**Laboratorio 04**

**Integrantes:**

**Sebastián Cardona Parra**

**Diego Alexander Cárdenas Beltrán**

**Universidad Escuela Colombiana de Ingeniería Julio Garavito**  
**Programación Orientada a Objetos**  
 **María Irma Rozo**

**13/04/2024**

Texto

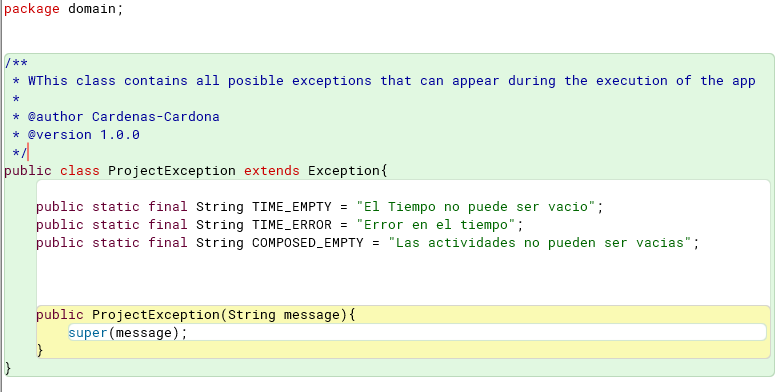
Descripción generada automáticamente



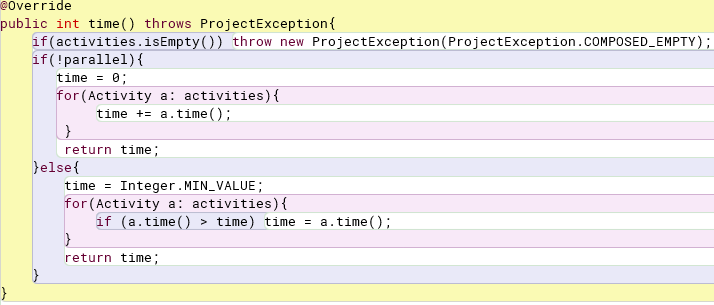
Es un árbol no binario, cuya raíz es una actividad compuestas y hojas actividades simples o compuestas.



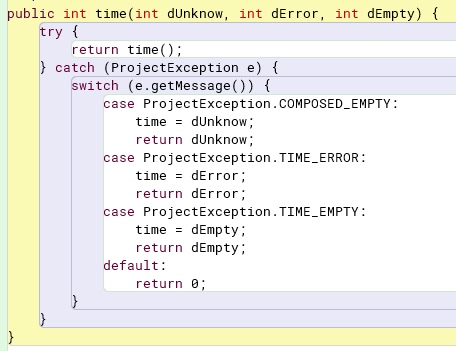
Debido a que el proyecto no cuenta con la clase ProjectException que es el que controla los posibles errores, por lo que hace falta crear las clases exception.



