



Universidad de Granada

[decsai.ugr.es](http://decsai.ugr.es)

# Inteligencia Artificial

## Seminario 2

### Agentes Reactivos / Deliberativos



DECSAI

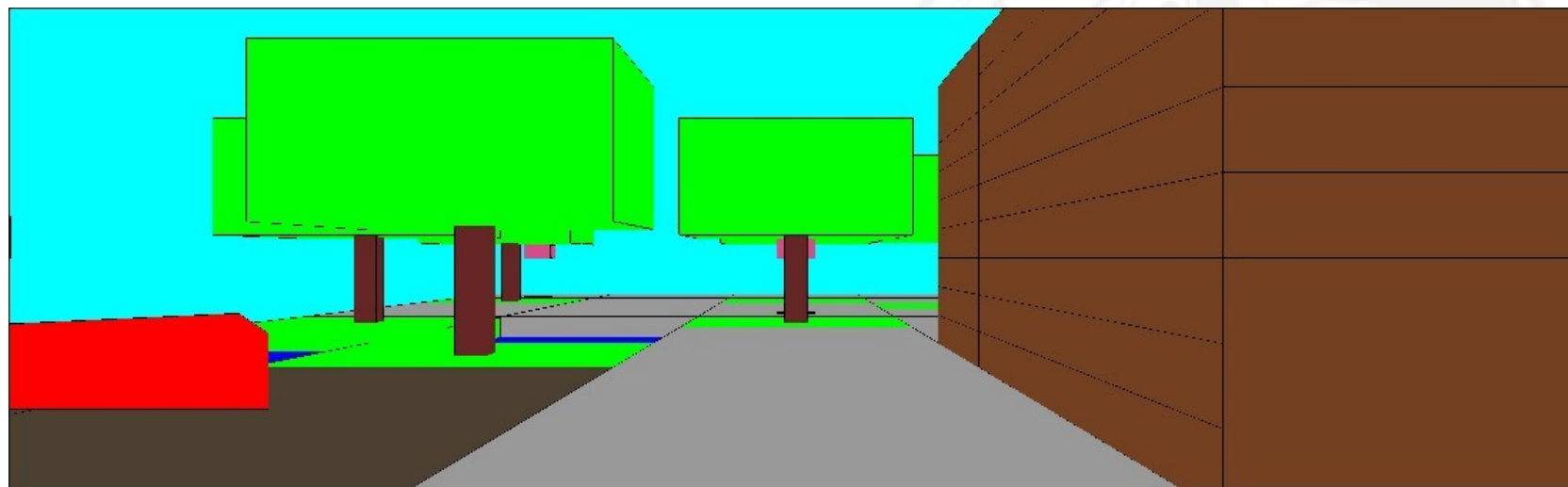
**Departamento de Ciencias de la  
Computación e Inteligencia Artificial**

- **Introducción**
- **Presentación del problema**
- **Los extraños mundos de BelKan**
- **Objetivos**
- **Software**
- **Método de evaluación y entrega de prácticas**

- **Introducción**
- **Presentación del problema**
- **Los extraños mundos de Belkan**
- **Objetivos**
- **Software**
- **Método de evaluación y entrega de prácticas**



- Diseñar e implementar un agente reactivo y deliberativo, capaz de percibir el ambiente y actuar considerando una representación de las consecuencias de sus acciones y siguiendo un proceso de búsqueda.



- En esta práctica se diseñará e implementará un agente reactivo y deliberativo basado en los ejemplos del libro *Stuart Russell, Peter Norvig, “Inteligencia Artificial: Un enfoque Moderno”, Prentice Hall, Segunda Edición, 2004.*
- El simulador que utilizaremos fue inicialmente desarrollado por el profesor Tsung-Che Chiang de la NTNU (Norwegian University of Science and Technology, Trondheim), pero la versión sobre la que se va a trabajar ha sido desarrollada por los profesores de la asignatura.

- Originalmente, el simulador estaba orientado a experimentar con comportamientos en aspiradoras inteligentes.
- En su versión más simple, una aspiradora inteligente presenta un comportamiento **reactivo** puro: busca suciedad, la limpia, se mueve, detecta suciedad, la limpia, se mueve, y continúa con este ciclo hasta que se cumple alguna condición de parada.



- Otras versiones más sofisticadas permiten al robot recordar (mediante el uso de representaciones icónicas como mapas), lo cual permite que el aparato ahorre energía y sea más eficiente en su trabajo.
- Finalmente, las aspiradoras más elaboradas pueden, además de todo lo anterior, planificar su trabajo de modo que se pueda limpiar la suciedad en el menor tiempo posible y de la forma más eficiente. Estas últimas pueden ser catalogadas como **agentes deliberativos**.





