

Documento de diseño (Entrega 2)

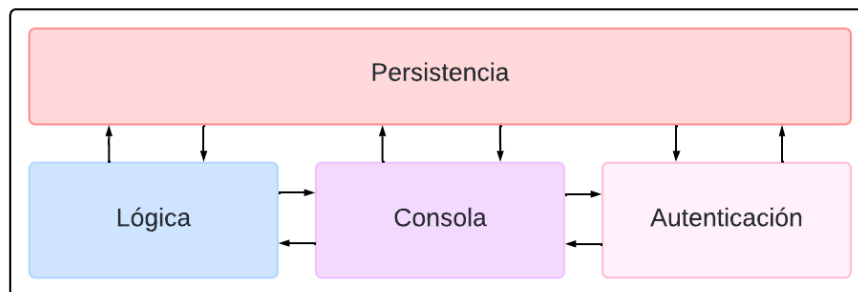
Daniela Martínez Rincón 202310127

María Inés Velásquez Acosta 202224325

Martín Riveira 202321727

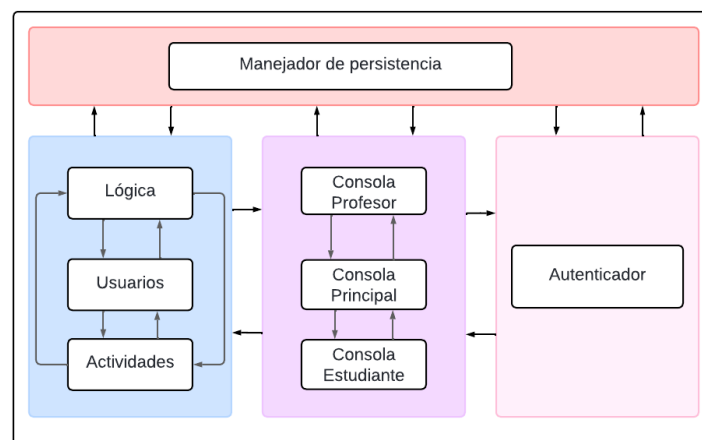
Descomposición del proyecto

En primer lugar, para poder entender el problema y diseñar una solución que fuera capaz de cumplir con todos los funcionamientos necesarios identificados en la fase de análisis se realizaron varias descomposiciones del sistema de Learning Paths. Primero se encontraron los siguientes cuatro grupos principales necesarios para cumplir con los requerimientos.



Estos grupos hacen referencia a que principalmente el sistema debe ser capaz de interactuar con el usuario por medio de la consola, debe poder manejar la información de y calcular porcentajes con la lógica, debe persistir para poder guardar la información de los usuarios existentes junto con sus actividades y Learning Paths y debe poder cumplir con autenticar el ingreso de los usuarios con la autenticación.

Al revisar cada uno de estos grupos y contrastando con las clases encontradas en el análisis se pudo encontrar una mayor descomposición de cada grupo.



Debe haber una clase o un grupo de clases que se encarguen de persistir la información de todo el sistema y también debe haber una clase encargada de realizar el registro y la autenticación de los usuarios. En cuanto a la consola, al momento de iniciar el sistema no se diferencia en que tipo lo está usando, sin embargo, al ingresar es necesario que el estudiante tenga unas opciones diferentes a las del profesor para poder así limitar las capacidades de ambos y que pueda así controlarse quien hace qué. Por otro lado, en la lógica se identificó que debe haber un grupo encargado de lo que se refiere a todas las actividades que se pueden crear y manejar en el sistema, un grupo encargado de especificar los perfiles de los usuarios (Profesor/Estudiante) y finalmente un grupo de clases encargado de realizar todos los cálculos de porcentajes de registrar y mantener la información de los Learning Paths generales en el sistema. No se determinó como necesario un grupo encargado de las excepciones, dado que en ningún momento es necesario realizar procesamiento demasiado complejo y los fallos en los procesos de la plataforma identificados se relacionan más con datos ingresados por el usuario, que pueden ser tratados con la clase Exception de Java.

