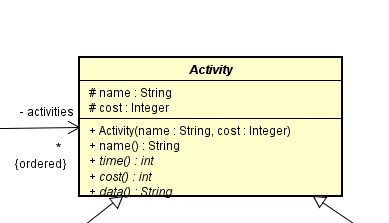
Daniel Alejandro Diaz Camelo

Miguel Ángel Motta Urrea

**Laboratorio 4**

**1.** En su directorio descarguen los archivos contenidos en activities.zip revisen el contenido y estudien el diseño parcial que está en el diagrama de clases. ¿Qué estructura de datos es Actividad? Justifique su respuesta

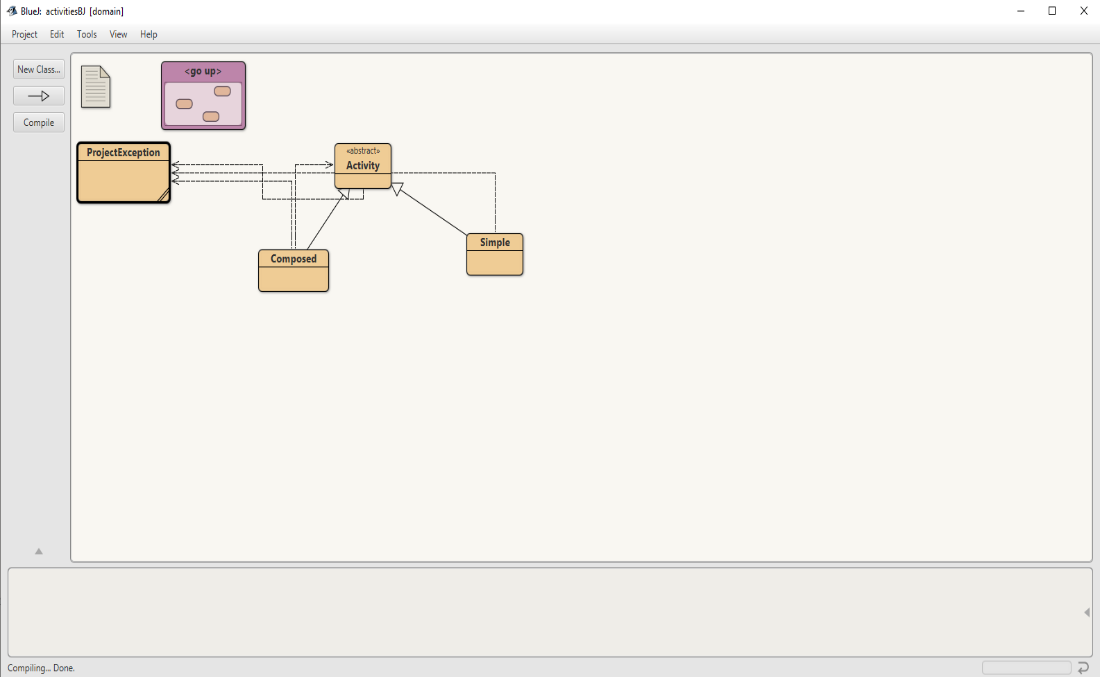
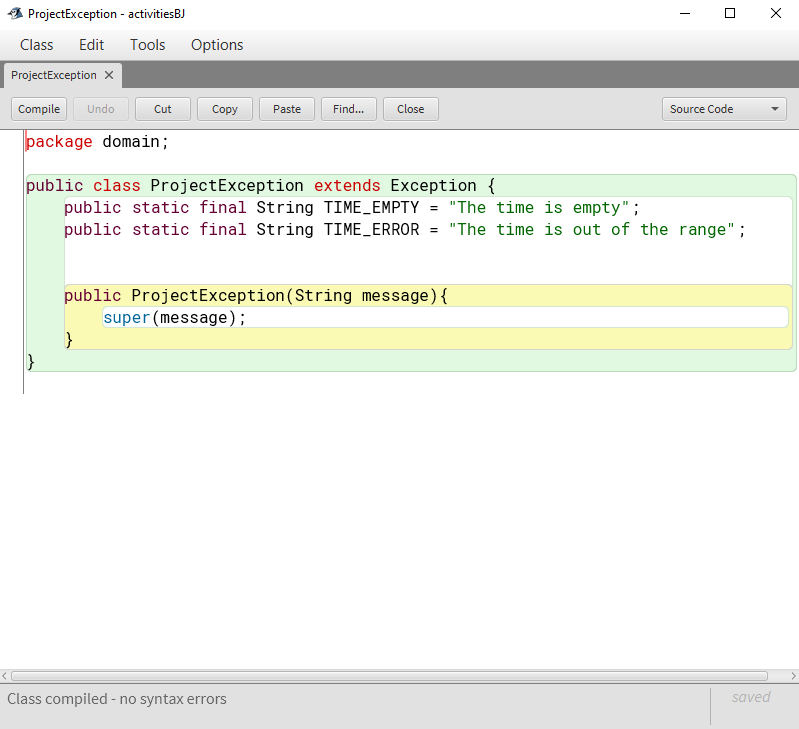
R/= Actividad es una clase abstracta, como lo muestra el diagrama de clases al tener el nombre inclinado, y como sale en el código de Actividad.java:

Además, la clase composed tiene varias actividades que las almacena en un ArrayList.

**2.** Expliquen por qué el proyecto no compila. Realicen las adiciones necesarias para lograrlo.

R/= El proyecto no compila debido a que ProjectException no existe, por lo que toca crear esta clase la cual debe extender la clase Exception para manejar nuestras propias excepciones.

**3.** Dadas las pruebas, documenten, diseñen y codifiquen el método time().

R/ = En BlueJ y en Astah

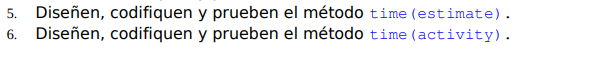
**4.** Dada la documentación y el diseño, codifiquen y prueben el método time(unknown, error, empy).

R/ = En BlueJ y en Astah

**5.** Documenten, diseñen, codifiquen y prueben el método price(estimate).

R/= En BlueJ y en Astah.

Suponemos que cuando dice “time” se refiere a “price”:



Escala de tiempo

Descripción generada automáticamente

**6.** Documenten, diseñen, codifiquen el método price(activity).

R/= En BlueJ y en Astah

En Consola

**1.** En su directorio descarguen los archivos contenidos en project.zip, revisen el contenido. ¿Cuántos archivos se tienen? ¿Cómo están organizados? ¿Cómo deberían estar organizados?

R/=

Se tienen 3 archivos:

Texto

Descripción generada automáticamente

No están organizados, solo están dentro del paquete de Project.zip.

Dado el código en cada uno de los ficheros, deberían estar organizados en dos directorios, domain y presentation, respectivamente:

Texto

Descripción generada automáticamente

**2.** Realicen el diseño del programa: diagramas de paquetes y de clases (presentación y dominio). ¿cuántos paquetes tenemos? ¿cuántas clases tiene el sistema? ¿cuál es la clase ejecutiva?

R/= Tendríamos dos paquetes (en src además del de test) que son domain y presentation, en estos se encuentran inicialmente los archivos.java de:

* domain:

Project.java

Log.java

* presentation:

ProjectManagerGUI.java

Diagrama

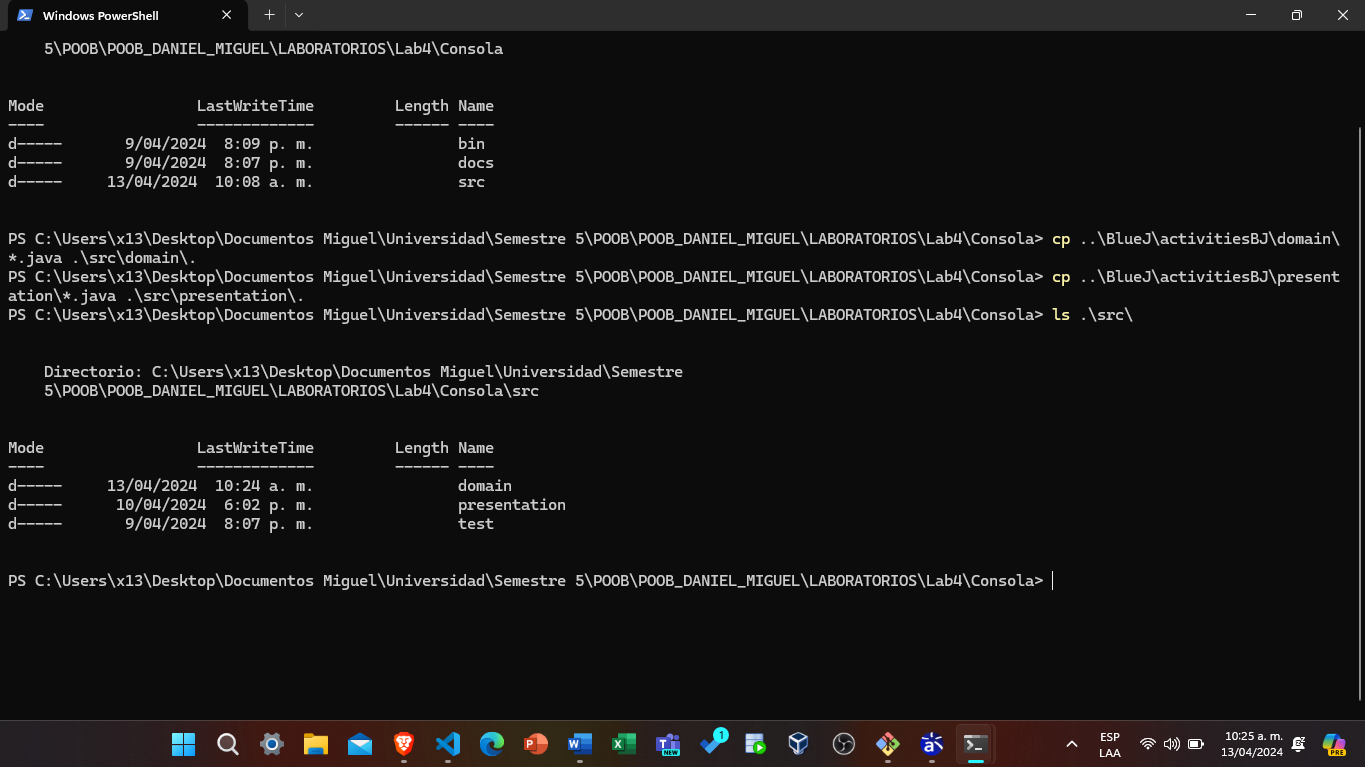
Descripción generada automáticamente

La clase ejecutiva es ProjectManagerGUI, que contiene el método main().

