**PRÁCTICA**

**OBLIGATORIA**

03179150 K

tsolano8@alumno.uned.es

**PRÁCTICA**

**OBLIGATORIA**

**PARA EL**

**CURSO 2023/2024:**

Tomás Solano Campos



Índice:

Contenido

[1. CUESTIONES TEORICAS 3](#_Toc155113425)

[2. Ejemplos de ejecución para distintos tamaños del problema 4](#_Toc155113426)

# CUESTIONES TEORICAS

1. Indica y razona sobre el coste temporal y espacial del algoritmo

En el peor caso el coste sería el coste del árbol completo ,cada nodo del árbol se divide en cuatro direcciones de búsqueda .A su vez el numero de niveles del árbol tendrá que ser las casillas del tablero ,como hay 2 tableros (la matriz de booleanos y la del camino) cada una con n casillas ,será N^2.

Luego el coste máximo será O(4^(N^2)).

1. Explica qué otros esquemas pueden resolver el problema y razona sobre su idoneidad.

Para realizar este problema,podríamos usar programación dinámica y ramificación y poda

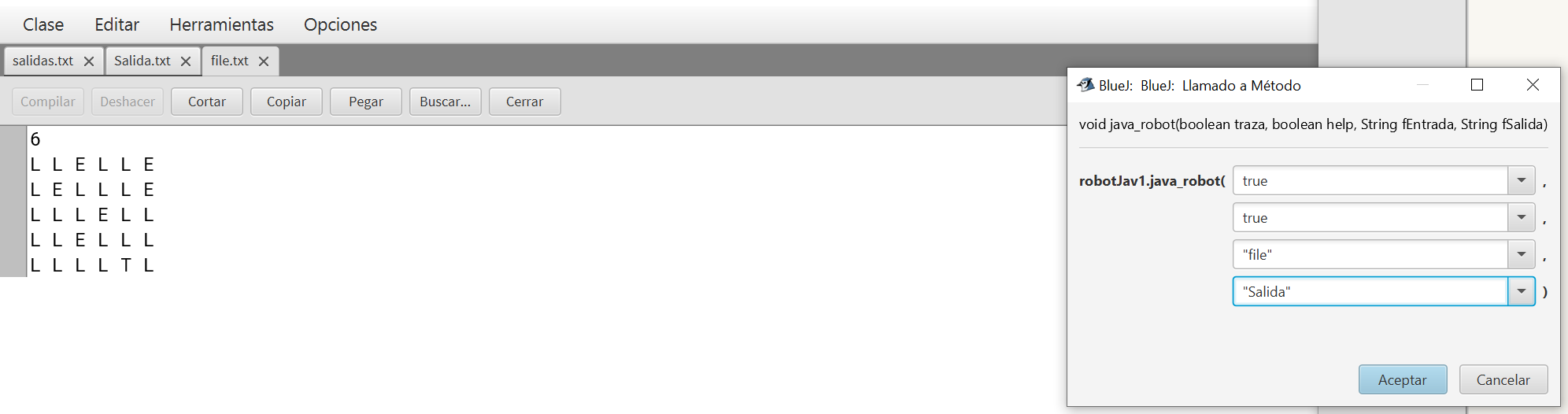
Voraz no valdría porque no hay nada que maximizar/minimizar ,además no podrías formular una solución para los pasillos sin salida sin volver hacia atrás.

