**Diseño del Programa**

**1. Introducción**

En este trabajo se presenta el diseño del programa. También se detallan las decisiones clave tomadas durante el desarrollo y sus justificaciones, con diagramas pertinentes que ilustran la arquitectura general y específica del sistema.

**Objetivos del Sistema**

Está creado para la administración de actividades didácticas, es decir, permite a los docentes crear y asignar actividades y a los estudiantes consultarlas y realizarlas con seguimiento automatizado del progreso mediante LearningPaths.

**2. Diagrama de Clases de Todo el Sistema**

El diagrama contiene todas las clases del sistema, incluyendo la lógica del dominio y la interfaz gráfica, con sus relaciones, atributos y métodos principales.Imagen que contiene Diagrama

Descripción generada automáticamente

Por el tamaño del diagrama no se alcanza a ver bien, por lo que se recomienda ver el archivo adjunto DiagramaUmlCompleto.

**3. Diagrama de Clases de la Interfaz (GUI)**

Utiliza Swing como biblioteca gráfica, organizando las ventanas en una jerarquía modular y bien estructurada. A continuación se muestra un diagrama específico para las clases relacionadas con la GUI, centrándose en las ventanas y sus interacciones con la lógica.Diagrama

Descripción generada automáticamente

Por el tamaño del diagrama no se alcanza a ver bien, por lo que se recomienda ver el archivo adjunto DiagramaGui.

**4. Decisiones Clave de Diseño**

**4.1 Separación de la lógica y la interfaz**

La lógica del sistema se separó de la interfaz gráfica por las siguientes razones:

1. Para simplificar el mantenimiento y la escalabilidad posteriores del sistema.

2. Para hacer posible la prueba unitaria del dominio sin la interfaz.

**4.2 Uso de clases de persistencia centralizadas**

* **Decisión:** La clase ArchivoPersistencia Centraliza el almacenamiento y la recuperación de información del sistema.
* **Justificación:** Esto evitaría la duplicación de código y mantendría una única fuente de verdad en relación con los datos.
* **Consideración:** Se estaba discutiendo la encapsulación del acceso a esta clase, pero se hizo pública para que los estudiantes interactuaran con ella directamente y, por lo tanto, simplificaran el flujo.

**4.3 Elección de Swing como la biblioteca gráfica**

La principal razón para optar por Swing fue:

1. Portabilidad en muchas implementaciones diferentes de la plataforma.

2. Tener suficientes herramientas que puedan soportar los requisitos del sistema.

3. Tener un buen soporte y una documentación extensa.

**5. Diagramas de Secuencia**

Se incluyen diagramas para representar el flujo de control en las siguientes funcionalidades principales:

1. Inicio de sesión.
2. Registro de un nuevo usuario.
3. Creación de actividades por parte del profesor.
4. Consulta de actividades por parte del estudiante.

Diagrama

Descripción generada automáticamente Diagrama

Descripción generada automáticamente Diagrama

Descripción generada automáticamente Diagrama

Descripción generada automáticamente

**6. Detalles Técnicos**

**6.1 Paquetes**

El sistema está organizado en los siguientes paquetes:

* **GUI**: Contiene todas las clases relacionadas con la implementación de una interfaz grafica.
* **Interface**: en este paquete se encuentras las clases principales desde la cual el usuario puede interactuar con el programa como profesor o estudiante. Aquí se generan las funcionalidades y se le da uso a los clases de la lógica.
* **Proyecto1**: Contiene las clases principales del negocio, como Profesor, Estudiante, LearningPath, Actividad y otras más.
* **Persistencia**: Incluye únicamente una clase con métodos para guardar, recuperar y actualizar datos.

**7. Conclusión**

El diseño modular y bien separado busca maximizar la mantenibilidad y escalabilidad del sistema. El sistema puede adaptarse a nuevas funcionalidades o requisitos sin comprometer su arquitectura actual.

​