**3.** Prepare los directorios necesarios para ejecutar el proyecto. ¿qué estructura debe tener? ¿qué clases debe tener cada directorio? ¿dónde están esas clases? ¿qué instrucciones debe dar para ejecutarlo?

R/= Primero se deben conectar las clases hechas en el primer punto, para ello agregamos todos los archivos .java que se trabajaron en el punto de BlueJ, sin contar las pruebas.

Esto lo hacemos con el siguiente comando:



cp ..\BlueJ\activitiesBJ\domain\\*.java .\src\domain\.

cp ..\BlueJ\activitiesBJ\test\\*.java .\src\test\.

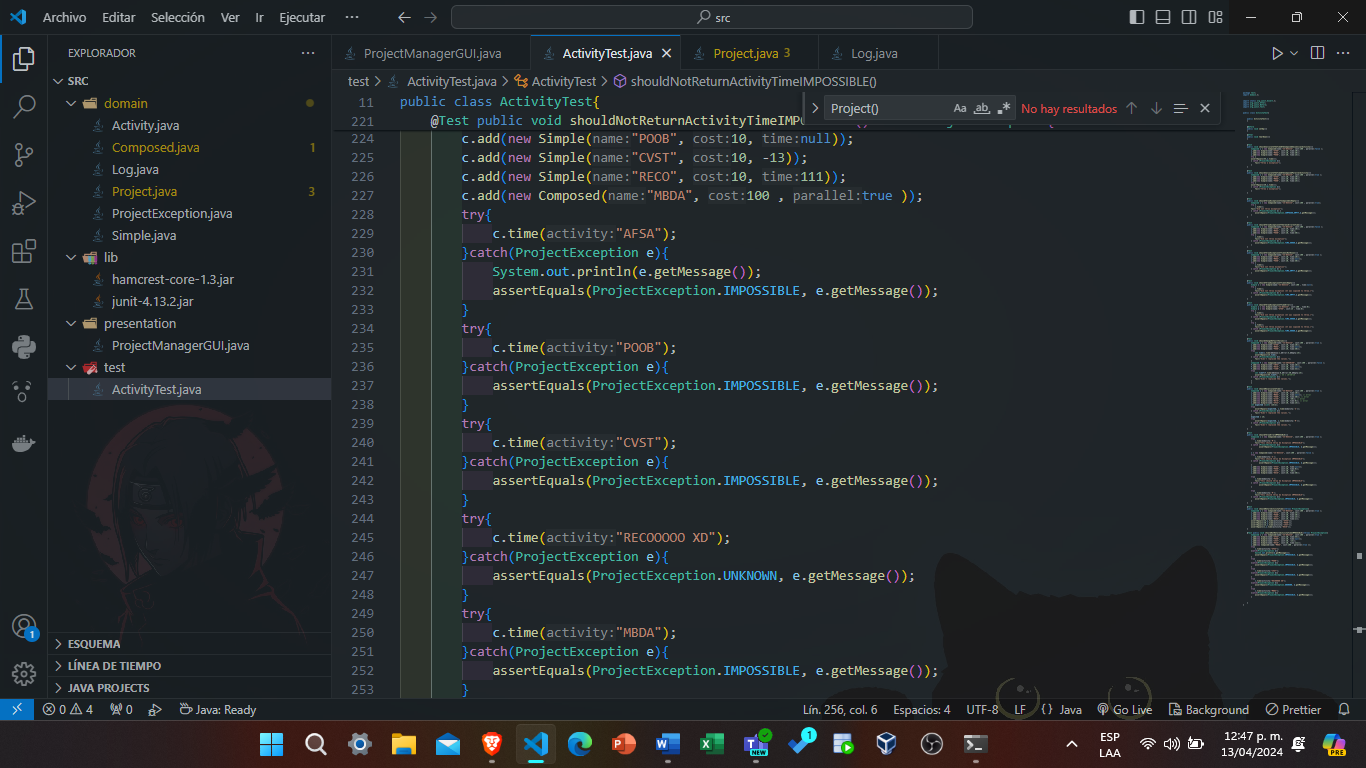
Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Solo con hacer esto se resuelven los problemas del compilador:



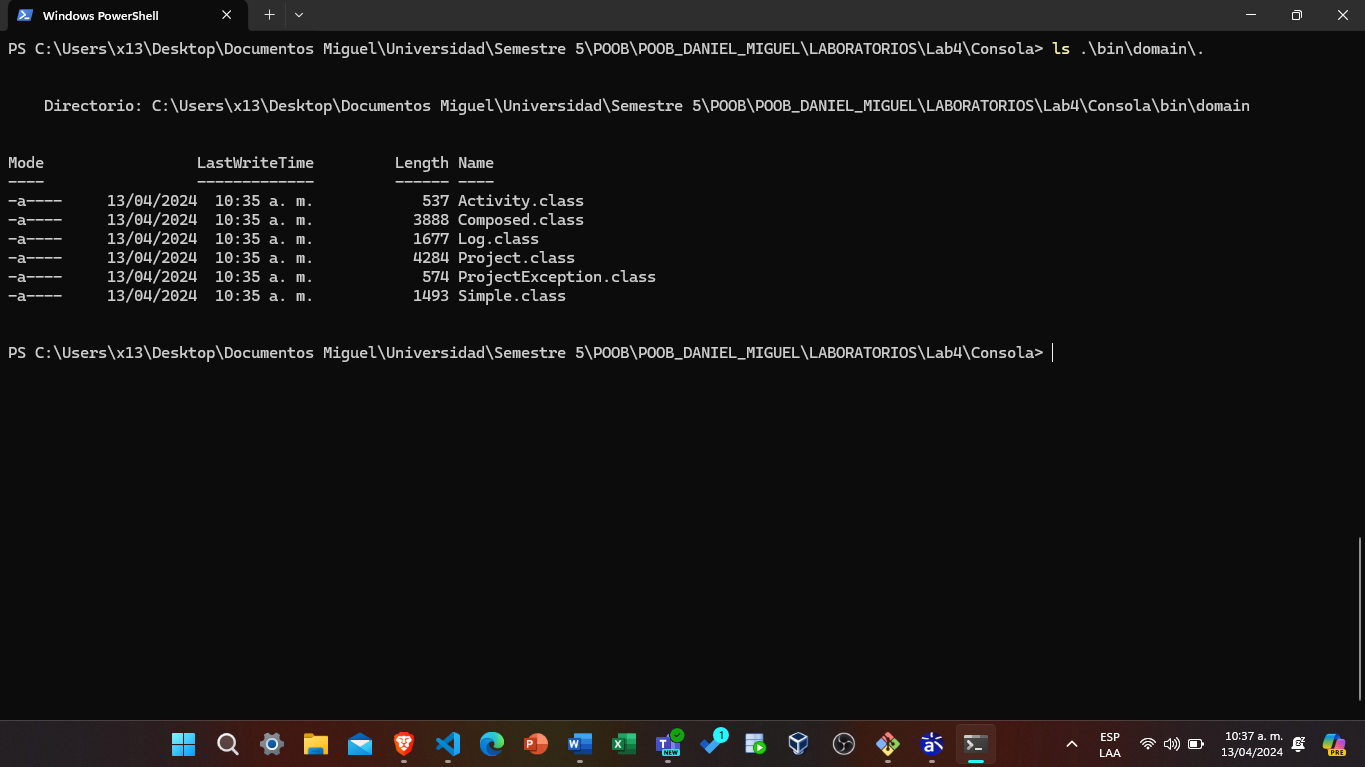
Luego para compilarlo, ejecutamos el siguiente comando:

javac -d .\bin\ .\src\domain\\*.java .\src\presentation\\*.java

Texto

Descripción generada automáticamente

Al examinar los directorios tenemos:

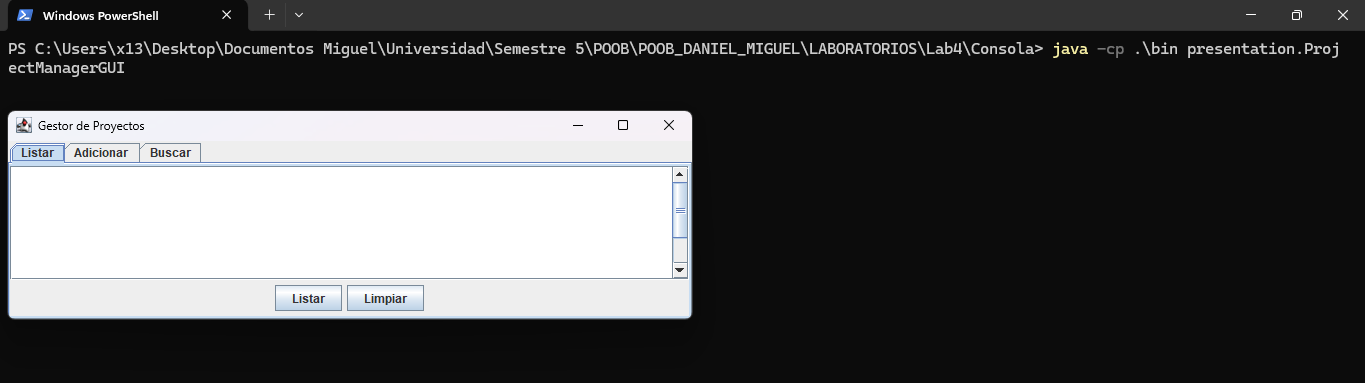


Texto

Descripción generada automáticamente

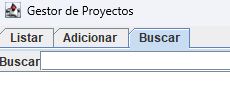
Ahora, para ejecutarlo, lo hacemos desde la función main(), que se encuentra en la clase ProjectManagerGUI:

java -cp .\bin presentation.ProjectManagerGUI



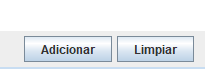
**4.** Ejecute el proyecto, ¿qué funcionalidades ofrece? ¿cuáles funcionan?