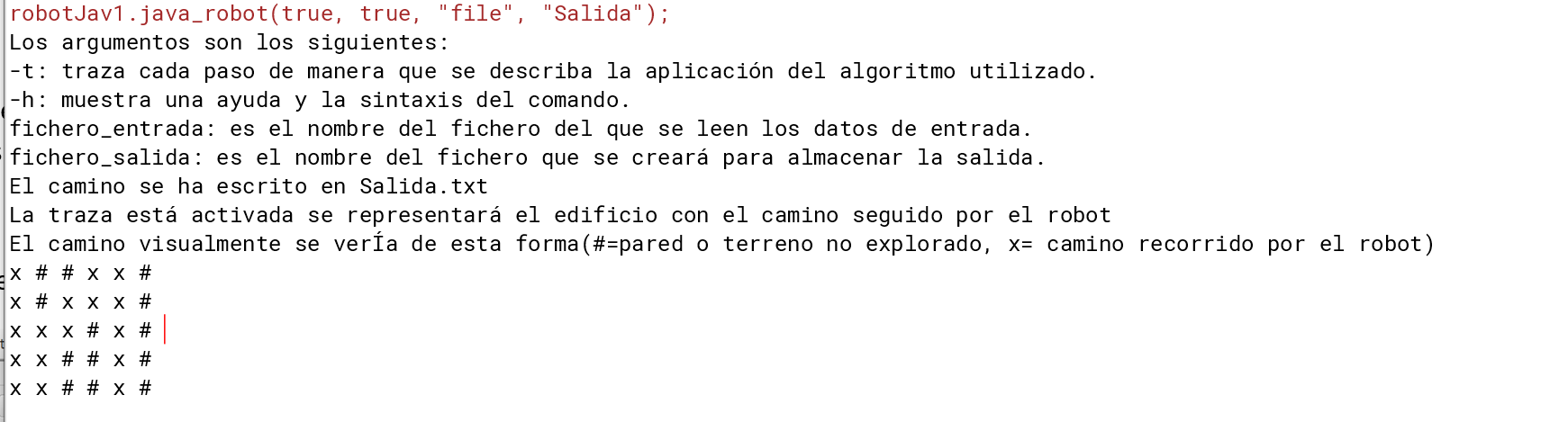
Divide y Vencerás tampoco valdría ya que descartaría algunos caminos,evitando el recorrido en profundidad ,no podrías dar una solución si divides cada casilla en un subproblema ,ya que el tornillo solo se encontraría en una casilla y no se sabría el recorrido hecho.

Programacion dinámica al guardar las etapas intermedias si que podría dar con una solución ,pero sería altamente ineficiente,ya que tendría que repetir el árbol multiples veces.

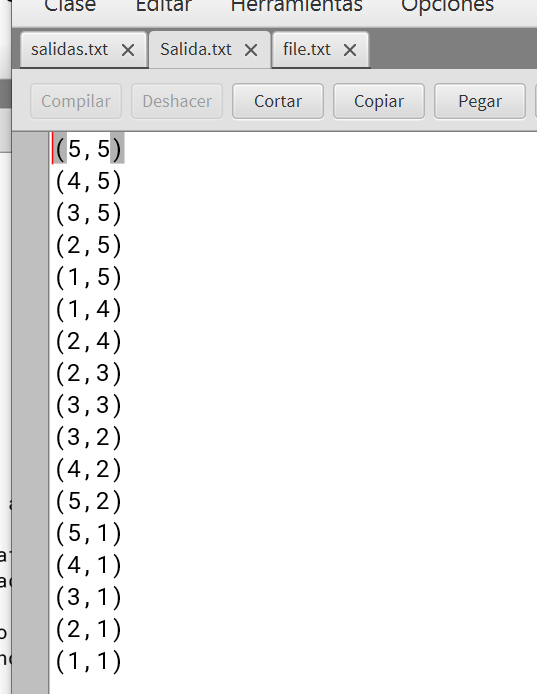
Por último, ramificación y poda también funcionaría ,ya que hace un recorrido en profundidad ,pero no tendrías que podar nada ,quizá únicamente las casillas que son E,lo que lo haría básicamente un vuelta atrás.