Roles y componentes

De la descomposición y de la fase de análisis se definió que el proyecto va a tener una estructura centralizada en donde una clase tendrá el mayor control de la parte lógica del sistema, esto con el fin de poder simplificar la interacción entre los componentes de la plataforma y en un caso futuro poder acoplarla mejor al relacionarla con LMS. Considerando que el proyecto no requiere de una consola, se puede identificar que se compone por tres estructuras principales: el sistema (la lógica), la persistencia y la autenticación.

En este sentido, el sistema es el encargado de realizar todos los métodos u opciones que tiene cada tipo de usuario. Será el encargado de crear objetos, manejar la información entre estos, realizar cálculos, entre otros. La persistencia se encarga de procesar toda la información y guardarla en archivos de texto y al mismo tiempo será capaz de leer la información guardada anteriormente para que el sistema sea capaz de funcionar con información previa a la inicialización de este. Finalmente, la autenticación debe ser capaz de manejar el ingreso de los usuarios, diferenciando su tipo y registrando estos. Es decir, debe ser capaz de permitir el ingreso correcto de los usuarios a la plataforma,

Con base en esto, se identificó que la aplicación tendrá diferentes clases encargadas de estructurar, mantener información y realizar servicios o transformar información. Por esta razón se realizó la siguiente asignación de los posibles candidatos de la plataforma, considerando los identificados en la descomposición y los del dominio:

Rol	Nombre	Explicación
-----	--------	-------------

structurer	Sistema	Es el elemento principal de la plataforma, que es capaz de conocer todos los usuarios, Learning Paths y actividades, así como sus relaciones.
interfacer	ManejadorDePersistencia	Es el elemento identificado para transformar la información de la plataforma en archivos de texto y así mismo, cargar los datos de este tipo de archivo cuando sea llamado durante el funcionamiento de la plataforma.
structurer	Progreso	Este es el elemento que conoce todos los elementos relacionados con un estudiante y un Learning Path específico. Es capaz de manejar la información de esta relación, así como es responsable de establecer la conexión entre estos dos elementos.
Information holder	Usuario	Tanto este como sus herederos, mantienen solo información básica de los usuarios y sus tipos.
Service provider	Autenticador	Es la clase encargada de registrar a todos los usuarios y de permitirles el uso de la plataforma, por lo tanto, debe poder manejar correctamente la información de todos los usuarios e identificar si los datos ingresados por el usuario son correctos.
Information holder	LearningPath	Es la clase que mantiene la información de un Learning Path con todos sus componentes, pero no realiza ninguna función compleja.

Information holder	Actividad	Es la clase encargada de la información de cada actividad que al igual que sus herederos no implementa ninguna lógica compleja.
Information holder	PreguntaCerrada	Es el elemento de mantener y entregar la información de las preguntas cerradas, no implementa ninguna función lógica complicada.
Information holder	Opcion	Es el elemento de mantener y entregar la información de las opciones de las preguntas cerradas, no implementa ninguna función lógica complicada.
Information holder	PreguntaAbierta	Es el elemento de mantener y entregar la información de las preguntas abiertas, no implementa ninguna función lógica complicada.
Information holder	Feedback	Es el elemento de mantener y entregar la información de los feedbacks de las actividades, no implementa ninguna función lógica complicada.

Responsabilidades

Considerando los requerimientos funcionales, se identificaron las siguientes responsabilidades básicas y se le asignaron a cada uno de los componentes.

No.	Responsabilidad	Componente
1	Diferenciar entre los usuarios para permitirles utilizar sus funciones	Consola
2	Recibir la información ingresada por el usuario	Consola

3	Mostrar los resultados de las opciones utilizadas a los usuarios	Consola
4	Cumplir con las peticiones del usuario	Sistema
5	Conocer toda la información de la plataforma para realizar los procedimientos de la plataforma	Sistema
6	Crear los elementos del sistema (objetos)	Sistema
7	Crear los usuarios y registrales	Autenticador
8	Conocer toda la información relacionada con la actividad del usuario con un Learning Path en la plataforma	Progreso
9	Conseguir la información para realizar los procesos de la plataforma	Sistema
10	Almacenar la información	ManejadorDePersistencia
11	Cargar la información de archivos existentes	ManejadorDePersistencia
12	Consultar la base de datos y permitirles a los usuarios ingresar	Autenticador

En general, estas fueron las responsabilidades principales encontradas en el funcionamiento de la plataforma, en las cuales se agrupan las responsabilidades como crear actividad, registrar usuario, crear Learning Path, etc.