R/=



Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

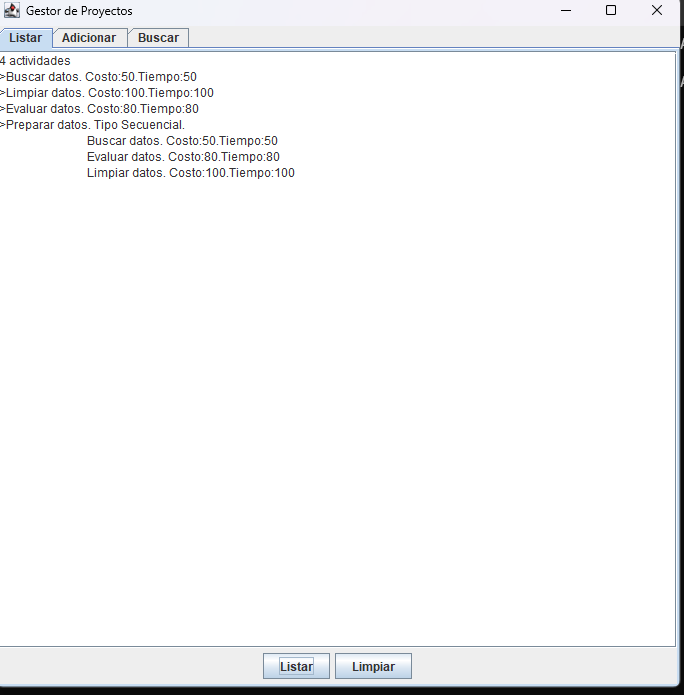
Descripción generada automáticamente





Una interfaz gráfica con botones en la parte superior para limpiar y listar, limpiar limpia el espacio de la Interfaz y listar genera un conjunto de actividades, además de los botones de la parte superior que dicen, Listar, Adicionar y Buscar, que parecen pestañas de navegación:

Imagen que contiene Texto

Descripción generada automáticamente   


Parece que solo listar y Limpiar funcionan, pues los botones de Adicionar muestran algún tipo de cambio, y en Buscar está la barra de búsqueda pero no los botones.

**5.** Revisen el código y la documentación del proyecto. ¿De dónde salen los disfraces iniciales? ¿Qué clase pide que se adicionen? ¿Qué clase los adiciona?

R/= La interfaz gráfica viene de este código:

Texto

Descripción generada automáticamente

Que se extiende a su vez en el constructor de Project()

Texto

Descripción generada automáticamente

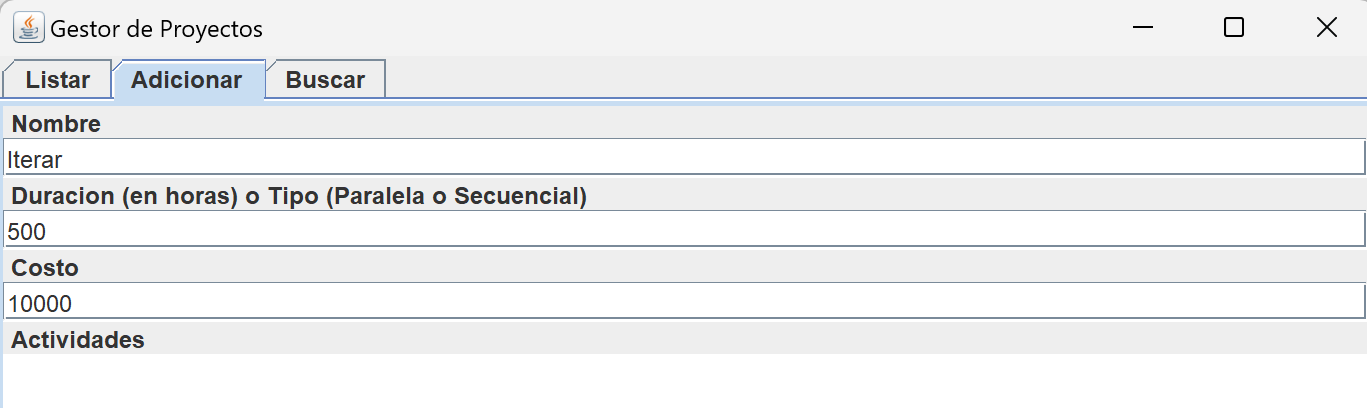
Aquí podemos ver que los adiciona la clase Project.

**Adicionar y listar. Todo OK.**

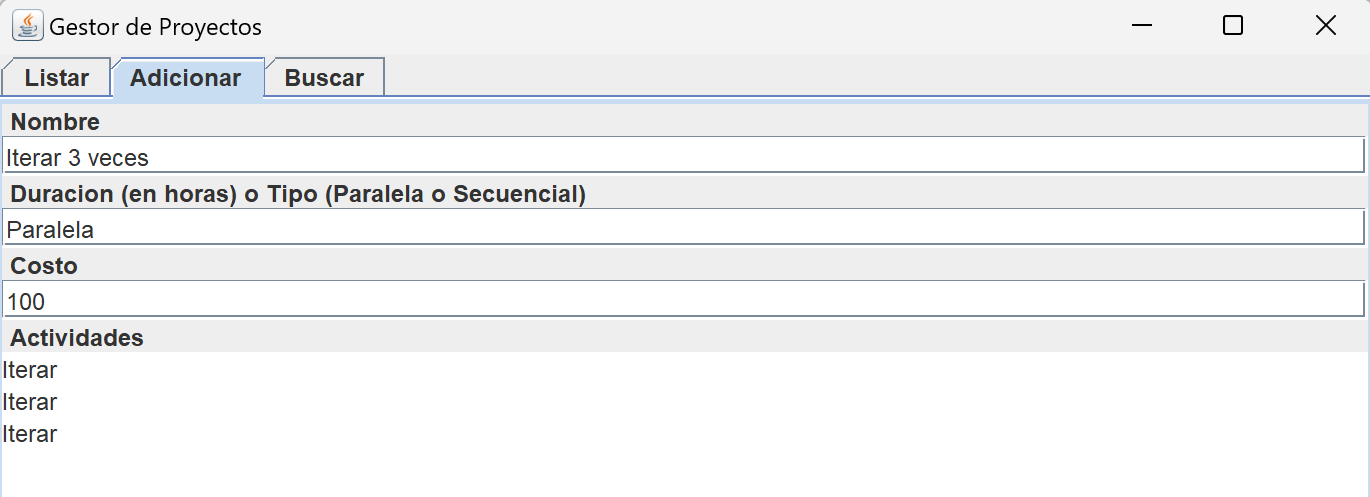
**1.** Adicionen una nueva actividad simple y una nueva actividad compuesta

R/=

Simple



Completo



Lo que ocurre es que se añaden las respectivas actividades y lo podemos comprobar en la pestaña de listar:

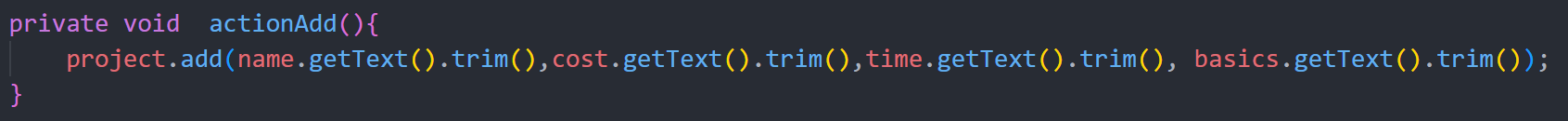


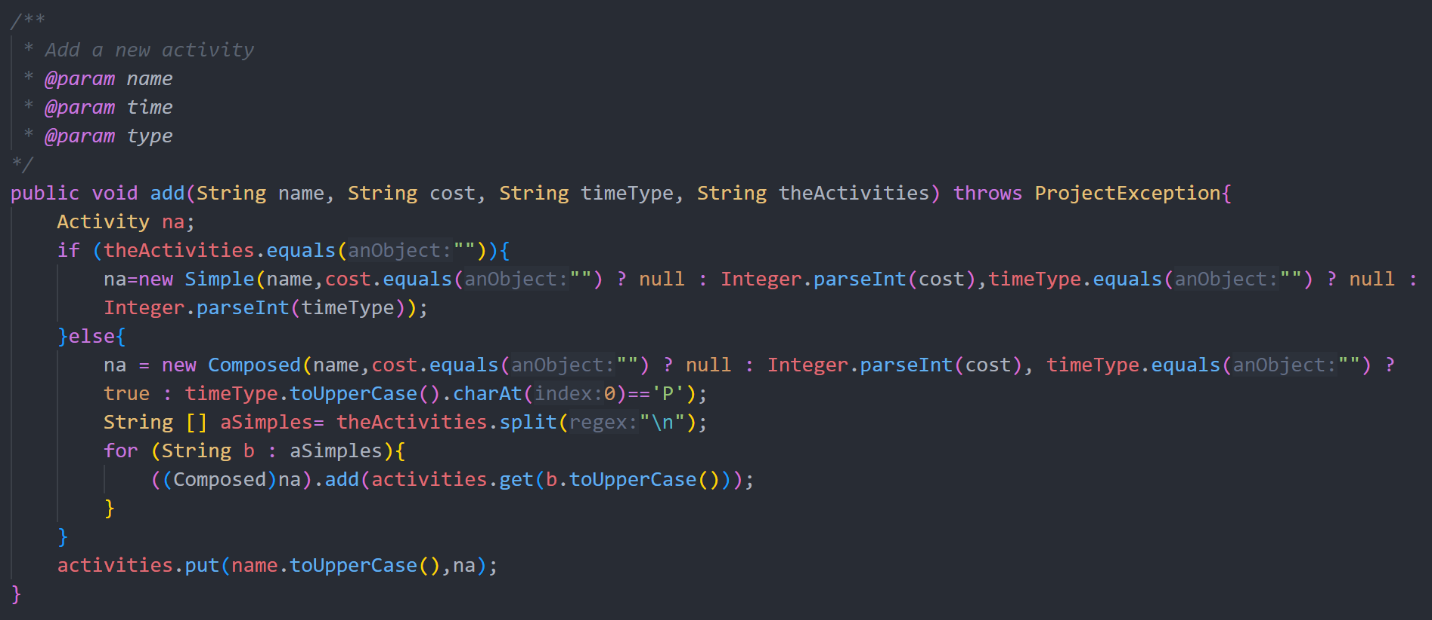
En cuanto al comportamiento si es lo esperado, pero no del todo, pues se le ingresó a la tarea paralela que tendría un costo de 100 en la cual no es claro si lo tuvo en cuenta o no, puesto que no se refleja en la interfaz gráfica.

**2.** Revisen el código asociado a adicionar en la capa de presentación y la capa de dominio. ¿Qué método es responsable en la capa de presentación? ¿Qué método en la capa de dominio?

R/=

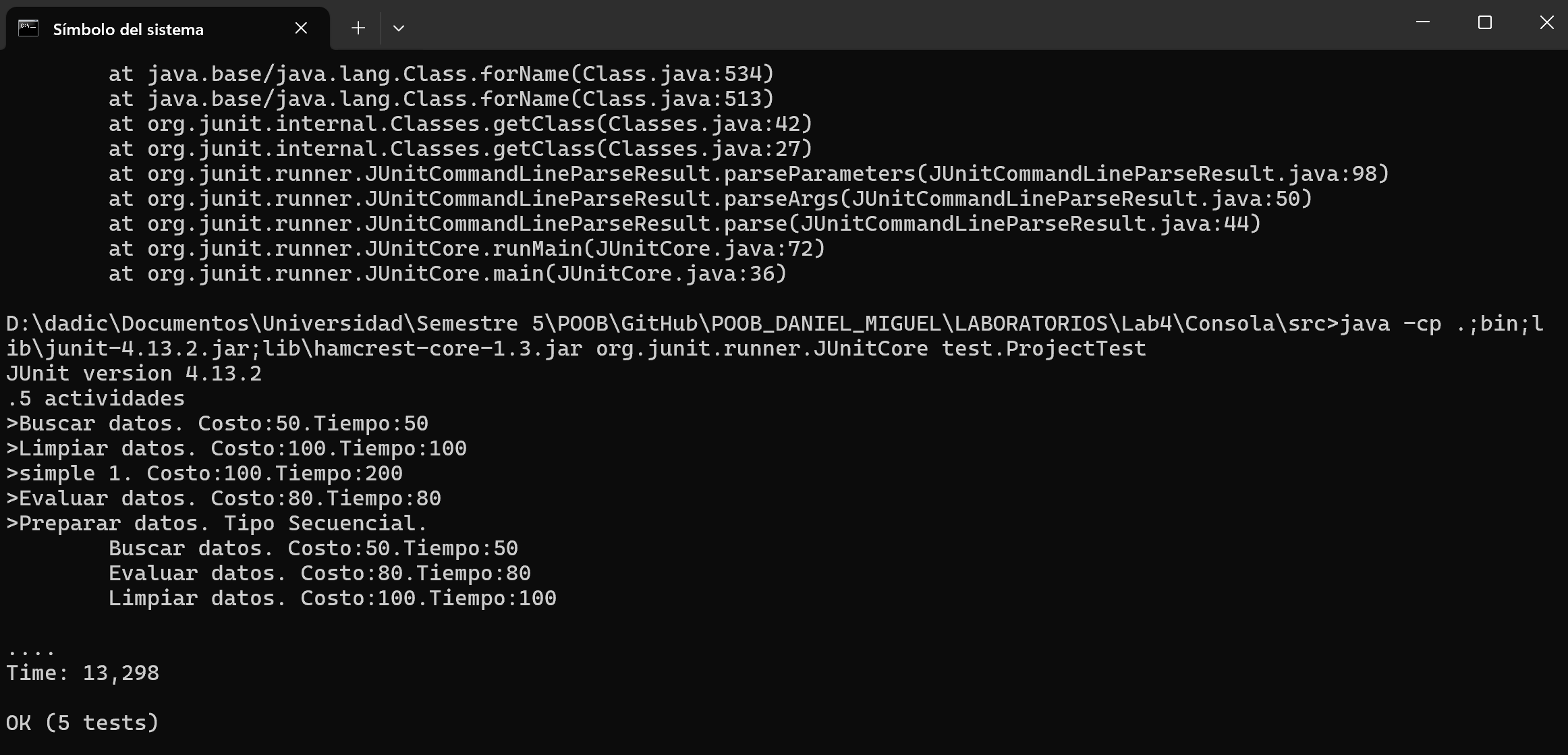
El metodo asociado a presentacion es actionAdd(), en donde se le indica al programa que al darle click en “Adicionar” se debe ejecutar el metodo add() el cual esta en project.java

El metodo asociado a dominio es add(name, cost, timeType, theActivities), el cual se encarga de adicionar ya sea la tarea simple como la compleja



**3.** Realicen ingeniería reversa para la capa de dominio para adicionar. Capturen los resultados de las pruebas de unidad.

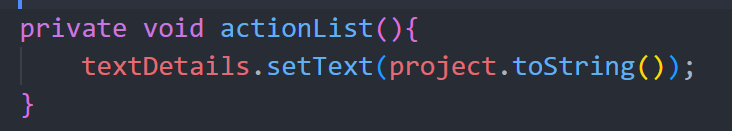
R/=

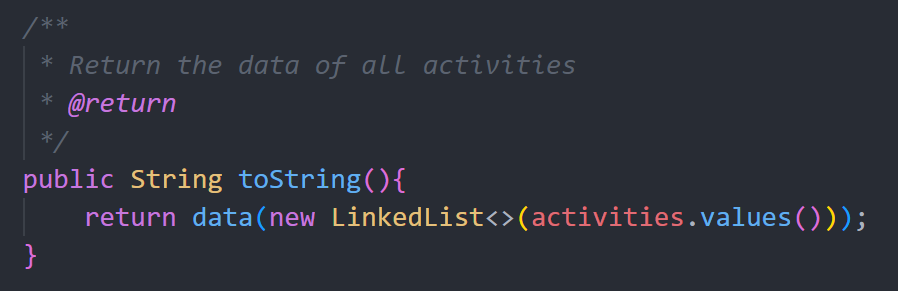


**4.** Revisen el código asociado a listar en la capa de presentación y la capa de dominio. ¿Qué método es responsable en la capa de presentación? ¿Qué método en la capa de dominio?