# Ejemplos de ejecución para distintos tamaños del problema

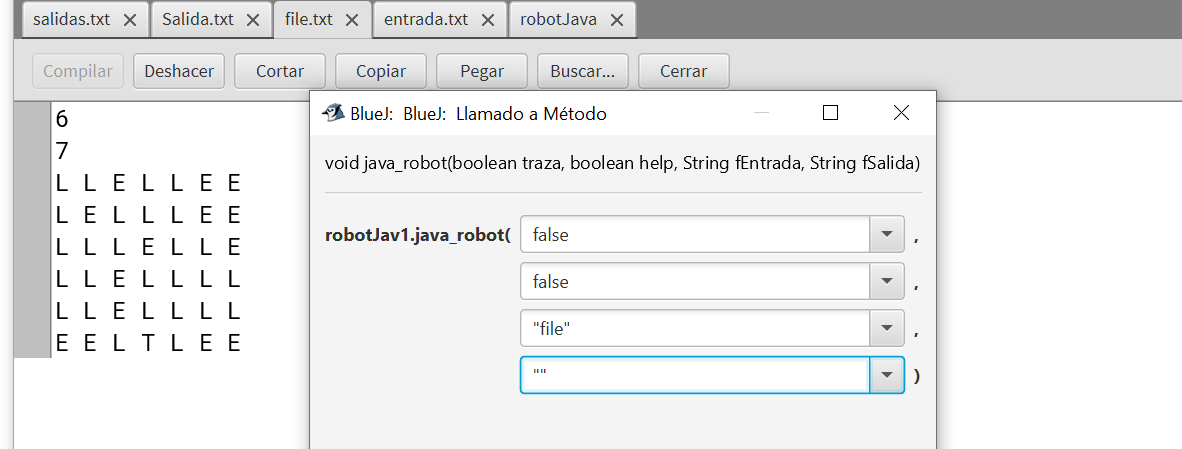
Iteración 1 (Prueba de un archivo coherente usando entradas y salidas



Resultado:



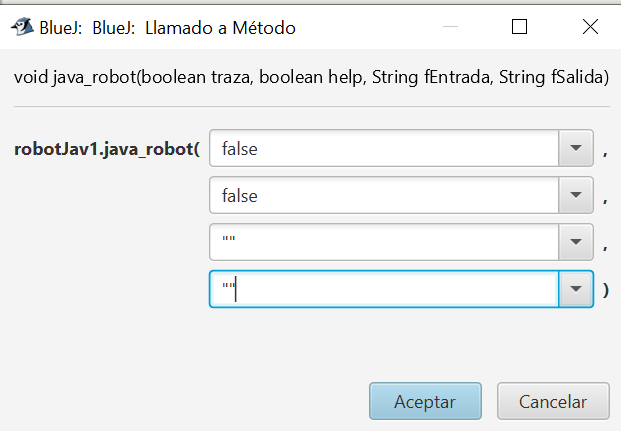
Iteración 2(Prueba de un archivo coherente usando solo el fichero de entrada)

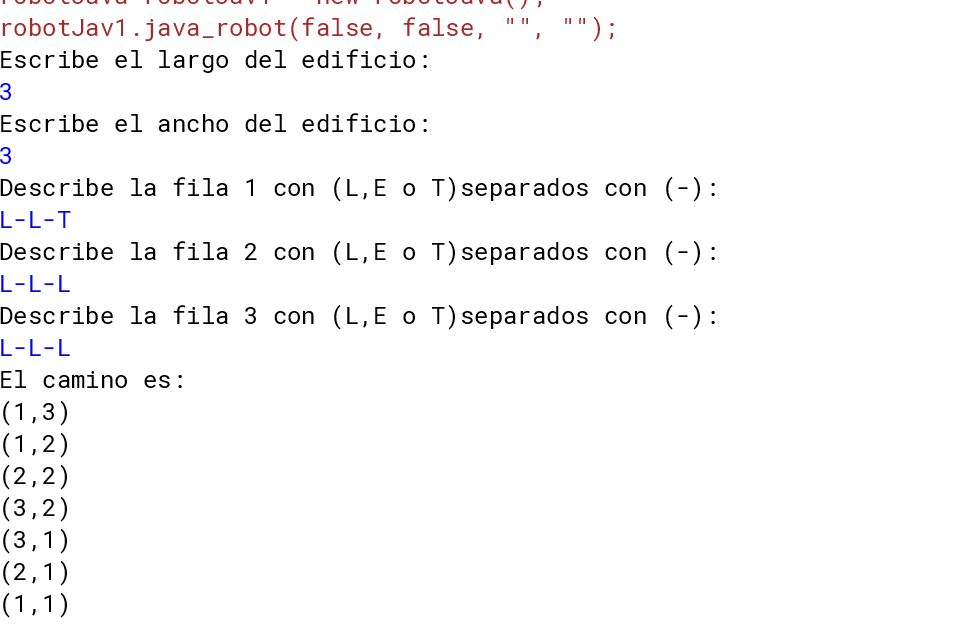


Da como resultado:

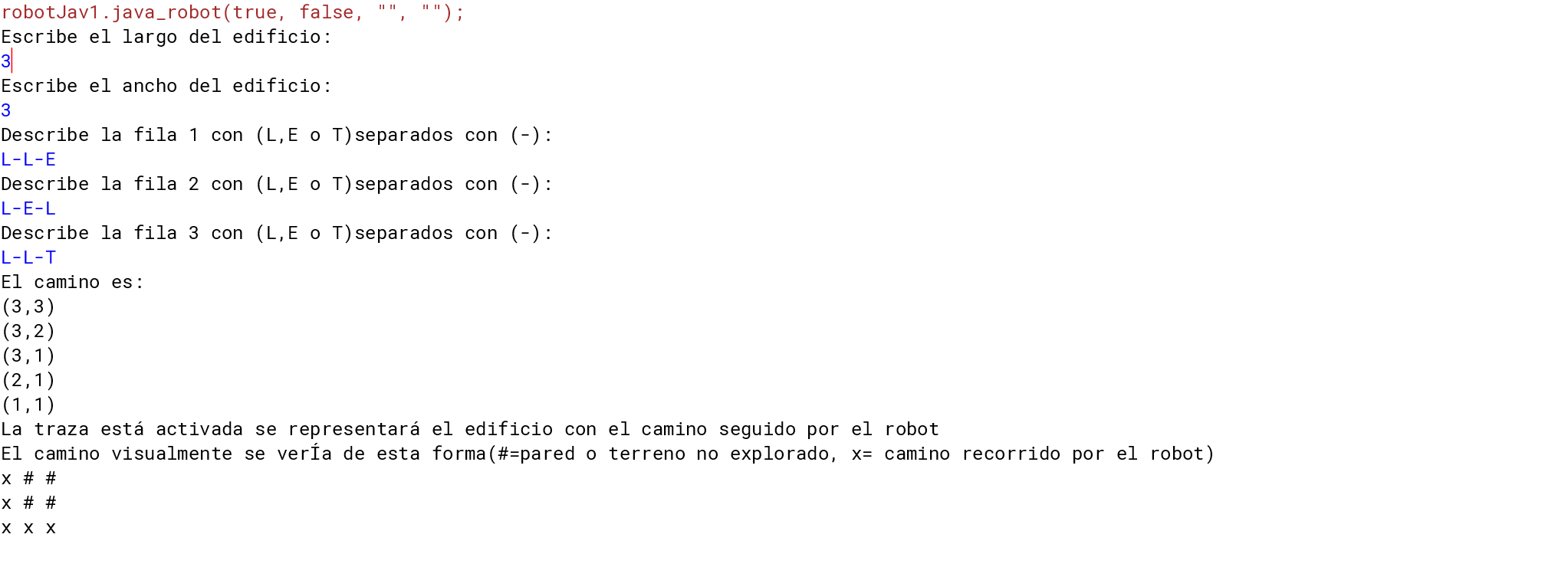


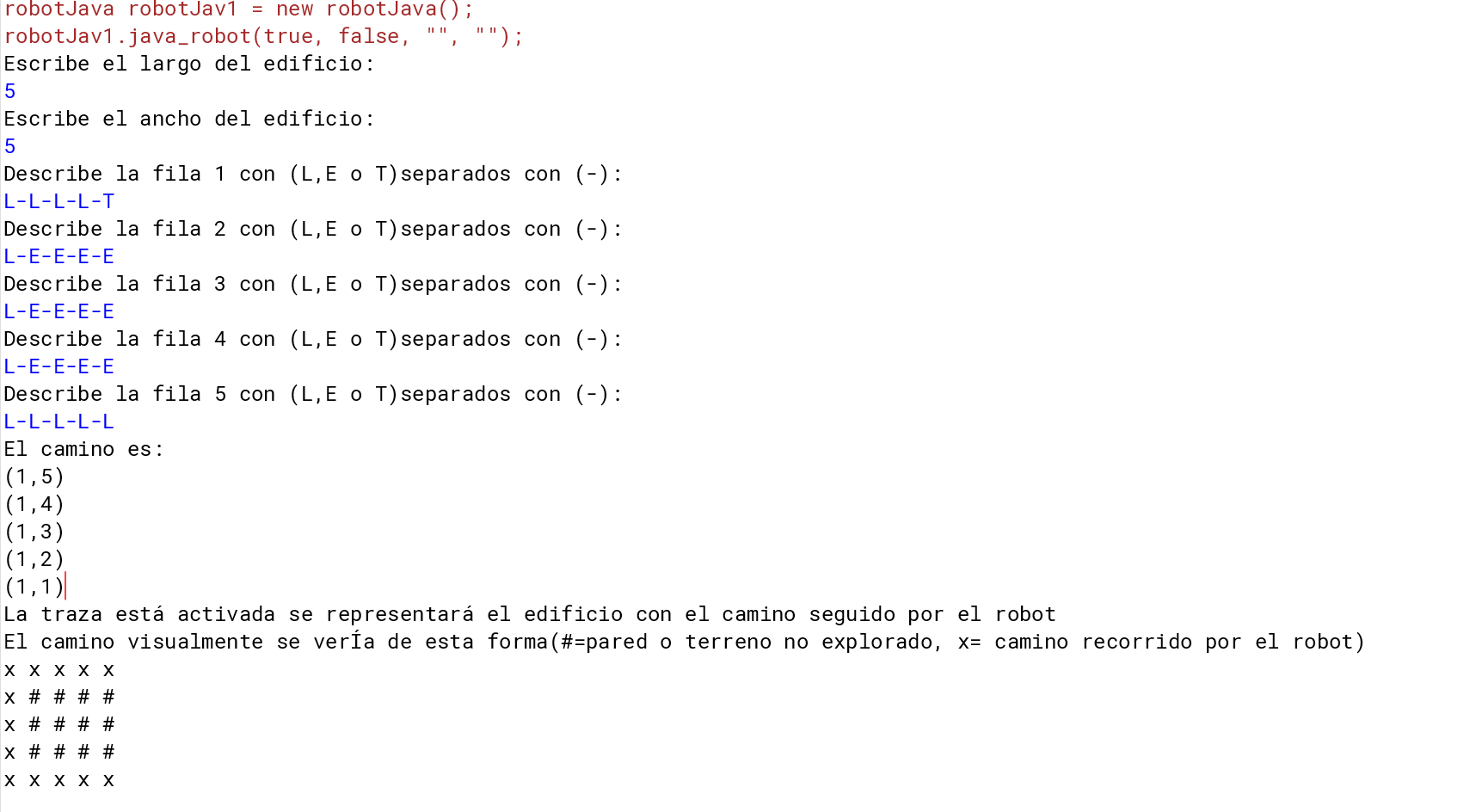
Iteración 3(Prueba sin usar ningún archivo,además sin usar trazas ni ayudas)



