

Estereotipos asignados a las clases:

- Usuario (Information holder):
 - **Motivo:** Almacena información básica de los usuarios como nombre, contraseña y tipo, sin realizar operaciones complejas.
- Pregunta (Information holder):
 - **Motivo:** Guarda datos específicos de cada pregunta, como el enunciado, opciones y respuestas correctas, sin realizar lógicas avanzadas.
- Recurso **Educativo** (Information holder):
 - **Motivo:** Representa un recurso educativo con detalles como tipo de recurso y URL, limitándose a almacenar y proporcionar esta información.
- Sección (Information holder):
 - **Motivo:** Guarda información sobre las secciones y posteriormente mostrárselas al usuario dentro de una tarea o actividad, como su número, título y descripción. Se centra en mantener datos estáticos.
- Metadato (Information holder):
 - **Motivo:** Almacena información de creación y modificación de un objeto como fecha y versión, sin gestionar otras clases.
- **LearningPath** (Structurer):
 - **Motivo:** Organiza las actividades, estudiantes inscritos y objetivos del camino de aprendizaje, manteniendo una estructura clara de las relaciones de estas entidades.
- Avance (Structurer):
 - **Motivo:** Estructura el progreso de un estudiante en el camino de aprendizaje, relacionando actividades completadas y pendientes, así como porcentajes y tasa de éxito.
- **ManejadorLP** (Service provider):
 - **Motivo:** Gestiona el conjunto de LearningPaths disponibles, permitiendo la creación y modificación de estos caminos de aprendizaje. Provee un servicio central para la gestión de LearningPaths.

- **PersistenciaLP** (Service provider):

- **Motivo:** Facilita la carga y salvado de datos de los LearningPaths, gestionando la interacción con el almacenamiento de datos de manera autónoma y compleja.

- **ManejadorSesion** (Coordinator):

- **Motivo:** Coordina la creación de usuarios y el inicio de sesiones, delegando la lógica de negocio a otras clases sin realizar operaciones complejas por sí mismo.

- **LPRS** (Controller):

- **Motivo:** Controla la ejecución principal del sistema y toma decisiones sobre la carga inicial y la configuración de los datos, actuando como un núcleo de control.

- **PersistenciaUsuario** (Interfacer):

- **Motivo:** Maneja la persistencia de los datos de usuario, actuando como una interfaz entre el sistema y el almacenamiento externo, transformando la información según sea necesario para guardarla o recuperarla.

- **ExamenRealizable, EncuestaRealizable, TareaRealizable, QuizRealizable** (Service provider):

- **Motivo:** Proveen servicios para realizar, calificar y enviar actividades específicas (Examen, Encuesta, Tarea, Quiz) para cada estudiante, encapsulando lógica compleja y específica para cada tipo de actividad.

- **ActividadRealizable** (Service provider):

- **Motivo:** Encapsula las actividades a realizar por un estudiante, incluyendo métodos para realizar, guardar y calificar actividades, proporcionando servicios específicos para el progreso en el Learning Path.

- **Examen, Encuesta, Quiz, Tarea** (Information holder/Service provider):

- **Motivo:** Estas clases contienen la estructura de datos para sus respectivas actividades (Information holder). También actúan como proveedores de servicios al crear instancias de actividades realizables para los estudiantes, por lo cual pueden considerarse como Service providers.

Grupo 6

Lista De requerimientos funcionales:

1. Creación de usuario
2. Autenticación de usuario (Inicio de sesión)
3. Creación de un learning path por parte de un profesor
4. Crear y agregar actividades a un learning path
5. Edición de un learning path
6. Calificación de una actividad
7. Clonar un learning path
8. Reseñar y calificar una actividad como profesor
9. Consultar el avance de un estudiante dentro del learning path
10. Inscripción a un learning path
11. Visualización de avances por parte de estudiante
12. Revisar un recurso educativo
13. Realizar una tarea
14. Realizar un quiz
15. Realizar un examen
16. Completar una encuesta

Descripción de implementación de Requerimientos Funcionales.

1. Creación de usuario

Pedirle al usuario un Nombre de usuario y contraseña (String Usuario, String contraseña)

Pedirle al usuario el tipo de usuario que quiere crear (Estudiante o Profesor)

Utilizar el manejadorSesion para crear el usuario usando
manejadorSesion.crearUsuario(usuario, contrasenia)

Caso 1: Usuario no repetido:

- Agregar al HashMap de manejadorSesion usando Usuario.agregarUsuario(Usuario).
- Mandar mensaje de confirmación para notificar la finalización del proceso.