R/=

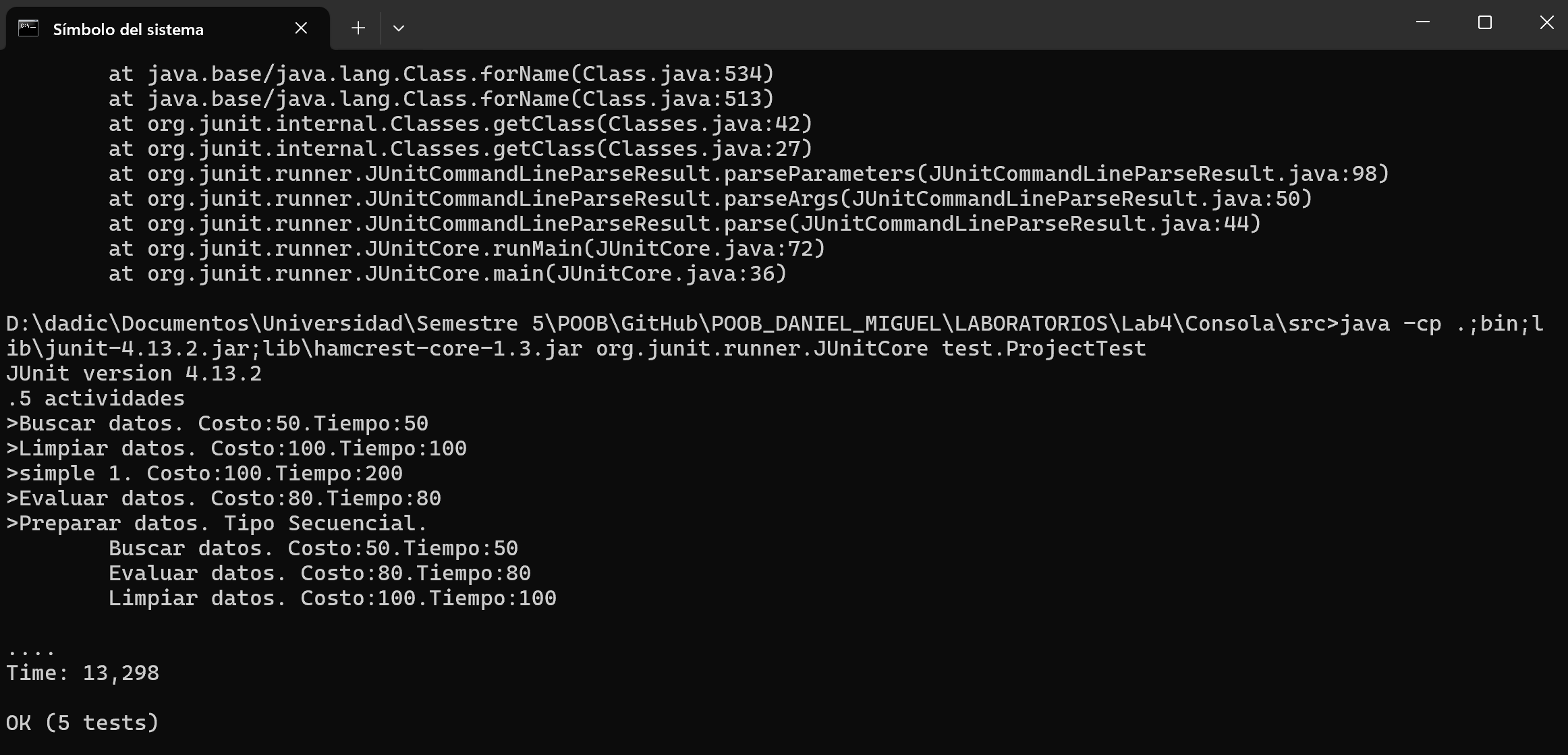
El metodo asociado a presentacion es actionList(), en donde se le indica al programa que al darle click en “Listar” se debe ejecutar el metodo toString() el cual esta en project.java

El metodo asociado a dominio es toString(), el cual se encarga de mostrar la informacion de todas las actividades agregadas.

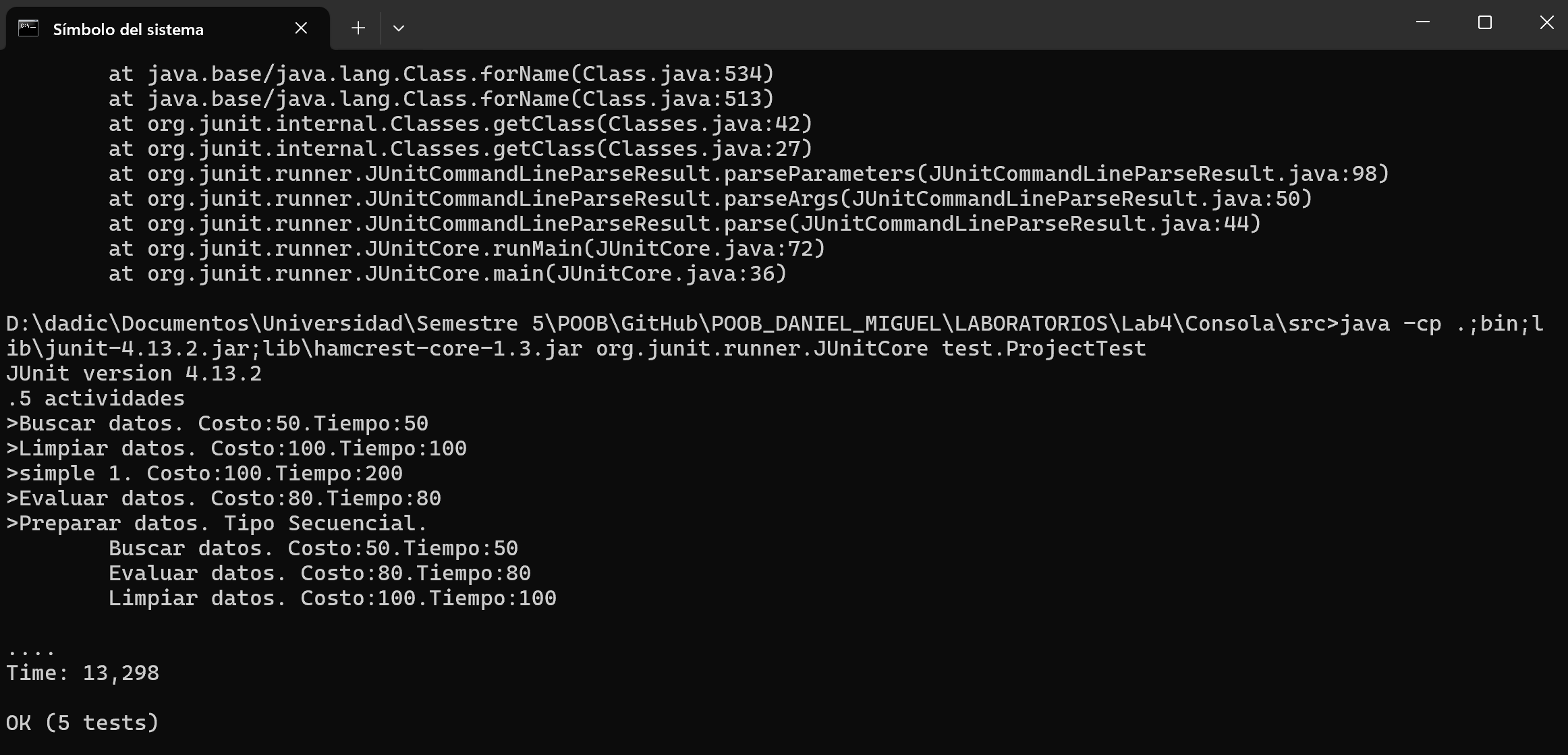


**5.** Realicen ingeniería reversa para la capa de dominio para listar. Capturen los resultados de las pruebas de unidad.

R/=



**6.** Propongan y ejecuten una prueba de aceptación.



**Adicionar una actividad. Funcionalidad robusto**

El objetivo es perfeccionar la funcionalidad de adicionar un curso para hacerla más robusta.

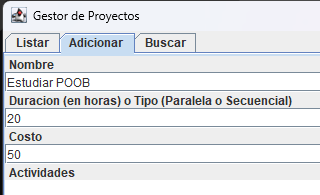
1. ¿Y si el nombre de la actividad ya existe?

**1.** Propongan una prueba de aceptación que genere el fallo.

R/=

Que primero se cree una Actividad:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Luego que se cree nuevamente otra tarea con el mismo nombre:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamenteInterfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Ahora vemos que se cambió, pero no debería ser así y debería impedirlo o generar un fallo como mensaje para el usuario.

**2.** Analicen el diseño realizado. Para hacer el software robusto: ¿Qué método debería lanzar la excepción? ¿Qué métodos deberían propagarla? ¿Qué método debería atenderla explique claramente?

R/=

* El método que debería lanzar la excepción sería:

add(Activity a)

Puesto que este método está en Project, y a su vez, Project conoce las demás actividades, por lo cual puede corroborar si la actividad existe o no, y lanzar el error. Esto verifica que no se repita en una actividad compuesta, pero también debe lanzar una excepción si se repite en Project, entonces se lanzaría en dos Clases.

* Los métodos que deberían propagar la excepción serían:

add(String name, String cost, String timeType, String theActivities)

actionAdd()

Puesto que es un método que conecta el llamado para agregar una nueva actividad desde la interfaz y el método de agregar actividad de proyecto (Project).

* El método que debería atenderla, sería:

actionPerformed(ActionEvent ev)

Ya que es el metodo final, que es más próximo a recibir las entradas del usuario.

Una pantalla de fondo

Descripción generada automáticamente con confianza media**3.** Construya la solución propuesta. Capture los resultados de las pruebas de unidad.

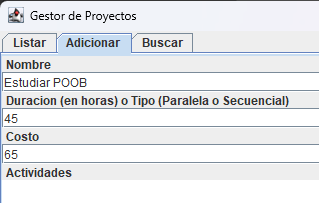
Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente**4.** Ejecuten nuevamente la aplicación con el caso de aceptación propuesto en 1. ¿Qué sucede ahora? Capture la pantalla.Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamenteInterfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Ahora si intentamos añadir nuevamente una clase ya existente en proyecto, se genera un error.

1. ¿Y si en precio o costo no da un número? ¿si el tipo no es paralelo o secuencial?
2. Propongan una prueba de aceptación que genere el fallo.

p.add("POOB", "asdf", "55", "");

1. Analicen el diseño realizado. Para hacer el software robusto: ¿Qué método debería lanzar la excepción? ¿Qué métodos deberían propagarla? ¿Qué método debería atenderla explique claramente?

Esto lo debería lanzar el método de add(..) en Project, pues es donde se origina el Error, ser propagado en los métodos:

add(String name, String cost, String timeType, String theActivities)

actionAdd()

porque no tiene sentido atenderlo ahí, no se mantiene el flujo sino que se detendría en métodos que no reciben la interacción directa con la causa del fallo, que fue el error del usuario. Y ser atendido en el método:

actionPerformed(ActionEvent ev)

Pues recibe la acción directa del usuario.