Colaboraciones

Con respecto a las distintas interacciones de los elementos de la plataforma, se identificó que hay ciertos componentes que se relacionan estrechamente, sobre todo los Information holders que son manejados por el sistema y por el progreso. Por esta razón tomando el diagrama de descomposición más específico presentado anteriormente, se considera que los usuarios, las actividades y la lógica deben ir en un solo paquete, dado que la relación entre estos es la que compone todas las actividades y todos los procesos de la plataforma, es necesario estar en continuo contacto entre estas clases dado que deben transmitirse información frecuentemente para cumplir con las funcionalidades. Por otro lado, se encuentra la consola con los elementos de las consolas de los diferentes tipos de usuarios que se encargaría si fuera necesario implementarla de la interacción entre el usuario y el sistema. En cuanto a la

autenticación se observó que este elemento se utiliza pocas veces y solamente necesita relacionarse con la lógica al momento de ingresar o registrarse, por lo que no es necesario el paso de información continuo y tampoco requiere de mucha información de la lógica solamente de los usuarios. La persistencia por otro lado se encuentra aparte, dado que si bien se encuentra relacionada con todos los elementos de la plataforma solo se necesita en momentos específicos de guardar y cargar información, no es necesaria en todos los procesos de la plataforma. Cabe aclarar que se decidió guardar la información en cada etapa de los procesos para que en caso de que haya una falla en el entorno, no se pierda la información del usuario.

Diagramas de secuencia

Con el contexto presentado, se plantearon los siguientes diagramas de secuencia de las funcionalidades más complejas para entender mejor la colaboración entre los elementos de la plataforma.

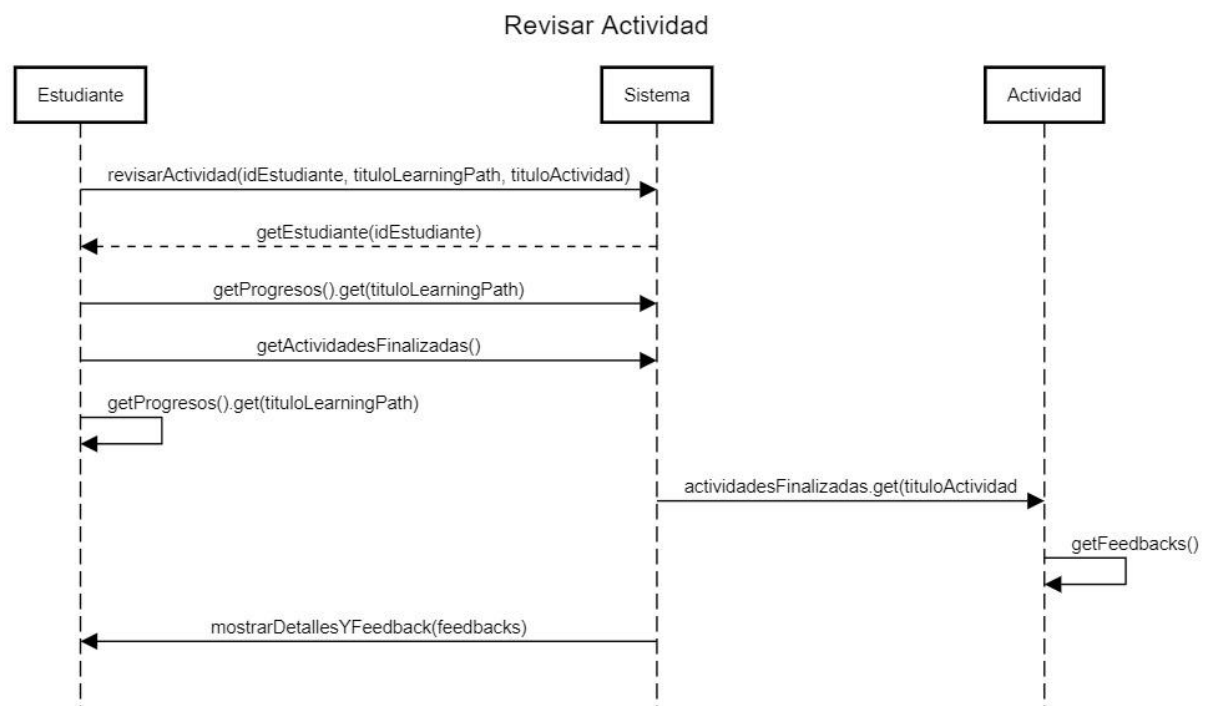
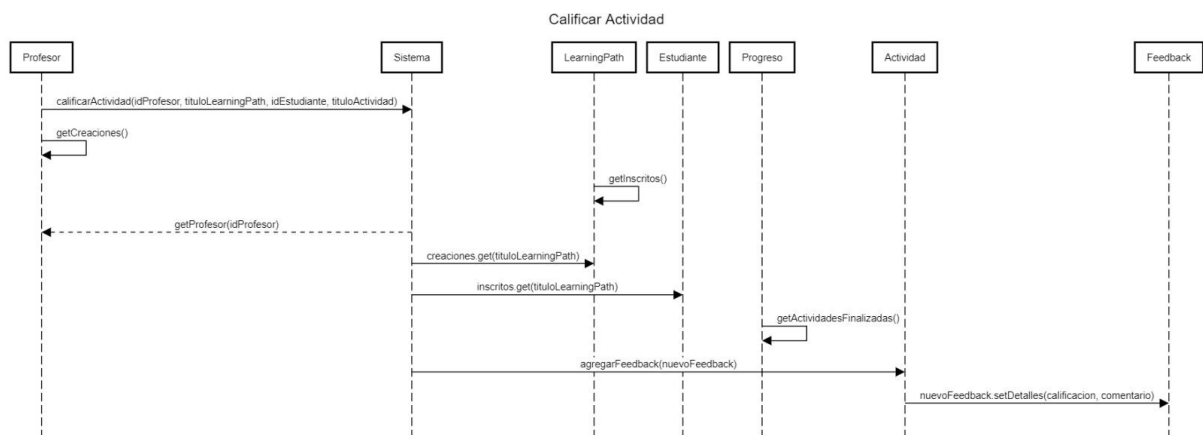
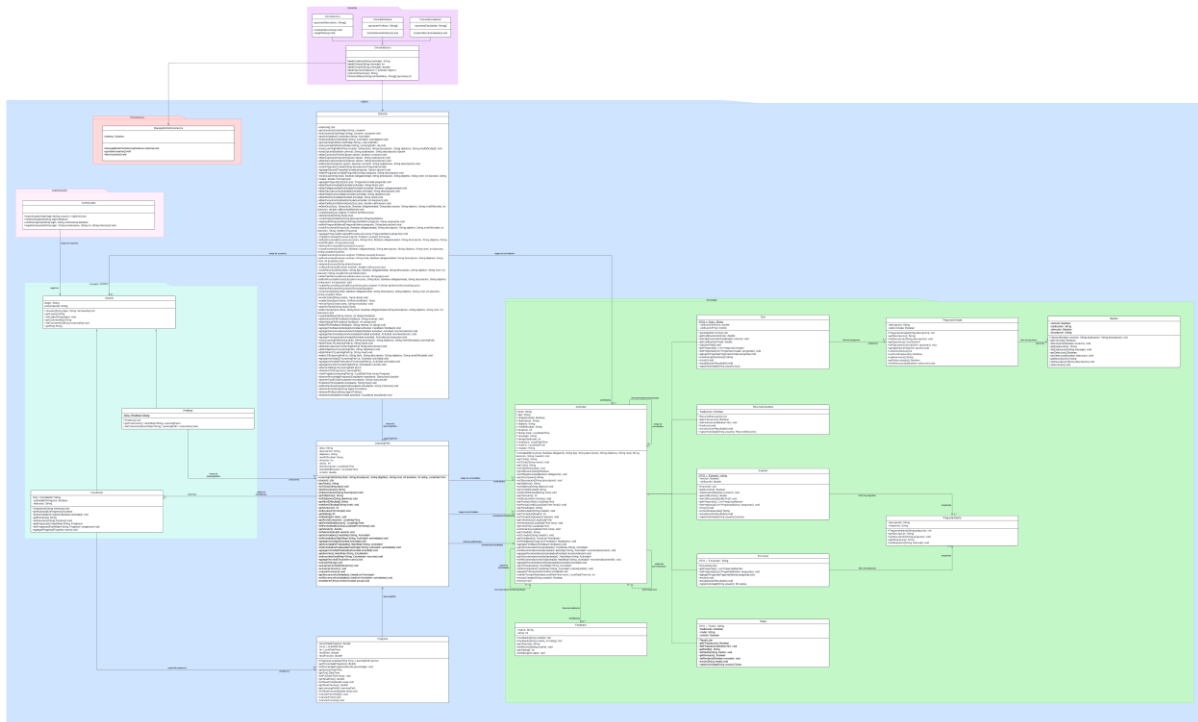


