

Entrega 1- Análisis del proyecto

Lucas Valbuena Leon 202311538

Juan Goyeneche Saenz 202320863

Alejandro Garcia Rojas 202122516

1. Requerimientos funcionales (Historias de usuario):

- Como profesor quiero:
 - Poder crear *Learning Paths* y actividades.
 - Poder modificar los *Learning Paths* que haya creado.
 - Poder clonar actividades y editar su copia.
 - Poder ver el progreso de los estudiantes en los *Learning Paths*.
 - poder ver reseñas y comentarios en los *Learning Paths*.
- Como estudiante quiero:
 - Poder ver e inscribirme en *Learning Paths*. De igual forma, en caso de inscribirme, mostrar la fecha de inicio del path, el tiempo dedicado por actividad, tasa de éxito y cuando fueron completadas.
 - Poder elegir en qué actividad quiero empezar o continuar.
 - Poder dar *feedback* o *rating* a una actividad.
- Como administrador de la plataforma quiero:
 - Poder permitir el registro de estudiantes y profesores mediante un sistema de autenticación.
 - Poder rastrear las actividades y progreso del estudiante. El progreso debe mostrar un porcentaje de las actividades obligatorias completadas con éxito.
 - Mostrar el *feedback* y el *rating* de las actividades.
 - Permitir que los profesores marquen actividades como "obligatorias" u "opcionales".
 - Mostrar el rating promedio de los *Learning Paths*.
 - Mostrar el *rating* en forma de un número decimal de 0 a 5.
 - Recomendar las actividades futuras, teniendo en cuenta el desempeño y actividades previas de los estudiantes.
 - Advertir a los estudiantes si están intentando realizar una actividad sin haber completado las actividades previas, explicando por qué no es recomendable.

- No permitir avanzar a la lectura siguiente si el quiz fue "insatisfactorio".
- Recomendar actividades futuras a los estudiantes basándose en su desempeño y en las actividades previas completadas.

2. Requerimientos no funcionales (modelo de calidad ISO 25010):

- Rendimiento:
 - Debe tener capacidad para tener múltiples estudiantes conectados.
- Seguridad:
 - Los usuarios deben tener un login y contraseña para acceder al sistema.
- Usabilidad:
 - Para la fase 1 no se requiere implementar una interfaz gráfica.
- Mantenibilidad:
 - El sistema debe ser capaz de ser editado para añadir futuras mejoras y correcciones dependiendo de los resultados de cada etapa.
- Compatibilidad:
 - El sistema debe ser capaz de integrar un LMS (*Learning Management Systems*) como bloque neón en el futuro.
- Escalabilidad:
 - El sistema debe permitir la adición de nuevos tipos de actividades y Learning Paths sin afectar la estructura existente.

3. Restricciones:

Construcción de software:

- El sistema debe ser escrito en Java.
- En la etapa 1 se debe implementar la lógica.
- En la etapa 2, se debe permitir la interacción del usuario mediante una interfaz gráfica
- El tiempo para culminar el proyecto va hasta la finalización del semestre (entrega análisis semana 8, entrega diseño semana 10).

Operación del software:

- Almacenar la información en archivos planos o binarios (persistencia).

4. Reglas del dominio:

- Solo un profesor puede crear o modificar un *Learning Path*.
- Todo usuario deben tener un usuario y contraseña
- Solo el creador original de una actividad puede modificarla.
- Solo se puede realizar la encuesta después de haber hecho la actividad.
- El quiz basado en la lectura debe cerrarse habiendo pasado una hora determinada.
- Si el quiz relacionado con una actividad previa fue calificado como insatisfactorio se deberá volver a la lectura.
- Las actividades pueden estar marcadas como obligatorias u opcionales.