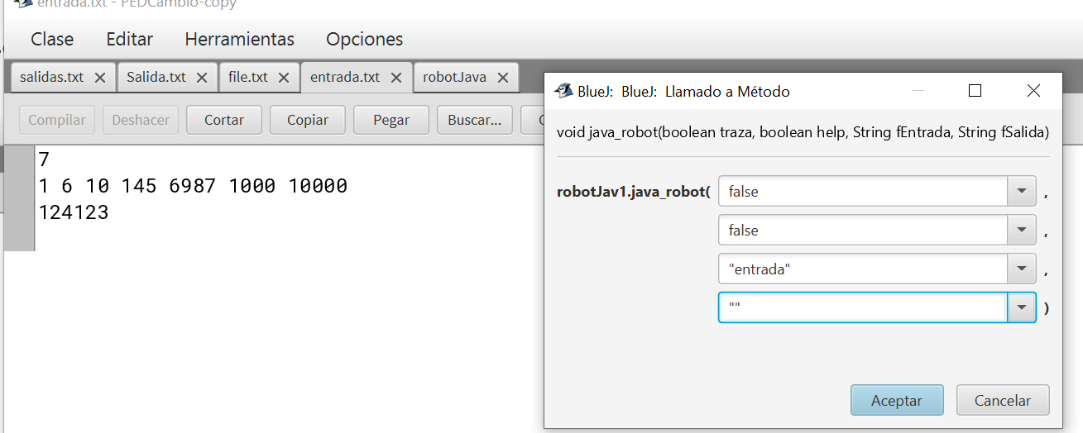


Iteración 4(Pruebas sin usar ningún archivo,con traza pero no ayuda)

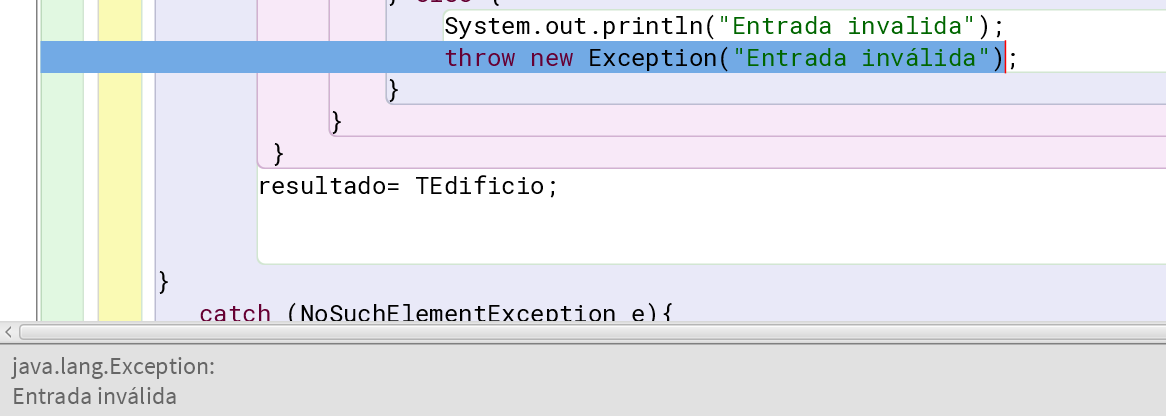


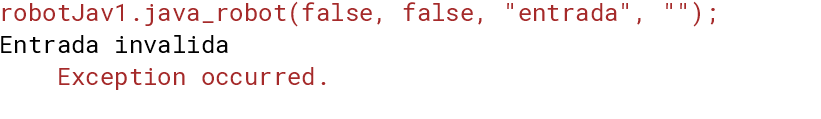


Iteración 4(Con archivo erroneo o vacío)