Diagrama de clases de diseño

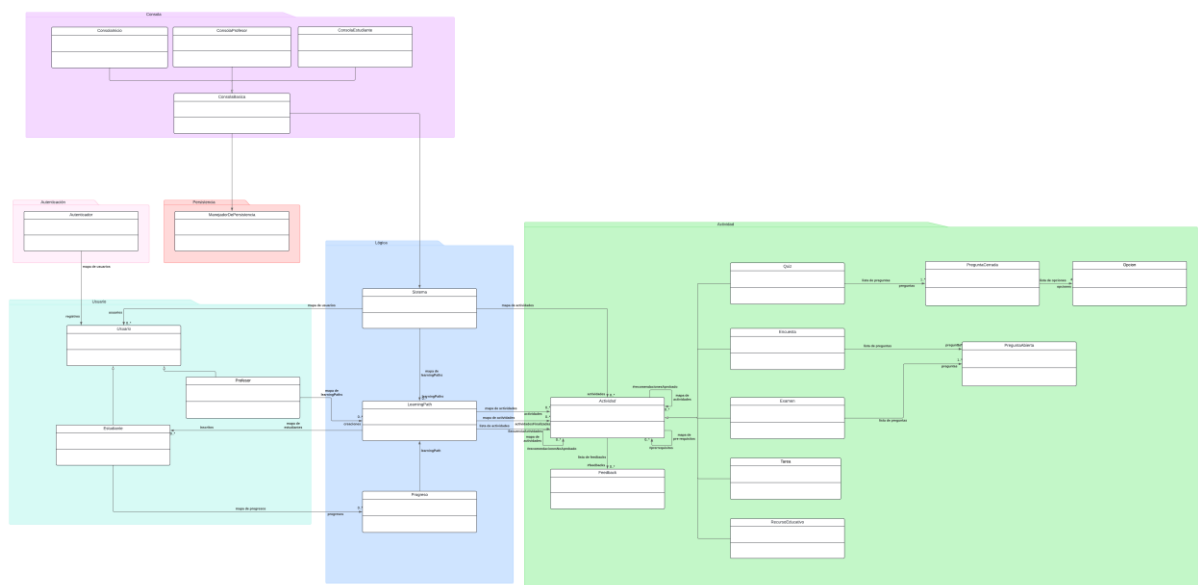
Así mismo, se diseñó el siguiente diagrama de la plataforma que es el que será implementado.



Que también se adjunta a esta entrega para mejor visualización.

Diagrama de clases de alto nivel

Considerando el diagrama anterior, se implementó un diagrama de clases de alto nivel para entender mejor el relacionamiento entre las diferentes clases de la plataforma.



Este diagrama también se adjuntó a la entrega para mejor visualización.