1. Captura de pantalla de computadora

   Descripción generada automáticamenteConstruya la solución propuesta. Capture los resultados de las pruebas de unidad.[en domain/\*.java]
2. Ejecuten nuevamente la aplicación con el caso de aceptación propuesto en 1.

**Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamenteInterfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente**

Ahora le muestra su error al usuario

1. Proponga una nueva condición

¿Qué pasa si los tipos de actividades dados no existen en el proyecto?

1. Propongan una prueba de aceptación que genere el fallo.

p.add("POOB", "100", "Paralela", "Jugar Bolos);

1. Analicen el diseño realizado. Para hacer el software robusto: ¿Qué método debería lanzar la excepción? ¿Qué métodos deberían propagarla? ¿Qué método debería atenderla explique claramente?

Esto lo debería lanzar el método de add(..) en Project, pues es donde se origina el Error, ser propagado en los métodos:

add(String name, String cost, String timeType, String theActivities)

actionAdd()

porque no tiene sentido atenderlo ahí, no se mantiene el flujo sino que se detendría en métodos que no reciben la interacción directa con la causa del fallo, que fue el error del usuario. Y ser atendido en el método:

actionPerformed(ActionEvent ev)

Pues recibe la acción directa del usuario.

1. Texto

   Descripción generada automáticamenteConstruya la solución propuesta. Capture los resultados de las pruebas de unidad.[en domain/\*.java]
2. Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

   Descripción generada automáticamenteInterfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

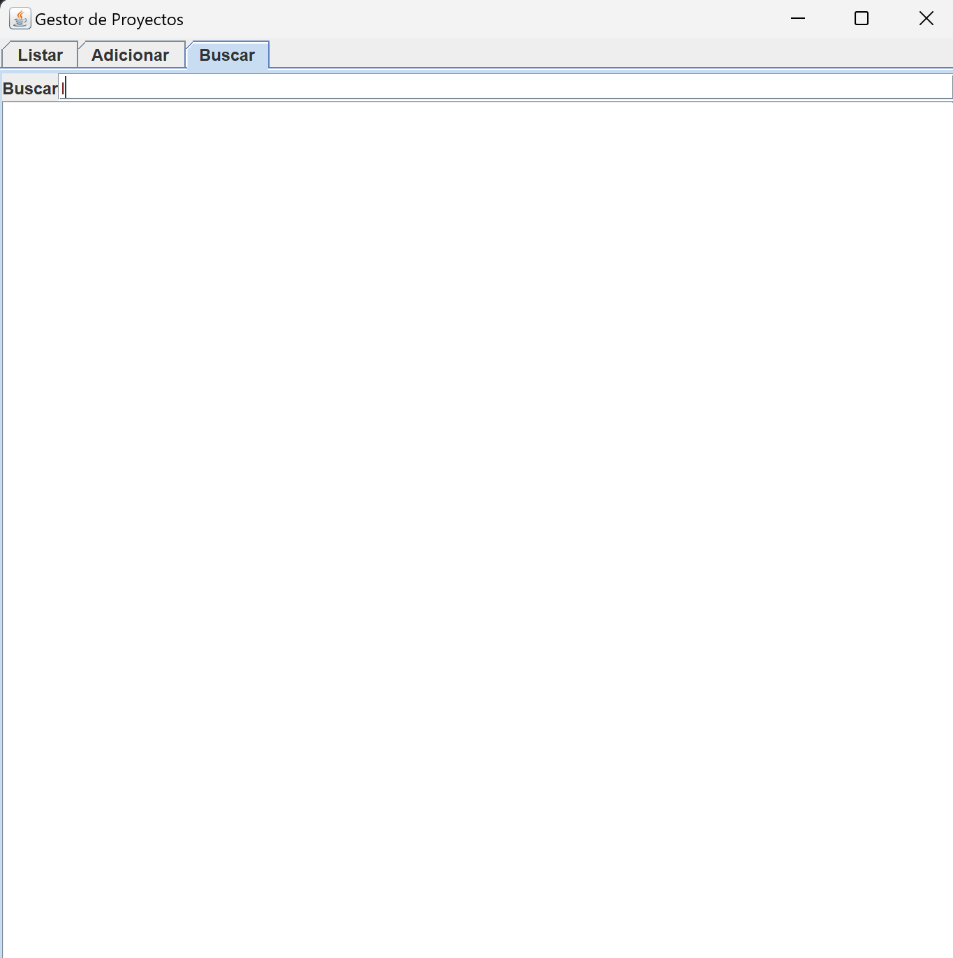
   Descripción generada automáticamenteEjecuten nuevamente la aplicación con el caso de aceptación propuesto en 1.

Ahora se lanza un error indicando que si quiere crear una actividad con subactividades tendrá que ser con actividades existentes.

**Consultando por patrones. ¡No funciona y queda sin funcionar!**

**1.** Consulten una actividad que inicie con I. ¿Qué sucede? ¿Qué creen que pasó? Capturen el resultado. ¿Quién debe conocer y quien NO debe conocer esta información?

R/=





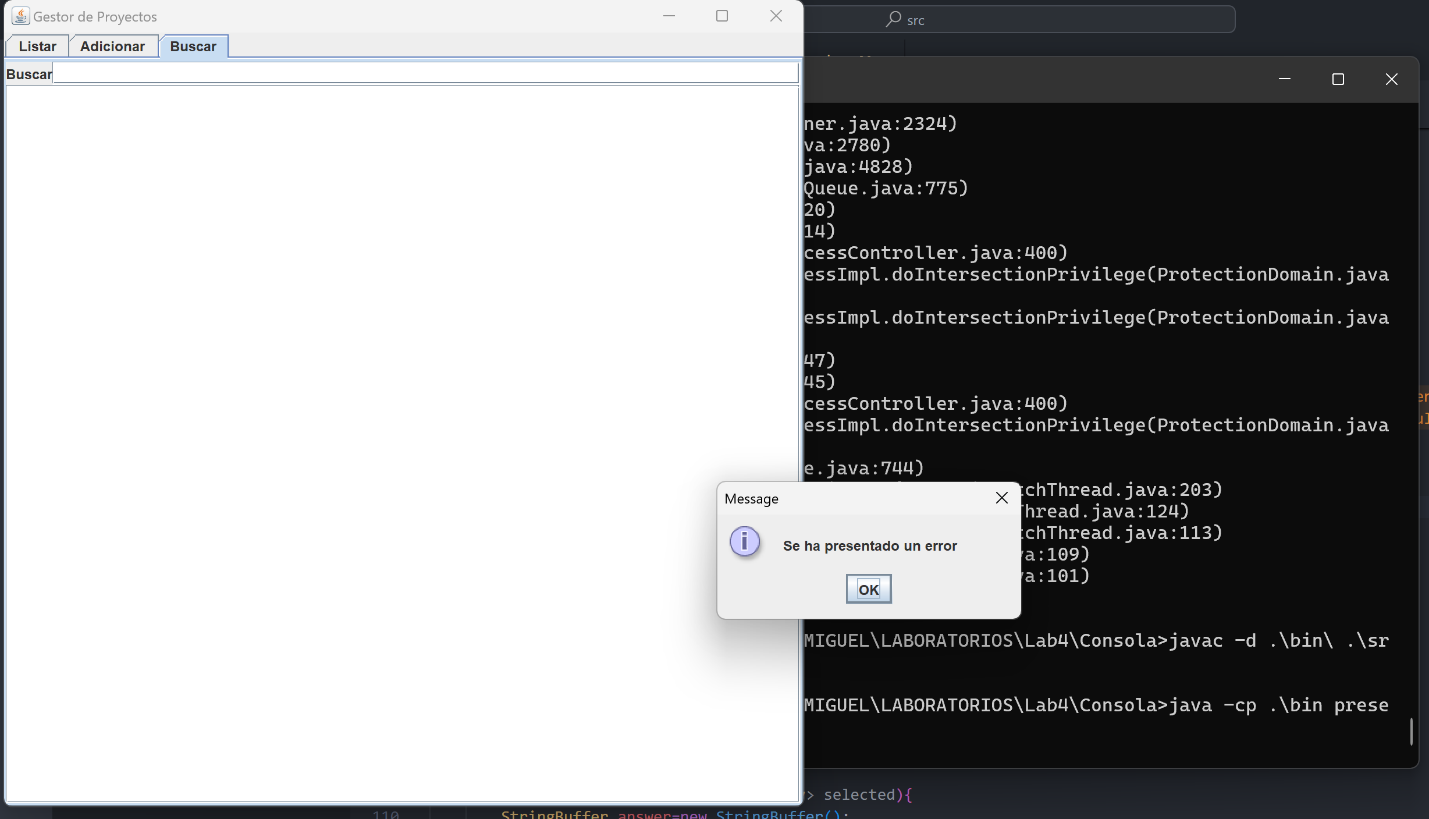
Al buscar la actividad pedida no se muestra nada, ya que lanza un error de que el objeto es nulo por lo que no puede acceder al método name(). En cuanto a quien debe conocer la información de la actividad es el usuario que está usando el programa mientras que los desarrolladores no necesitan saber de esta, puesto su trabajo es implementar la funcionalidad.