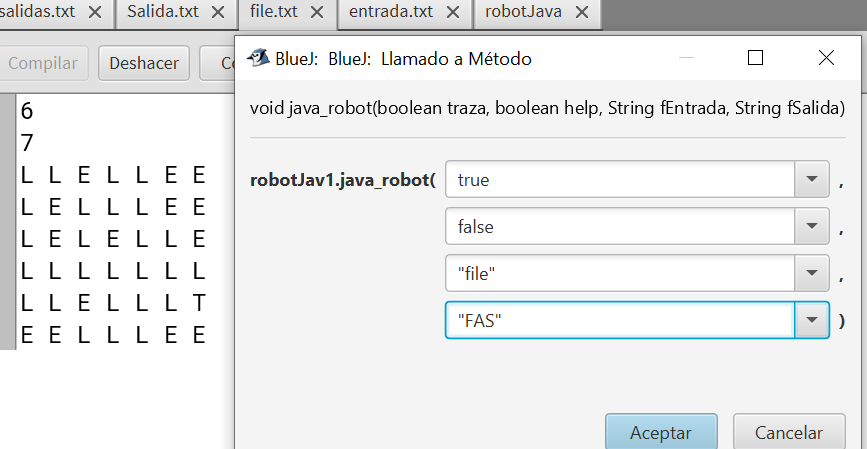


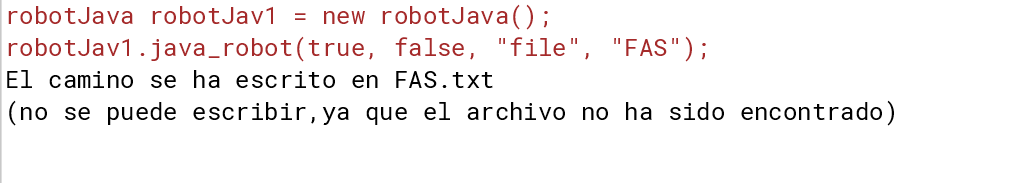
Te lanza una excepción en el compilador y en el terminal





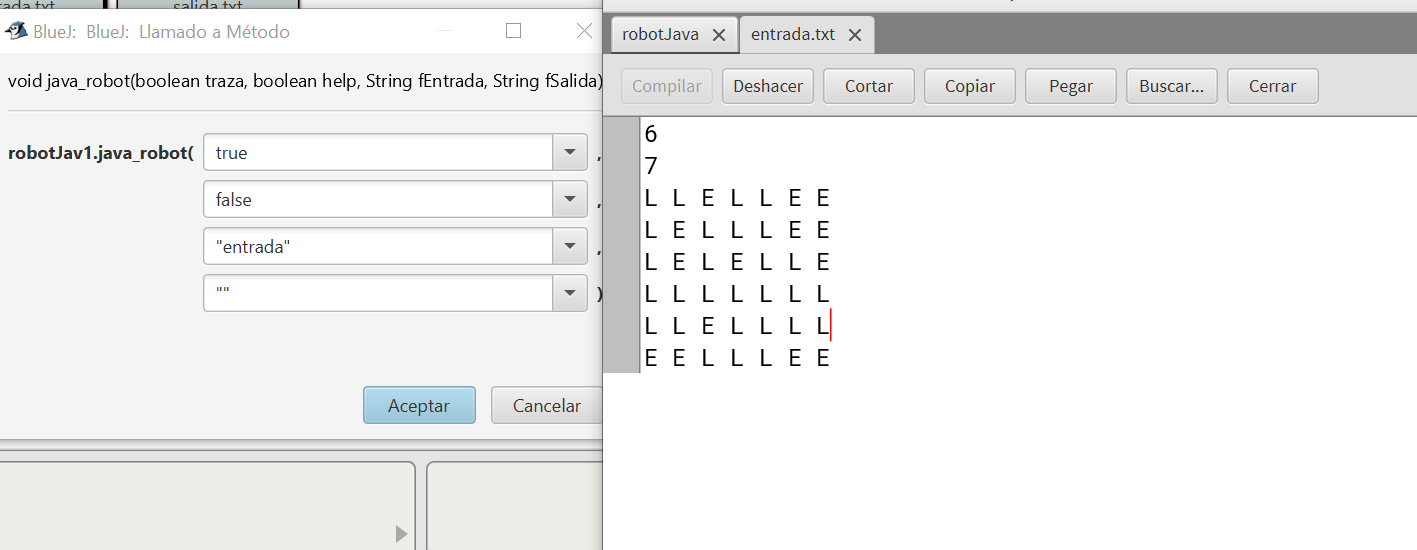
Iteración 5 (Con Archivo de Salida erroneo):





Como se ve la traza no se ejecuta ya que el programa finaliza antes de llegar a traza

Iteración 6(sin tornillo)



Resultado:

