadorRespuestaMultiple**: Son clases que tienen la responsabilidad de calificar exámenes y quizzes respectivamente. El **CalificadorEnvioExamen** se encarga de enviar las respuestas de preguntas abiertas al profesor y devolver la nota final basada en el peso de las preguntas. El **CalificadorRespuestaMultiple** es responsable de calificar automáticamente las respuestas de opción múltiple en quizzes.
12. **ControladorUsuario**: Esta clase es responsable de gestionar los registros y autenticaciones de **Profesor** y **Estudiante**. Como singleton, centraliza el manejo de usuarios en el sistema y garantiza que cada usuario esté correctamente registrado y autenticado.
13. **Sistema**: Esta clase es la responsable de gestionar todos los LearningPaths y Actividades dentro del sistema. Como singleton, centraliza el manejo del sistema permitiendo encontrar LP y Actividades con base en su id.
14. **PersistenciaUsuarios, CentralPersistencia**: Son clases encargadas de la persistencia de los datos. **PersistenciaUsuarios** guarda y recupera la información de usuarios, mientras que **CentralPersistencia** se encarga de gestionar de manera general la persistencia del sistema.
15. **Constructores** (ConstructorExamen, ConstructorQuiz, ConstructorEncuesta, ConstructorTarea, ConstructorRecurso): Cada uno de estos constructores es responsable de la creación de instancias específicas de actividades, siguiendo el patrón de diseño *Builder*. Su rol es proporcionar una manera flexible de construir actividades con diferentes configuraciones, separando la lógica de creación de la lógica de negocio.

Justificaciones de las decisiones del diseño:

1. **Patrón de Herencia para las Actividades**: Se ha utilizado una clase abstracta **Actividad** de la cual heredan los diferentes tipos de actividades (Examen, Quiz, Encuesta, etc.). Esta decisión se justifica porque todas las actividades comparten atributos y comportamientos comunes, como descripciones, duración, nivel de dificultad, entre otros. La herencia reduce duplicación de código y permite que las subclases añadan o sobrescriban comportamientos específicos.
2. **Composición de Preguntas**: La clase **Examen** y **Quiz** no tienen directamente una lista de preguntas, sino que dependen de la clase **Pregunta** y sus subtipos. Esto promueve un diseño flexible que permite agregar nuevos tipos de preguntas en el futuro sin afectar las actividades ya existentes. Además, permite compartir lógica común entre diferentes tipos de preguntas.

3. **Uso de Constructores**: El uso de constructores (patrón *Builder*) para crear instancias de actividades permite manejar la complejidad de las configuraciones de una manera más controlada y legible. Esta decisión permite una mayor flexibilidad al crear actividades, ya que se pueden añadir o modificar pasos en la construcción de una actividad sin afectar otras partes del sistema.
4. **Persistencia Separada**: La decisión de separar la lógica de persistencia en clases especializadas como **PersistenciaUsuarios** y **CentralPersistencia** es crucial para distribuir responsabilidades y reducir acoplamiento. Esto facilita la evolución futura del sistema, donde la lógica de persistencia puede cambiar (por ejemplo, al migrar a una base de datos diferente) sin afectar el resto de las funcionalidades. Adicionalmente al hacer pruebas es mas facil identificar de donde estan viniendo los problemas.
5. **Uso de ControladorUsuario y Sistema como Singleton**: Esta clase garantiza que solo exista una instancia responsable de gestionar a todos los usuarios, y LP. El patrón singleton asegura que no haya inconsistencias en el manejo de usuarios dentro del sistema y que toda la lógica de autenticación y registro esté centralizada en un único punto. Asimismo, este patrón permite centralizar todos los LP y actividades creadas, permitiendo clonar Actividades y encontrar LP con mayor facilidad.
6. **Calificadores especializados**: La decisión de tener calificadores especializados para diferentes tipos de actividades asegura que el comportamiento de calificación sea preciso y ajustado a las necesidades específicas de cada tipo de pregunta. Esto permite manejar eficientemente las diferencias entre un examen de preguntas abiertas y un quiz de opción múltiple.

Estas decisiones de diseño han sido tomadas con el objetivo de crear un sistema robusto, extensible y mantenible, permitiendo futuras adiciones de tipos de actividades, usuarios o métodos de calificación sin generar cambios disruptivos en el código existente.