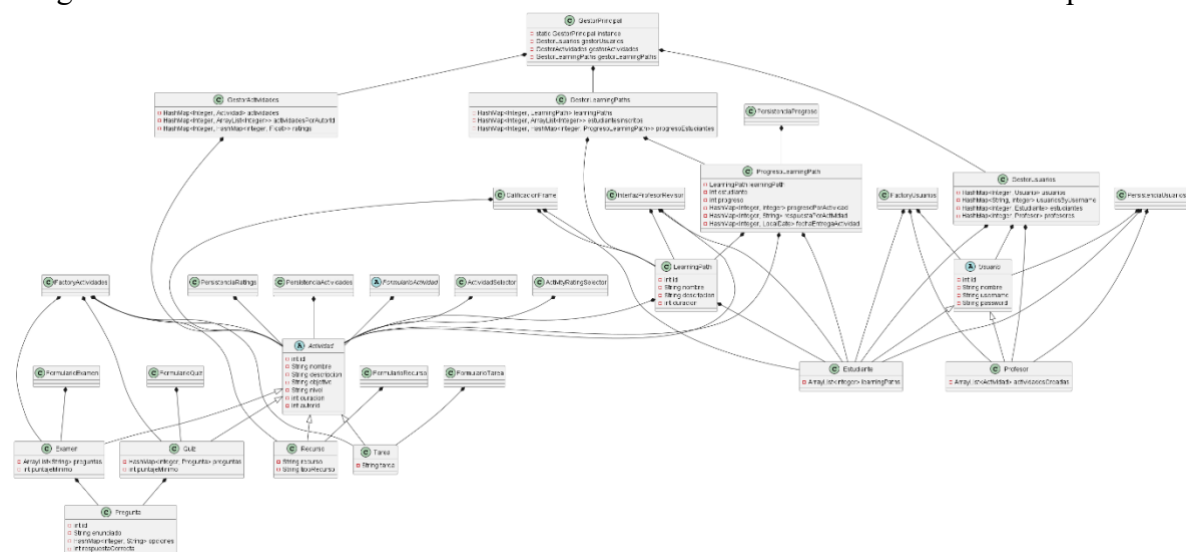


Diagrama

simplificado:

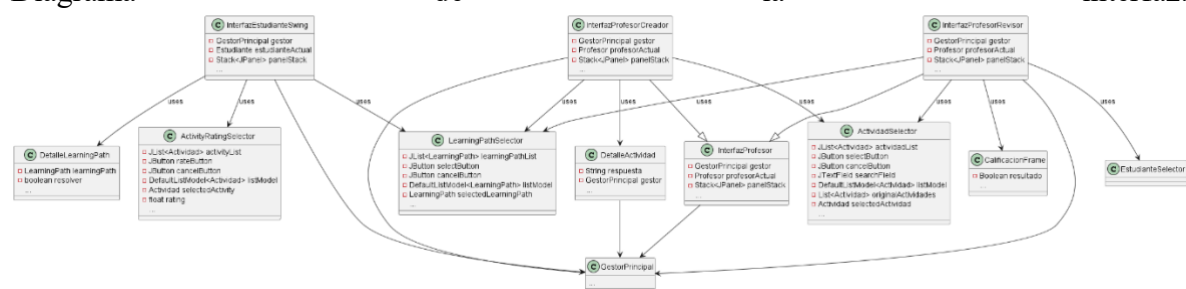


Diagrama

de

1a

interfaz:



Para esta ultima entrega, creamos 3 interfaces, que incluyen todas las funciones de la entrega anterior pero ahora en versión gráfica en lugar de consola, más varios añadidos y cambios.

En el diseño actual utilizamos varios componentes y formularios que reutilizamos para varias funciones, por ejemplo, un selector de learning paths y actividades, que nos permite buscar dinámicamente entre una lista que se le proporcione de learning paths o actividades. Debido a que depende de la lista que se entregue, podemos utilizar un selector para usar un objeto de estos del total del programa o de una selección más limitada (como los learning paths a los que anda suscrito un estudiante o creados por un profesor).

Gracias a esta implementación que reutiliza varios componentes, pudimos agilizar e implementar más fácilmente distintas funcionalidades que requerían cosas similares, como buscar los learning paths a los que un estudiante está inscrito para revisarlos o hacer una actividad.

Otro recurso que utilizamos fue la herencia, primero para compartir la parte de login y gestión del usuario actual entre ambas interfaces de profesor y otro para gestionar los formularios para los distintos tipos de actividades, ya que utilizan la misma información base pero para

los detalles correspondientes a cada tipo de actividad añadimos formularios distintos, lo que nos permitió tener distintos tipos de inputs sin hacer clases totalmente nuevas.

Cabe aclarar que también hay varias funciones que están detrás de otras ya que dejarlas desde el respectivo menú principal era innecesario, por ejemplo, copiar una actividad de otro profesor o responder a una actividad de un learning path. Para acceder a funciones como estas, hay que estar en el momento necesario, por ejemplo, al crear un learning path y elegir una actividad que no es de autoría del profesor logeado en el momento o cuando se está revisando un learning path al cual ya se inscribió.

Para ver el rating que hay de una actividad utilizamos JFreeChart (importado mediante Maven), con el cual se crea un gráfico de torta con las calificaciones que se le ha puesto a la actividad. Por otro lado, para mostrar el gráfico de actividades hechas según la época del año, hicimos que estas se mostraran mediante un calendario hecho con Java2D para ilustrarlo y colorearlo en el momento

Fuera de lo mencionado, no hubo decisiones muy específicas sobre como implementar nuestro diseño, pero buscamos crear un diseño simple para evitar perder tiempo afinando detalles visuales y así priorizar la parte funcional, que en lo que probamos hasta el momento, funciona bien tanto para crear nuevas actividades o learning paths, como para editarlos.

Respecto a cosas que ya estuvieran anteriormente, añadimos algunas funciones extras para calcular datos que mostrar en la interfaz, por ejemplo, para recopilar la cantidad de actividades entregadas por los estudiantes cada día. Esta información fue necesaria para crear de forma adecuada el gráfico de entregas por día y no requirió cambios grandes en el resto de la aplicación debido a que la lógica ya funcionaba adecuadamente desde la entrega pasada. Además de esto, como ya teníamos Maven en nuestro proyecto desde entregas anteriores, fue fácil añadir las dependencias a nuestro proyecto y facilita agregar más dependencias si lo viéramos necesario.