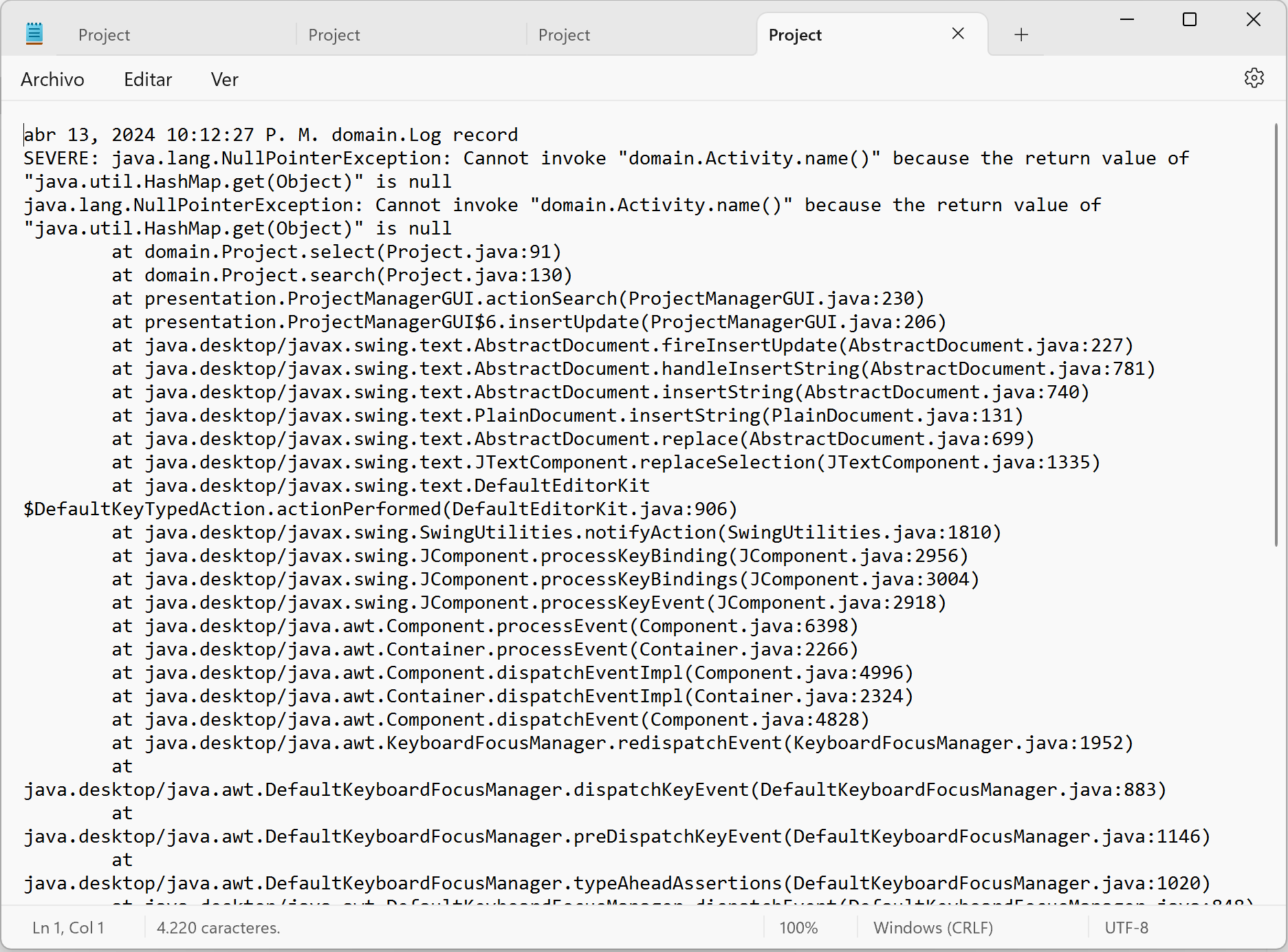
**2.** Exploren el método récord de la clase Log ¿Qué servicio presta?

R/= El método récord lo que hace es registrar todas las excepciones que ocurran en el programa en un archivo.

**3.** Analicen el punto adecuado para que EN ESTE CASO se presente un mensaje especial de alerta al usuario, se guarde la información del error en el registro y continúe la ejecución. Expliquen y construyan la solución.

R/= Para realizar este punto se implementó un bloque try-catch en el metodo select() (en Project) en el cual si captura una excepcion, se llama el metodo record() (en Log) para que asi almacene la excepcion en un archivo. Ademas para que le muestre un mensaje al usuario se implemento un JOptionPane en el try de record().



Como podemos ver al haber una excepción se le muestra un mensaje al usuario informándole, donde además guarda las excepciones en un archivo.

**5.** ¿Es adecuado que la aplicación continúe su ejecución después de sufrir un incidente como este? ¿de qué dependería continuar o parar?

R/= En este caso es adecuado que continue la aplicación debido a que el error solo ocurre en una búsqueda, por lo que no afecta en gran parte la ejecución del programa afectando otras areas. La gravedad del error y su impacto en la funcionalidad de la aplicación son factores clave para determinar si se debe continuar o detener la ejecución.

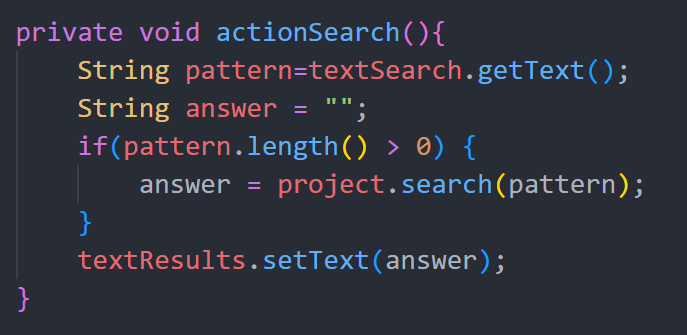
**6.** Modifiquen la aplicación para garantizar que SIEMPRE que haya un error se maneje de forma adecuada. ¿Cuál fue la solución implementada?

R/=

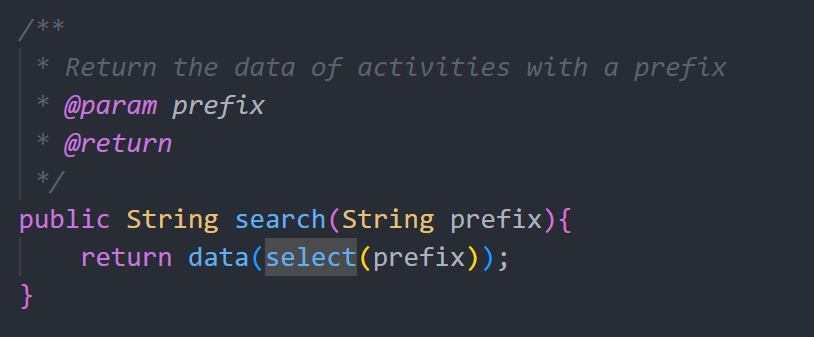
**Consultando por patrones. ¡Ahora si funciona!**

**1.** Revisen el código asociado a buscar en la capa de presentación y la capa de dominio. ¿Qué método es responsable en la capa de presentación? ¿Qué método es responsable en la capa de dominio?

R/= El metodo asociado a presentacion es actionSearch(), en donde se le indica al programa que al buscar por un prefijo se debe ejecutar el metodo search(pattern) el cual esta en project.java

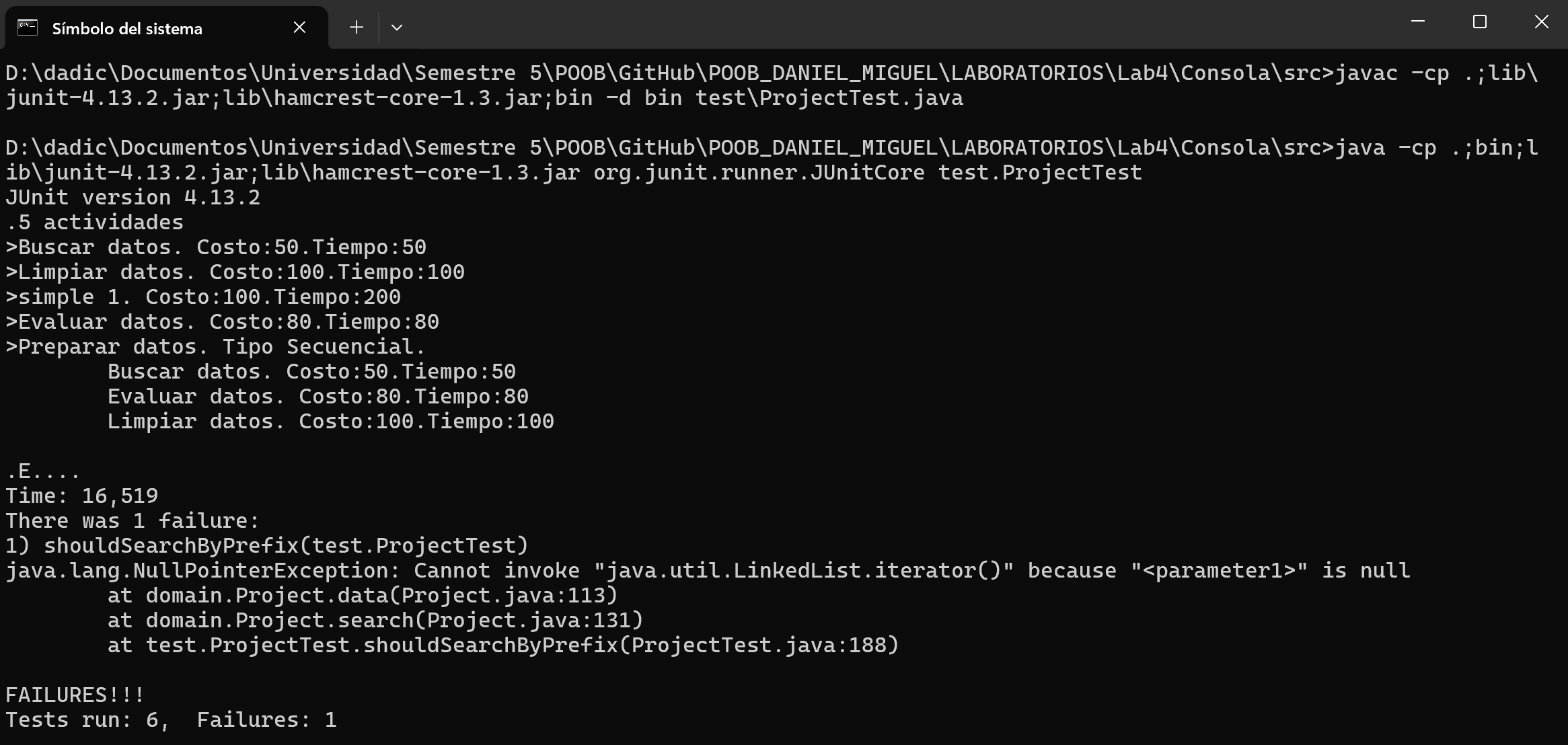


El método asociado a dominio es search(prefix), el cual llama al metodo data() para mostrar la actividad que se busca haciendo uso del método select(prefix) el cual devuelve la actividad que se busca mediante el prefijo.