Caso 2: Usuario repetido:

- Utilizar una excepción para notificar al usuario del usuario repetido y pedir la creación de otro usuario.
- Intentar la creación de usuario de nuevo.

2. Autenticación de usuario (Inicio de sesión)

Grupo 6

En caso de que se inicie una nueva instancia de la aplicación, cargar los usuarios a partir de la persistencia los usuarios anteriormente creados (`Persistencia.cargarUsuario()`)

Pedir el usuario y la contraseña al usuario por medio de la consola.

Usar el método estático `Usuario.iniciarSesion(String Usuario, String Contraseña)` para verificar la existencia del usuario y que la contraseña sea válida.

Caso 1: Usuario no encontrado en el sistema

- Utilizar una excepción notificando al usuario que el nombre de usuario ingresado no se ha encontrado en el sistema
- Dar la opción de reintentar

Caso 2: Usuario encontrado, contraseña incorrecta:

- Utilizar una excepción notificando al usuario que la contraseña ingresada no se ha encontrado en el sistema
- Dar la opción de reintentar

Caso 3: Usuario encontrado, contraseña correcta:

- Enviar un mensaje de confirmación al usuario de un inicio de sesión correcto.
- Asignar la consola que corresponde al tipo de usuario

3. Creación de un learning path por parte de un profesor

Hacer autenticación de usuario dado un usuario y contraseña (Requerimiento 2), y verificar que el tipo de usuario sea profesor.

Pedirle al profesor el título, descripción, Nivel de dificultad, y un número no determinado de objetivos

Usar `profesor.crearLearningPath(titulo, descripcion, nivelDificultad,objetivos)`

Enviar un mensaje de confirmación al usuario sobre la creación exitosa del nuevo learning path junto con su ID asignado.

Grupo 6

4. Crear y agregar actividades a un Learning Path.

Hacer autenticación de usuario dado un usuario y contraseña (Requerimiento 2), y verificar que el tipo de usuario sea profesor.

Mostrar los learning paths creados por el profesor y seleccionar un learning path (Si no se ha creado, se hace en el requerimiento 3)

Se le pide al usuario el tipo de actividad que desea realizar (Recurso Educativo, Quiz, Examen, Encuesta o Tarea)

Se le pide la información de la actividad correspondiente

Con el Learning Path seleccionado, dependiendo del tipo de actividad a realizar, se usará crearRecurso, crearQuiz, crearExamen, crearEncuesta o crearTarea. (en general los datos que se piden son titulo: String, descripcion: String, nivelDificultad: String, objetivos: List<String>, duracion: double, rating: double, numCalificaciones: int)

5. Edición de un learning path

Hacer autenticación de usuario dado un usuario y contraseña (Requerimiento 2), y verificar que el tipo de usuario sea profesor.

Mostrar los learning paths creados por el profesor y seleccionar un learning path (Si no se ha creado, se hace en el requerimiento 3)

Se le pide al usuario cual learning path quiere editar.

Luego se le piden los datos de un learning path normal y se cambia lo quiera ser editado. Se usa LearningPath.editarLearningPath(String titulo, String descripcion, String nivelDificultad, ArrayList<String> objetivos, Profesor profesorCreador).

Se pide en cada caso todos los datos y se mantienen los que no se quieren cambiar.

6. Calificación de una actividad

Hacer autenticación de usuario dado un usuario y contraseña (Requerimiento 2)

Verificar el tipo de usuario sea profesor, si lo es, mostrar la consola de profesor.

Se obtienen las actividades pendientes que tiene el profesor usando el método getActividadesPendientes(), se selecciona una y retorna la actividad realizable que hizo el estudiante, sobre la cual se hace actividadRealizable.calificarActividad() los datos de las calificaciones de cada trabajo dependiendo del tipo de actividad.

Grupo 6

7. Clonar un learning path

Hacer autenticación de usuario dado un usuario y contraseña (Requerimiento 2)

Verificar el tipo de usuario sea profesor, si lo es, mostrar la consola de profesor.

Mostrar los learning paths creados por los demás profesores.

Se clona usando `profesor.clonarLearningPath(String ID)`.

Luego queda guardado en la cuenta del profesor (y luego con el requerimiento 5 se edita el learning path clonado)

8. Reseñar una actividad

Hacer autenticación de usuario dado un usuario y contraseña (Requerimiento 2)

Reseñar la actividad usando `actividad.addResenia(Resenia resenia)`.

9. Consultar el avance de un estudiante dentro del learning path

Hacer autenticación de usuario dado un usuario y contraseña (Requerimiento 2)

Verificar el tipo de usuario sea profesor, si lo es, mostrar la consola de profesor.

Mostrar los learning paths creados por los demás profesores.

Mirar la lista de estudiantes inscritos al learning path y preguntar la información del estudiante para ver el progreso de avance del estudiante.

Usando `estudiante.getAvancesEstudiante()`.

10. Inscripción a un learning path

Hacer autenticación de usuario dado un usuario y contraseña (Requerimiento 2)

Verificar el tipo de usuario sea estudiante, si lo es, mostrar la consola de estudiante.

Ver el catálogo de learning paths disponibles y luego pedir al usuario la información de que learning path quiere inscribirse guardándolo en su perfil.

Usando `Estudiante.inscribirLearningPath(String ID)`.

Grupo 6

11. Visualización de avances por parte de estudiante

Hacer autenticación de usuario dado un usuario y contraseña (Requerimiento 2).

Verificar el tipo de usuario sea estudiante, si lo es, mostrar la consola de estudiante.

Revisar el catálogo de los learning paths al cual está inscrito el estudiante y pedir la información acerca del learning path que se quiere llegar a saber sobre el avance.

Usando `Estudiante.getAvancesEstudiante()`

12. Revisar un recurso educativo

Hacer autenticación de usuario dado un usuario y contraseña (Requerimiento 2).

Verificar el tipo de usuario sea estudiante, si lo es, mostrar la consola de estudiante.

Revisar el catálogo de los learning paths al cual está inscrito el estudiante y pedir la información acerca del learning path que se quiere llegar a saber sobre las actividades del learning path.

Luego revisar los tipos de actividades disponibles en el catálogo de learning path y elegir algún recurso educativo.

Luego mostrar los materiales didácticos disponibles en el recurso educativo.

Usando `RecursoRealizable.realizarActividad()`.

13. Realizar una tarea

Hacer autenticación de usuario dado un usuario y contraseña (Requerimiento 2).

Verificar el tipo de usuario sea estudiante, si lo es, mostrar la consola de estudiante.

Revisar el catálogo de los learning paths al cual está inscrito el estudiante y pedir la información acerca del learning path que se quiere llegar a saber sobre las actividades del learning path.

Luego revisar los tipos de actividades disponibles en el catálogo de learning path y elegir alguna tarea.

Luego poder hacer la tarea con los enunciados que se han propuesto por el profesor que ha creado el learning path.

Usando `TareaRealizable.realizarActividad()`.

Grupo 6

14. Realizar un quiz

Hacer autenticación de usuario dado un usuario y contraseña (Requerimiento 2).

Verificar el tipo de usuario sea estudiante, si lo es, mostrar la consola de estudiante.

Revisar el catálogo de los learning paths al cual está inscrito el estudiante y pedir la información acerca del learning path que se quiere llegar a saber sobre las actividades del learning path.

Luego revisar los tipos de actividades disponibles en el catálogo de learning path y elegir algún quiz.

Luego empezar por pedir los datos de alguna de las cuatro opciones si es de pregunta cerrada o pedir la información si es pregunta abierta y finalmente dejarlo en calificación pendiente.

Usando `QuizRealizable.realizarActividad()`.

15. Realizar un examen

Hacer autenticación de usuario dado un usuario y contraseña (Requerimiento 2).

Verificar el tipo de usuario sea estudiante, si lo es, mostrar la consola de estudiante.

Revisar el catálogo de los learning paths al cual está inscrito el estudiante y pedir la información acerca del learning path que se quiere llegar a saber sobre las actividades del learning path.

Luego revisar los tipos de actividades disponibles en el catálogo de learning path y elegir algún examen.

Luego empezar por pedir los datos de alguna de las cuatro opciones si es de pregunta cerrada o pedir la información si es pregunta abierta y finalmente dejarlo en calificación pendiente (Aunque tiene un tiempo límite para ser resuelto).

Usando `ExamenRealizable.realizarActividad()`.

16. Completar una encuesta

Hacer autenticación de usuario dado un usuario y contraseña (Requerimiento 2).

Verificar el tipo de usuario sea estudiante, si lo es, mostrar la consola de estudiante.

Grupo 6

Revisar el catálogo de los learning paths al cual está inscrito el estudiante y pedir la información acerca del learning path que se quiere llegar a saber sobre las actividades del learning path.

Luego revisar los tipos de actividades disponibles en el catálogo de learning path y elegir alguna encuesta.

Luego pedir la información si es pregunta abierta y finalmente dejarlo en entregado.

Usando `EncuestaRealizable.realizarActividad()`.