

Documento de Diseño

Funciones Críticas

- Crear un perfil en la plataforma
 - El usuario puede crear un perfil para acceder a las funcionalidades de la aplicación. El perfil necesita un nombre de usuario y contraseña dados por el usuario y el tipo de perfil que se va a crear. El perfil es guardado en el sistema para poder iniciar sesión después
- Iniciar sesión
 - El usuario ingresa su nombre de usuario y contraseña y el sistema autentica el perfil. Si son correctas, el sistema inicia la sesión en la plataforma. Si no, le manda un mensaje de error y no se realiza otra acción.
- Crear un Learning Path (solo para profesores)
 - El profesor puede crear un Learning Path, ingresando los detalles necesarios para darle sus atributos. El Learning Path es guardado en el sistema. Puede ser accedido por cualquier usuario en la plataforma pero solo puede ser modificado por el profesor que lo creó.
- Crear una actividad (solo para profesores)
 - El profesor puede crear una actividad, definiendo su tipo y sus atributos adicionales, que será guardada en la plataforma y en su perfil de usuario.
- Clonar una actividad (solo para profesores)
 - El profesor puede clonar una actividad. El sistema debe permitirle al profesor seleccionar una actividad

creada por otro profesor y colocarla en sus propias actividades

- Añadir actividades a Learning Paths (solo para profesores)
 - El profesor puede añadir sus actividades a sus Learning Paths. El sistema debe permitirle modificar los learning paths y debe guardar las modificaciones.
- Inscribirse a un Learning Path (solo para estudiantes)
 - El estudiante puede inscribirse a un learning path. El learning path. Se crea un progreso de dicho learning path dentro del estudiante
- Completar una actividad
 - El estudiante puede comenzar y marcar como completada. Cuando se completa la actividad, el progreso del estudiante dentro del Learning Path asociado cambia y pone la actividad en el grupo de actividades completadas dentro del progreso
- Dejar un review
 - El usuario puede dejar un review en la actividad. El sistema añade los reviews a la actividad y los guarda.

Breve Descripción de las clases:

Clases del dominio del problema:

Learning path:

- Clase para modelar los Learning Paths de la plataforma.
- Las instancias de Learning Paths son creadas por los profesores
- Las instancias se guardan en la clase Manejo De Datos dentro de un HashMap (Título del Learning Path, Learning Path) para ser accedidos y visualizados por todos los usuarios de la plataforma.

- Dentro de la clase se mantiene una lista de estudiantes inscritos y un HashMap (Nombre del estudiante, progreso) de los progresos de cada estudiante

Actividad:

- Clase para modelar las Actividades de los Learning Paths
- Las instancias de Actividades son creadas por los profesores
- Cada actividad tiene un ID único generado cuando se crean o cuando son clonadas por profesores
- Solo los profesores que crearon la actividad la pueden editar y usar para sus propios learning paths
- Las Actividades se pueden clonar:
 - Cuando los profesores las clonan, el ID y el autor de la actividad cambian, pero los otros detalles permanecen iguales hasta que el profesor los modifique
 - Cuando un estudiante se inscribe a un learning path, las instancias de las actividades del path se clonan dentro de su progreso pero su ID no cambia. Esto es para que cuando hagan una actividad se modifique el clon, pero se puede seguir identificando qué actividad era originalmente

Hay diferentes tipos de actividades:

- Recurso Educativo:
 - Incluye el contenido del recurso, el tipo de recurso y el enlace al recurso
 - Se puede marcar como completada (Exitosa) sin requisitos adicionales
- Tarea:
 - Modela una tarea. Puede ser marcada como Exitosa (cuando el profesor la califica y la aprueba), No Exitosa (cuando el profesor la reprueba) y Enviada (Cuando el estudiante la envía)

- Cuando es marcada como completa por el estudiante, este debe indicar el medio de entrega
- Prueba
 - Es una clase abstracta que modela una prueba.
 - Hay tres tipos de prueba (cada una es una clase): Encuesta, Exámen y Quiz
 - Para marcarla como completada, la prueba tiene que ser respondida por el estudiante
 - Quiz: Los quices tienen preguntas de opción múltiple. Una vez respondido, se califican automáticamente. Si la calificación obtenida es menor que la calificación mínima, se marca como No Exitosa, de lo contrario, Exitosa.
 - Examen: Los exámenes tienen preguntas abiertas. Una vez el estudiante marca como completada la prueba, su estado pasa a ser Enviado. Después de que el profesor le asigne una calificación, será marcado como Exitosa o No Exitosa
 - Encuesta: Las encuestas tienen preguntas abiertas, pero no son calificadas para marcarlas como completadas. Una vez respondidas, se asigna por default 5 a la calificación y se marca como Exitosa

Usuario: <Controller>

- Clase abstracta para modelar los usuarios de la plataforma
- De ella heredan Estudiante y Profesor
- Contiene los detalles básicos de los perfiles de usuario (login, contraseña, correo, tipo)
- Tiene métodos compartidos que todo usuario puede hacer, como dejar reviews en una actividad

Estudiante: <Controller>

- Clase para modelar a los estudiantes
- Un estudiante puede seleccionar un Learning Path e inscribirse a este