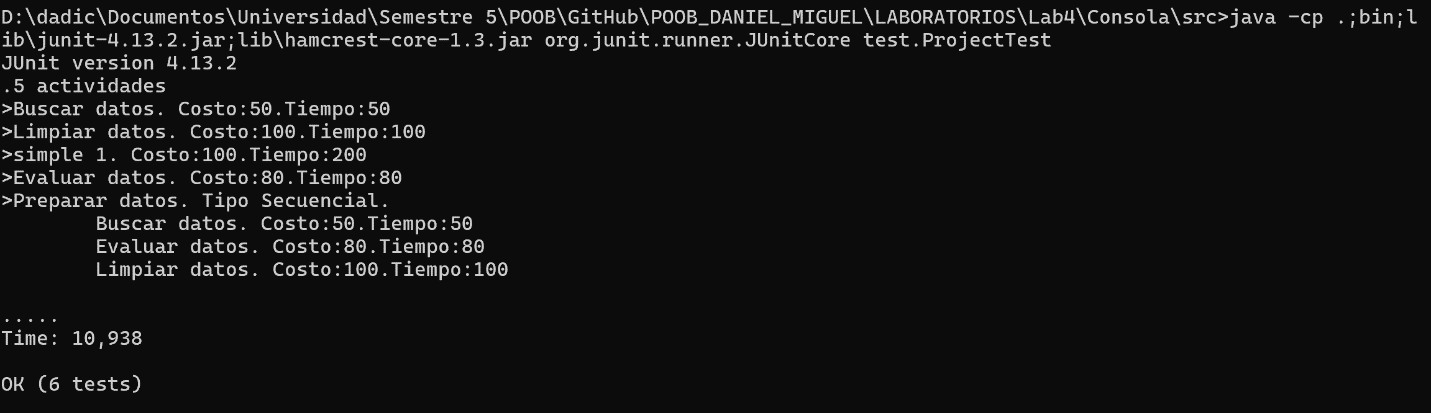
**2.** Realicen ingeniería reversa para la capa de dominio para buscar. Capturen los resultados de las pruebas. Deben fallar.

R/=



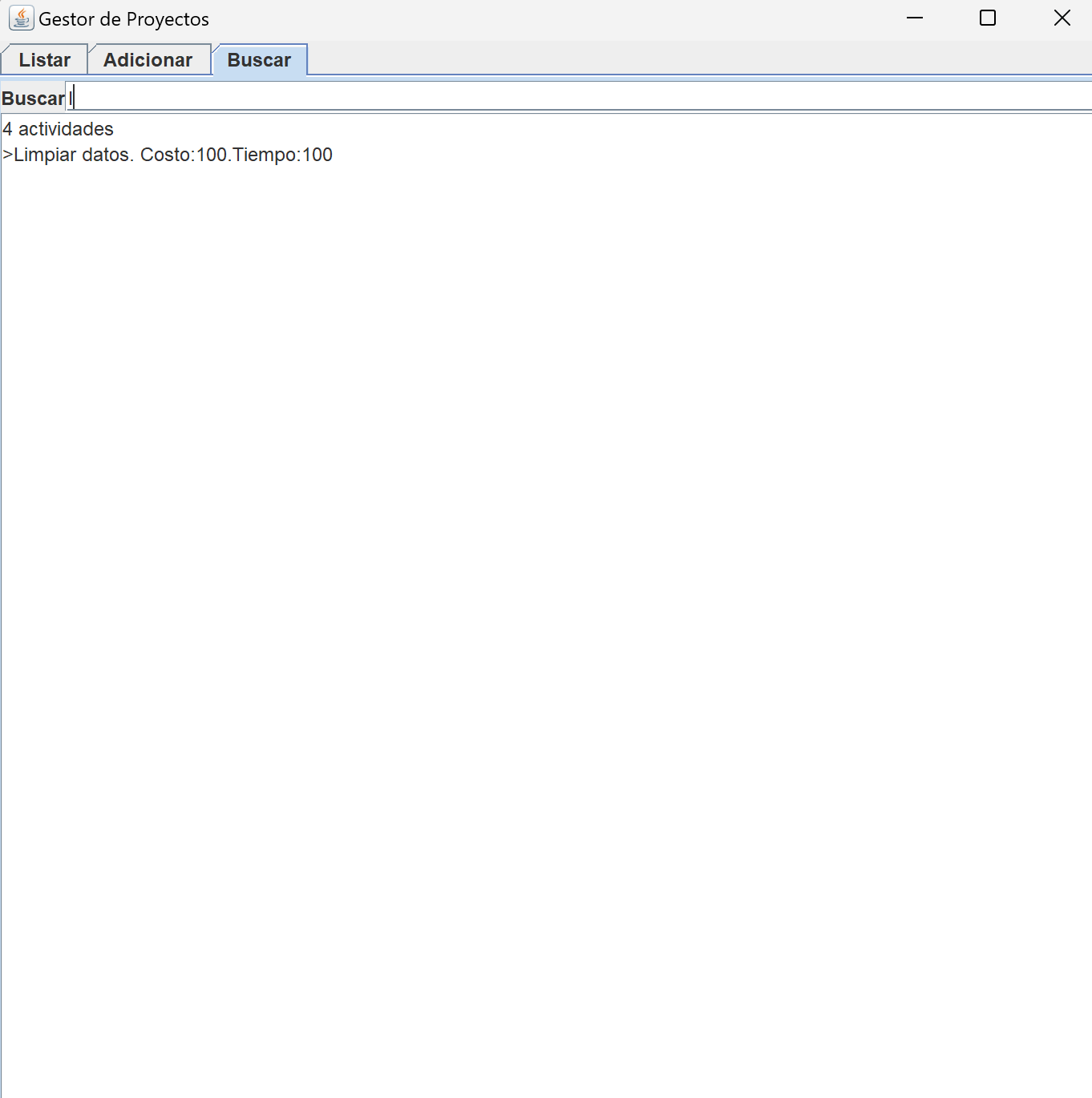
**3.** ¿Cuál es el error? Soluciónenlo. Capturen los resultados de las pruebas.

R/= El error está en la manera en la que se está recorriendo las actividades, ya que se están recorriendo por índice lo cual es incorrecto debido a que estas se almacenan en un hashMap.



**4.** Ejecuten la aplicación nuevamente con el caso propuesto. ¿Qué tenemos en pantalla? ¿Qué información tiene el archivo de errores?

R/= Al ejecutar de nuevo la aplicacion vemos que ya nos muestra la actividad.



Y en cuanto al archivo de errores no existe ya que no se generó ningún error.

**RETROSPECTIVA**

**1.** ¿Cuál fue el tiempo total invertido en el laboratorio por cada uno de ustedes? (Horas/Hombre)

R/=

Miguel: 20 hrs.

Daniel: 13 hrs.

**2.** ¿Cuál es el estado actual del laboratorio? ¿Por qué?

R/= El estado actual del laboratorio es parcialmente acabado, ya que falto el punto 6 (Consultando por patrones. ¡No funciona y queda sin funcionar!) y el punto 5 (Consultando por patrones. ¡Ahora si funciona!). Esto debido al tiempo, ya que este laboratorio se cruzó con la entrega del proyecto final y además ambas entregas se presentaron para la semana de examen, lo que dificulto mucho la realización de este.

**3.** Considerando las prácticas XP del laboratorio. ¿cuál fue la más útil? ¿por qué?

R/= Dado que el trabajo se realizó en conjunto y se enfocó en completar una aplicación de proyectos, la práctica XP más útil podría ser la Programación en Pareja o la Integración Continua. Estas prácticas permiten a los miembros del equipo colaborar estrechamente y garantizar que el código se integre y pruebe regularmente.

**4.** ¿Cuál consideran fue el mayor logro? ¿Por qué?

R/= El mayor logro podría ser completar partes significativas de la aplicación de proyectos y tener pruebas unitarias y de aceptación funcionales. Esto es importante porque demuestra progreso tangible hacia el objetivo final del proyecto.

**5.** ¿Cuál consideran que fue el mayor problema técnico? ¿Qué hicieron para resolverlo?

R/= El mayor problema técnico fue el manejo de la consola y el entendimiento de la aplicación, ya que había puntos los cuales no eran muy claros. Además, el factor más importante fue el tiempo, ya que como se explicó anteriormente no tuvimos el suficiente.

**6.** ¿Qué hicieron bien como actividades? ¿Qué se comprometen a hacer para mejorar los resultados?

R/= Realizar pruebas unitarias y de aceptación, agregar excepciones y completar partes del código son actividades positivas que contribuyen al progreso del proyecto. Para mejorar los resultados, nos comprometemos a gestionar mejor nuestro tiempo.