

# ENUNCIADO DE LA PRÁCTICA DE LA ASIGNATURA DE INTELIGENCIA ARTIFICIAL Y SISTEMAS INTELIGENTES 2022/2023

Fecha límite de entrega: 18 de abril

versión 1.1 (27 enero 2023)

## LABECOIN

Se pide diseñar e implementar un sistema relacionado con los movimientos y acciones de un robot que busque un camino para salir de un laberinto. Se proporciona un laberinto en un tablero de 10 x 10, que se da como parámetro de entrada, y un mínimo valor de monedas (**precio de salida**) que se deben conseguir en el camino. Como salida de la aplicación, deben mostrarse los movimientos del robot hasta la salida. El tablero de entrada podrá ser diferente en cada caso, aunque el tamaño es siempre de 10 x 10. Las celdas tienen un valor asignado que se corresponden con:

0 – CELDA VACÍA    9 – MURO    8 – ROBOT    7 – SALIDA    1-6 – MONEDAS

El movimiento del robot puede ser **Arriba**, **aBajo**, **Derecha**, **Izquierda**, y las diagonales (**AI**, **AD**, **BD**, **BI**). No se podrá mover a una casilla que haya un muro. Cuando se mueva a una casilla con monedas, debe sumarse esa cantidad al acumulado de monedas (**cartera**) del camino. Sólo se suma una vez la moneda al pasar por esa casilla. Es decir, que cuando se pasa por una casilla con la moneda, se “retira” y se pasa a la cartera.

Se deben utilizar técnicas de búsquedas de forma que proporcione una solución, consistente en indicar la secuencia de acciones (movimientos) del robot. Esta solución debe cumplir que, al llegar a la salida, el acumulado de monedas (cartera) debe ser igual o superior al precio indicado. Se parte desde el supuesto de que se dispone de muy poco tiempo para encontrar la solución, por lo que interesa utilizar **búsquedas con heurísticas**, y se quiere encontrar una solución que no tenga muchos movimientos. Siempre debe buscarse una solución para que supere la condición de que  $\text{cartera} \geq \text{precio}$ .

*Ejemplo con precio=12 (no se muestran los ceros):*

|   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|
| 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 |
| 9 | 2 | 9 |   | 6 |   |   |   | 1 | 9 |
| 9 |   | 9 |   |   | 9 | 9 | 9 | 9 | 9 |
| 9 | 1 |   | 8 | 9 |   | 9 |   |   | 9 |
| 9 |   | 9 | 9 | 9 |   |   | 4 |   | 9 |
| 9 |   |   |   |   | 9 | 9 | 9 | 9 |   |
| 9 | 1 | 9 |   |   |   |   |   | 1 | 9 |
| 9 | 1 | 9 | 9 | 9 |   | 9 | 9 | 9 | 9 |
| 9 | 1 | 6 | 6 | 9 |   |   |   | 3 | 9 |
| 9 | 9 | 9 | 9 | 9 | 9 | 7 | 9 | 9 | 9 |

Una posible solución para este ejemplo podría ser:

**AD, A, B, BD, BD, D, I, BI, B, B, B, D, D, D, I, BI**

### Aclaraciones:

- El fichero de entrada tendrá un formato de texto simple. La primera línea tendrá el valor del precio. Luego, contendrá diez filas correspondiente al laberinto, formado por los números separados por comas. Se supone que el fichero de entrada estará siempre bien formado. En el borde, siempre habrá muro o la casilla de salida. No se necesitan implementar funciones de comprobación.

## Requisitos de la práctica:

- Los equipos de la práctica lo deben formar como máximo 3 alumnos.
- Se puede desarrollar en cualquier lenguaje de programación que tenga su compilador y entorno instalado en las salas de prácticas de la asignatura.
- Como mínimo, la práctica debe leer un fichero llamado LABECOIN.TXT , y mostrar como salida: la técnica de resolución empleada, la secuencia de movimientos que forman la solución, el tiempo empleado en la resolución y el número de nodos del árbol de búsqueda generados en memoria.
- La práctica debe intentar resolver el tablero con, al menos, dos técnicas de resolución basadas en resolución de problemas con heurística de inteligencia artificial, y deben hacerse pruebas con al menos 10 ficheros de tableros.
- El proyecto debe poder compilarse y ejecutarse en los entornos y compiladores **instalados en la salas de prácticas** de la asignatura en la Escuela Politécnica.
- Se debe entregar un ZIP con: los ficheros necesarios para ejecutar la aplicación, y la documentación en un documento PDF con el formato que aparece en la plantilla de la práctica.
- A final de curso, se realizará la evaluación de la práctica donde uno (escogido al azar por el profesor) de los componentes del equipo expondrá de forma pública el funcionamiento y resultados del proyecto. Se podrá preguntar por el código o pedir desarrollar modificaciones al código.
- Cualquier coincidencia en algoritmos, estructura, o enfoque de todo o en parte de los ficheros de código fuente con otros grupos, implicará el suspenso de la asignatura en la parte práctica y teórica, y de todos los miembros de los equipos implicados. Si se hubiera aprobado en una convocatoria anterior, se podrá abrir un acta cerrada y generar un expediente disciplinario.
- Los requisitos que debe cumplir la práctica es lo que se ha descrito anteriormente, pero puede entregarse **además** otra versión con modificaciones de la práctica que pueden subir la calificación. Ejemplos de mejoras (se pueden combinar entre ellas y pueden ser otras): Hacer un tablero con más dimensiones; poner múltiples salidas; poner pesos de penalización al pasar por alguna casilla; poner casillas con valor de monedas aleatorio, etc.

## Evaluación:

- Para sacar como mínimo un aprobado en la práctica, es obligatorio: (1) que estén implementados al menos dos algoritmos de resolución **con heurística**, que muestren como salida la secuencia solución, tiempo y nodos generados; (2) que se entregue la documentación y ficheros fuente como se especifica en la plantilla; y (3) en la presentación, el alumno que la exponga sepa explicar cómo se ha desarrollado el proyecto y las técnicas usadas.
- En la calificación, se tendrá en cuenta principalmente: la implementación de las técnicas de resolución empleadas y las mejoras (60% aprox), la calidad de la documentación (20% aprox), la claridad de la exposición en la presentación (20% aprox).
- Si los equipos, en vez de tres miembros, lo forman dos, a la nota se le restará 0,5 puntos. Si lo forma un solo miembro, se le restará 1 punto.