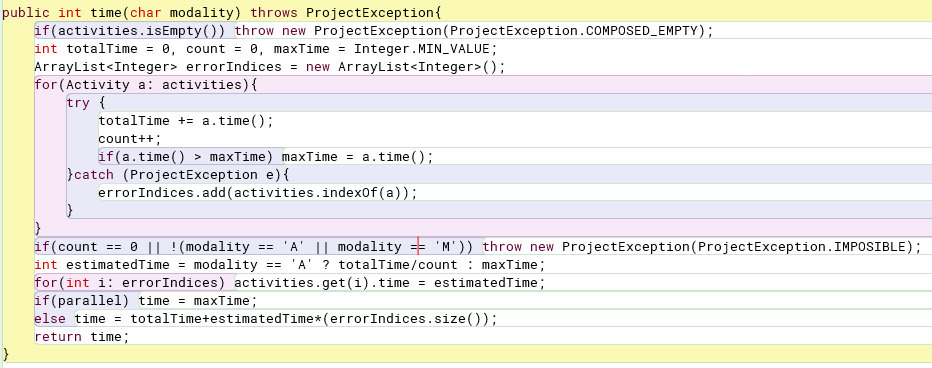




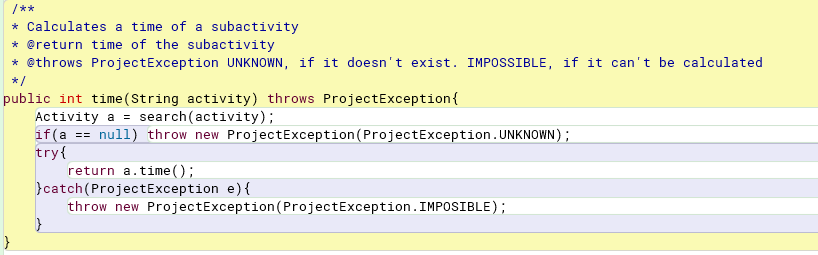


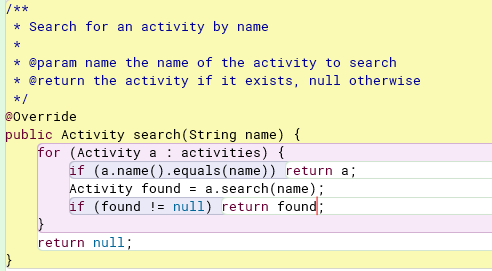


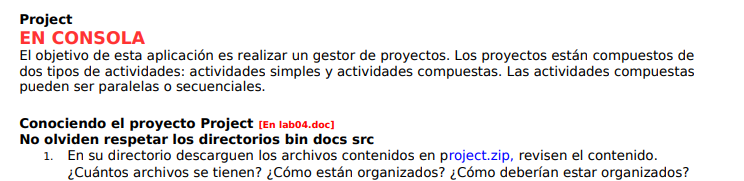






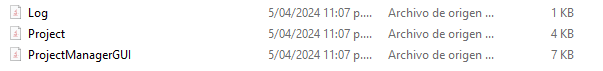






El contenedor tiene 3 archivos de tipo .java, en ellos está contenido la clase Project, que es quien maneja la lógica del gestor de proyectos, está ProjectManajerGUI, en él está la interfaz de usuario del gestor de proyectos y finalmente está la clase Log, y en esta están los registros para el mantenimiento y depuración del programa.

Deberían estar organizados de tal manera que estén en el directorio src.





Tenemos un total de 2 paquetes **domain** y **presentation**, además si queremos podemos añadir el paquete **test**

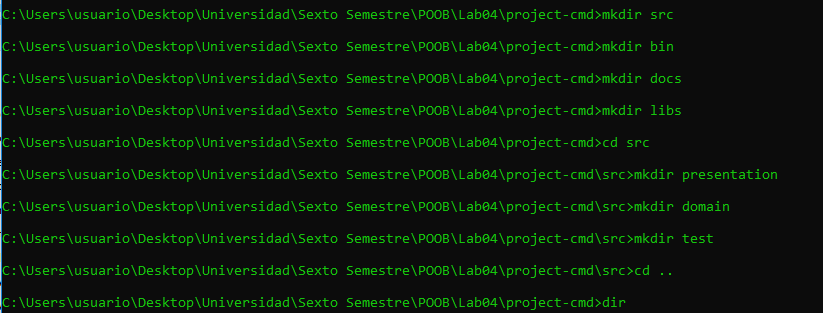
Para el proyecto total tenemos un total de 6 clases sin contar la clase de testeos y de excepciones

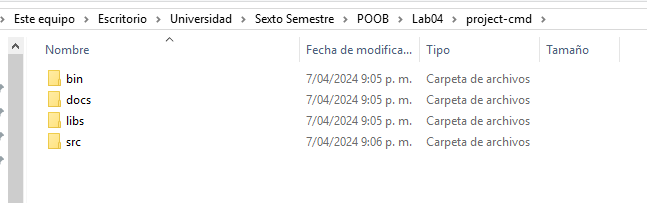
La clase ejecutiva será ProjectManajerGUI

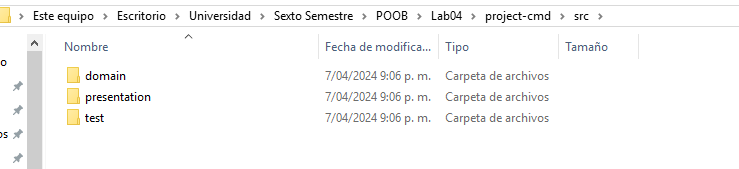


Debemos crear los directorios bin, src y docs, dentro de src estarán los paquetes presentation, domain y test, por lo cual en general en src van todas las clases .java

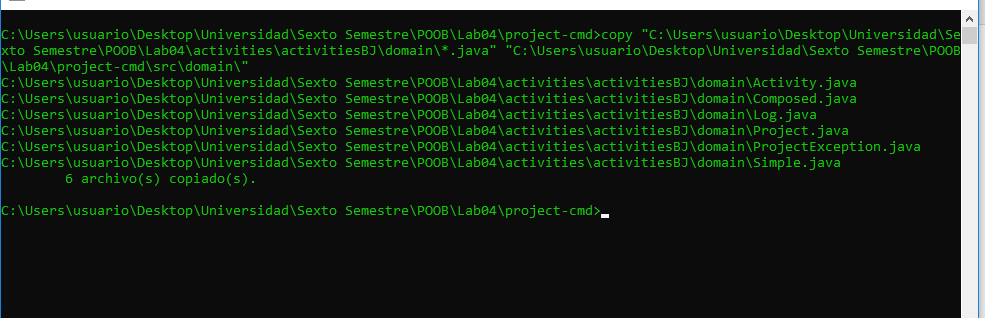
Estas clases originalmente se encuentran en el archivo de blueJ, por lo que hay que moverlas a la carpeta que creamos llamada project-cmd

Inicialmente crearemos los directorios con el comando de mkdir  






Procedemos a copiar las clases a src:  
copy "C:\Users\usuario\Desktop\Universidad\Sexto Semestre\POOB\Lab04\activities\activitiesBJ\domain\\*.java" "C:\Users\usuario\Desktop\Universidad\Sexto Semestre\POOB\Lab04\project-cmd\src\domain\"

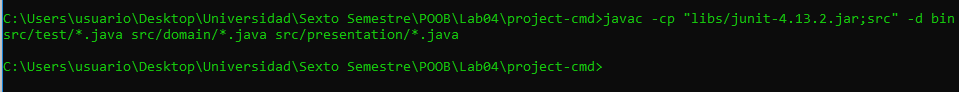


Repetimos con los demás paquetes

Y así ya tenemos los archivos .java en cada directorio correspondiente

Para compilar los archivos tengamos en cuenta que en libs, ya están los .jar del frameWork de Junit y hamcrest que funcionan para compilar y ejecutar las pruebas

javac -cp "libs/junit-4.13.2.jar;src" -d bin src/test/\*.java src/domain/\*.java src/presentation/\*.java



De esta manera ya se compile el Proyecto y está guardado como .class en el directorio bin

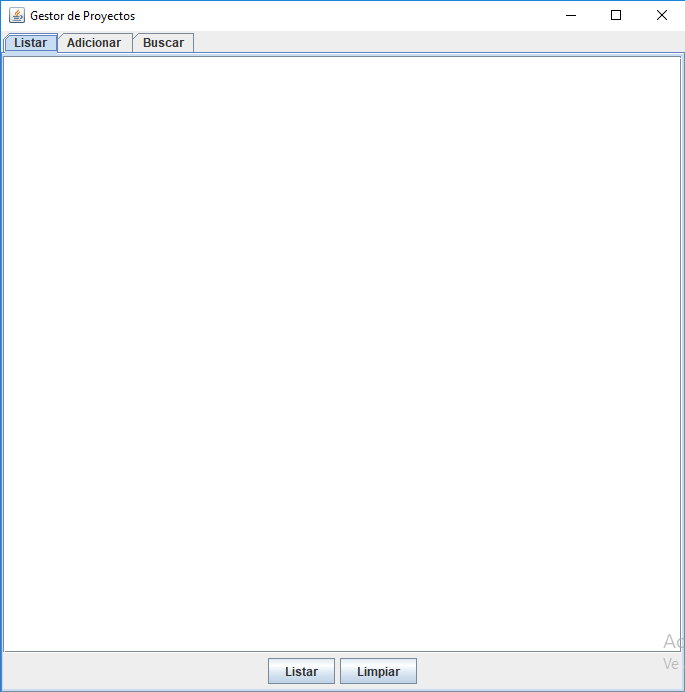
Ahora bien, vamos a crear los archivos para construir la documentación

javadoc -d docs -sourcepath src -subpackages domain:presentation -exclude test

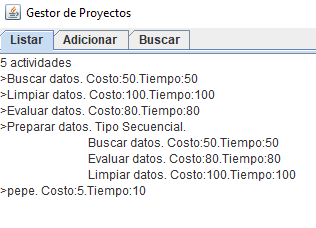
De esta manera el Proyecto queda listo para ejecutar usando la instrucción

java -cp bin presentation.ProjectManagerGUI





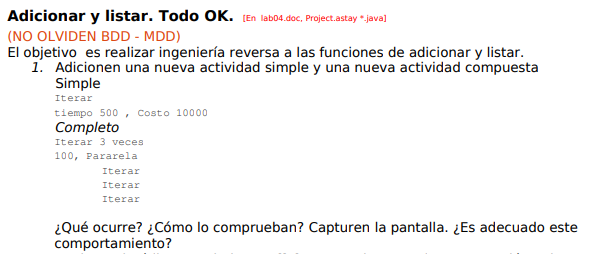
El proyecto permite editar, adicionar y buscar una actividad



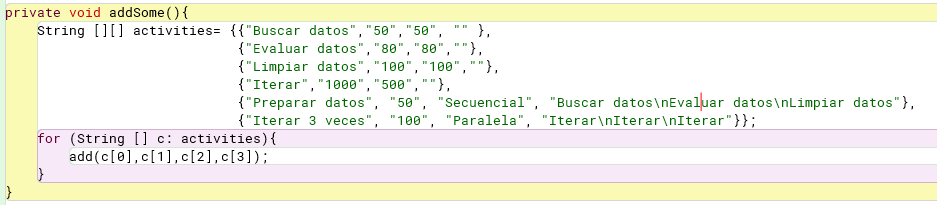
Funciona correctamente adicionar, listar, limpiar la pantalla, pero buscar no funciona bien.



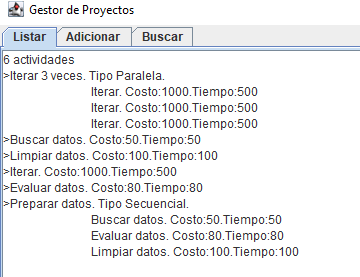
Las actividades iniciales nacen de la clase Activity, la clase Project pide que se adicionen usando el método addSome, esta que a su vez llama al método add para que añada cada actividad inicial.



Lo haremos usando el método de addSome



Al compilar y usar la funcionabilidad de listar aparecen todas las nuevas actividades creadas:



No es adecuado este comportamiento, ya que estamos usando tiempos para las actividades simples inválidos y mayores a 24, por lo que, si quiero calcular los tiempos de las actividades compuestas, nos va a lanzar un error.

Otro error es que nada me impide que haya dos actividades con el mismo nombre y esto no se debe poder pasar.

Que sucede cuando dije que una actividad era paralela o secuencial (compuesta) pero no le pongo sub-actividades, el código lo va a reconocer como una actividad simple, pero van a existir errores respecto a tiempo y la palabra paralelo o secuencial.

Lo mismo sucede si diseño una actividad simple (con tiempo) pero le pongo actividades, el código lo reconocerá como una actividad compuesta y habrá conflictos con el entero del tiempo y la palabra paralelo o secuencial.

Para una actividad compuesta, nada me impide que ponga otra palabra distinta a secuencial y paralelo, si pongo ejemplo “pepito” lo reconoce como un paralelo y es incoherente, lo mismo si pongo “Carlos”, lo reconocerá como secuencial

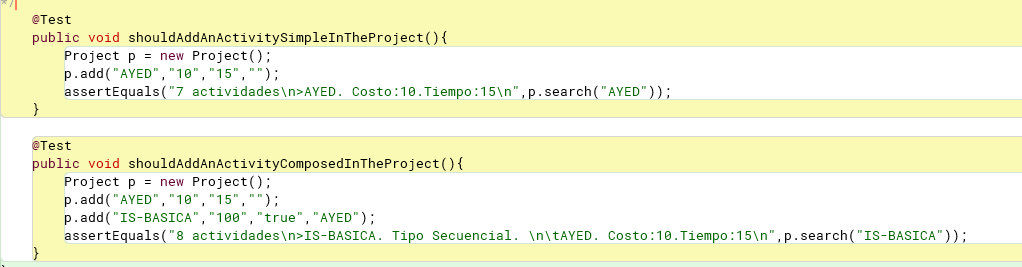
Esos párrafos resaltados creo que me equivoqué, van más adelante

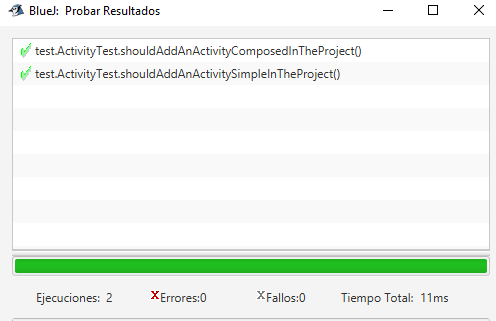


En la capa de dominio, el encargado de añadir actividades es el método add

En la capa de presentación, el método encargado de añadir actividades es ActionAdd



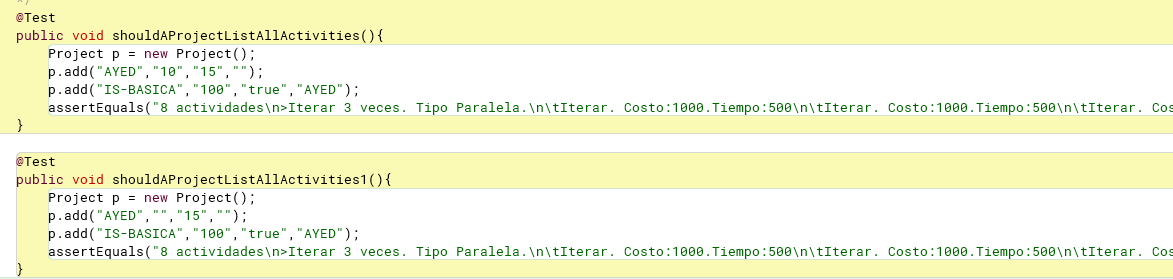


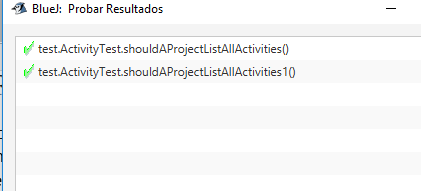


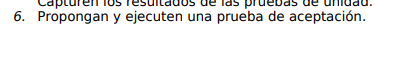


En la capa de domain el método responsable de listar todas las actividades es toString en la clase de Project, este a su vez usa el método de data que toma un linkedlist con las actividades que quiere listar. En la capa de presentación, el método encargado es actionList









Crearemos una nueva actividad compuesta por tres simple, después de ello verificaremos que, al consultar, verdaderamente existan y calculamos que el tiempo de la actividad compuesta sea correcto