- El estudiante no tiene como tal una interacción directa con el Learning Path y sus actividades, sino que tiene un progreso asociado al path que le permite completar actividades y avanzar en el Learning Path
- Un estudiante puede tener iniciada solo una actividad a la vez por cada Learning Path y puede marcarla como completada una vez cumpla con todos los requisitos para ello (Ej: si es una prueba, tiene que haberla respondido, si es una tarea, tiene que haberla mandado, etc)

Profesor: <Controller>

- Clase para modelar a los profesores
- Los profesores son los únicos que pueden crear Learning Paths y actividades
- Los profesores pueden clonar actividades de otros profesores y guardarlas en su perfil
- Pueden añadir actividades suyas a los Learning Paths que ellos crearon
- Un profesor puede tener varios Learning Paths creados
- Los profesores pueden consultar los ratings y reviews dejados en sus actividades
- Los profesores pueden consultar el progreso de sus estudiantes
- Los profesores pueden calificar y marcar como Exitosa o No Exitosa una actividad marcada como completada por el estudiante
 - Si la actividad es marcada como No Exitosa, el profesor puede devolver la actividad a la lista de actividades pendientes dentro del progreso del estudiante

Progreso: <Service Provider>

- Esta clase modela el progreso de un estudiante en un Learning Path
- Tiene un único Learning Path y Estudiante asociado

- Dentro de la clase, hay unas estructuras de HashMap y ArrayList que guardan copias de las actividades del Learning Path para que el estudiante empiece y marque como completadas
- Aquí se maneja todo lo relacionado con los avances del estudiante dentro de un Learning Path
 - Empezar actividad
 - Completar actividad (el estudiante hace la actividad)
 - Marcar una actividad como no completada (funcionalidad para ser usada exclusivamente por los profesores)

Review:

- Modela las reviews dejadas por usuarios en actividades
- Una review siempre debe tener un rating dado por el usuario
- Solo los usuarios pueden crear instancias de la clase para añadirlas a una actividad
- Solo el autor del review puede eliminar el review de una actividad

Sobre la persistencia

Para la persistencia se inició cargando desde lo más pequeño hasta lo más grande. Primero se carga las reviews en un hash map, donde la llave es el contenido y el valor la recién. Segundo, se cargan las preguntas en un hash map, donde la llave es el enunciado y el valor la pregunta. Tercero, se cargan las actividades, cuando los atributos de actividades son review o pregunta, no se guarda todo el objeto, solo la llave. Al cargar actividad también se usan los mapas de review y pregunta para crear los objetos. Actividad se guarda en un hash map donde la llave es el ID de la actividad y el valor es la actividad. Cuarto, se cargan los learning Paths, para los atributos antes mencionados solo se guarda la llave, y al cargarlo se busca en los mapas para crear el Learning Path. Este se guarda en un hash map, donde la

llave es el nombre del Learnig Path. Finalmente, se cargan los usuarios, similarmente a lo antes mencionado, solo guarda las referencias a otros objetos, esto para que en el JSON no queden clases de clases, y mantener toda la información organizada. El usuario también se guarda en un hash map donde la llave es una lista, con el usuario y la contraseña.

Cada vez que se crea una nueva instancia de las clases antes mencionadas, es necesario agregarlos a su respectivo mapa, y guardar el mapa. Esto para que toda la información está actualizada conforme se agregan datos. Todos los mapas, la carga y el guardado de datos se manejan desde la clase Manejo de Datos, cuyos atributos son los mapas antes mencionados y en el cual están todos los métodos necesarios para agregar y buscar la información requerida.

Clases adicionales para la funcionalidad del sistema

Persistencia Usuarios:

- Clase encargada de guardar y cargar la información de los usuarios a/desde un archivo JSON

Persistencia Learning Paths:

- Clase encargada de guardar y cargar la información de los Learning Paths a/desde un archivo JSON

Persistencia Actividades:

- Clase encargada de guardar y cargar la información de las Actividades a/desde un archivo JSON

Persistencia Reviews:

- Clase encargada de guardar y cargar la información de las reviews a/desde un archivo JSON

Persistencia Progreso:

- Clase encargada de guardar y cargar la información de los progresos a/desde un archivo JSON

Persistencia Preguntas:

- Clase encargada de guardar y cargar la información de las Preguntas a/desde un archivo JSON

Manejo de Datos: <Information Holder>

- Clase encargada del manejo de datos de la aplicación
- Se encarga de guardar y/o eliminar los objetos de actividades, Learning Paths y usuarios de la plataforma cada vez que se crean o se eliminan instancias de cada clase

Consola: <Interfacer>

- Clase encargada de servir como interfaz para conectar las interacciones de los usuarios con el resto de la aplicación

Excepciones:

Respuestas Inconsistentes Prueba Exception:

- Excepción que maneja cuando un estudiante responde una prueba y terminan habiendo más o menos respuestas de las esperadas

Ya Existe Actividad En Progreso Exception:

- Excepción que maneja cuando el sistema trata de marcar como completada una actividad en progreso que no ha sido iniciada por el estudiante