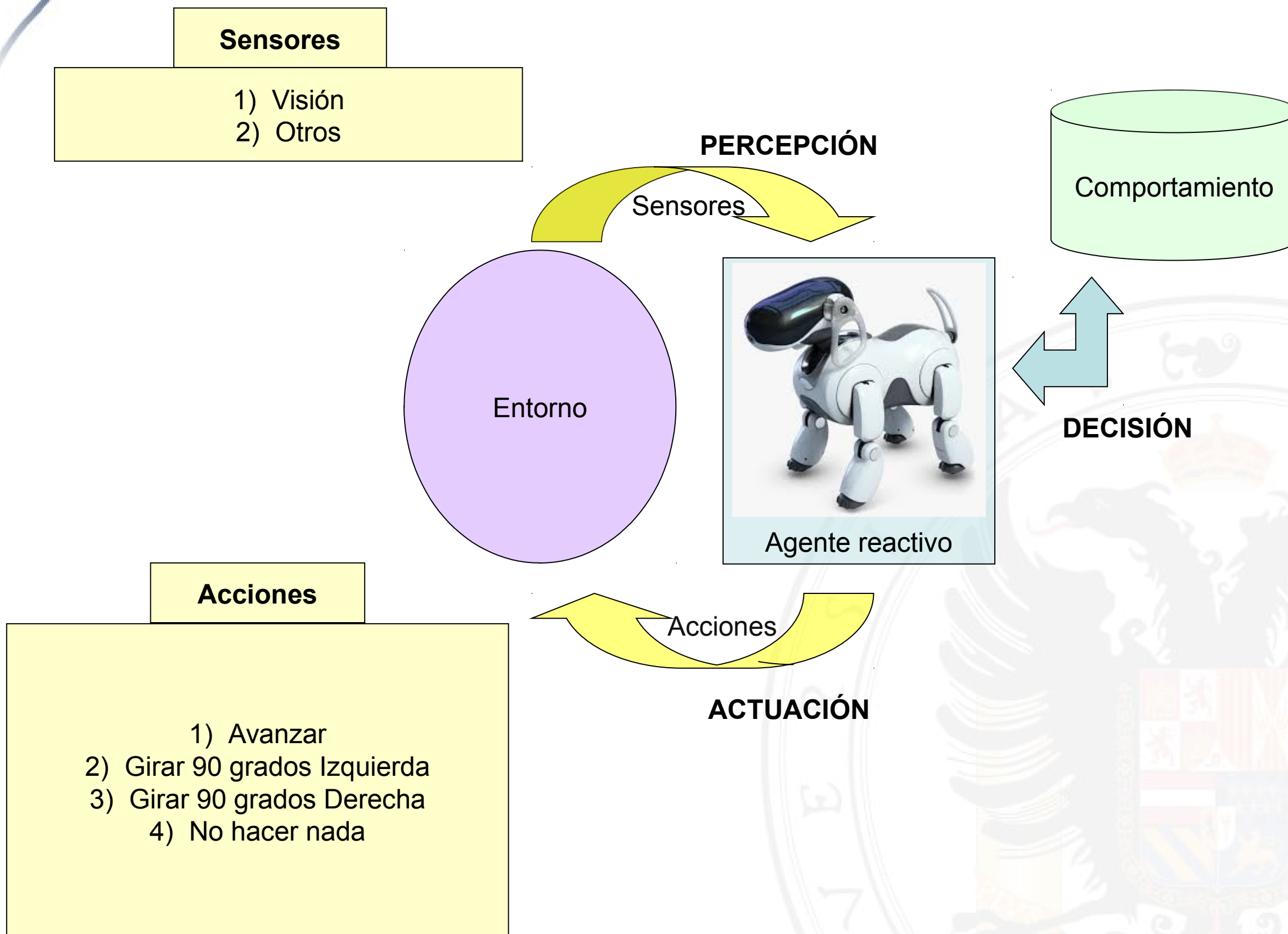
Esta práctica cubre los siguientes objetivos docentes:

- Conocer la representación de problemas basados en estados (estado inicial, objetivo y espacio de búsqueda) para ser resueltos con técnicas computacionales.
- Entender que la resolución de problemas en IA implica definir una representación del problema y un proceso de búsqueda de la solución.
- Analizar las características de un problema dado y determinar si es susceptible de ser resuelto mediante técnicas de búsqueda. Decidir en base a criterios racionales la técnica más apropiada para resolverlo y saber aplicarla.
- Entender el concepto de heurística y analizar las repercusiones en la eficiencia en tiempo y espacio de los algoritmos de búsqueda.
- Conocer distintas aplicaciones reales de la IA. Explorar y analizar soluciones actuales basadas en técnicas de IA.



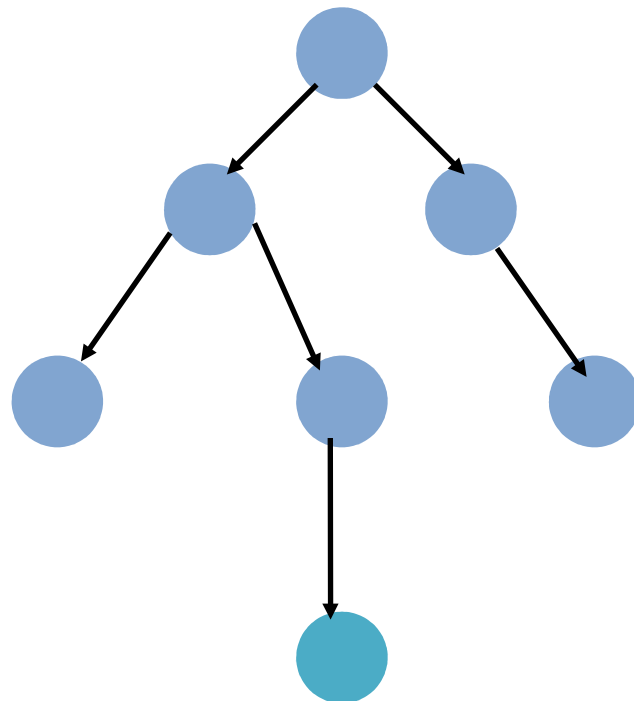
- Introducción
- **Presentación del problema**
- Los extraños mundos de Belkan
- Objetivos
- Software
- Método de evaluación y entrega de prácticas





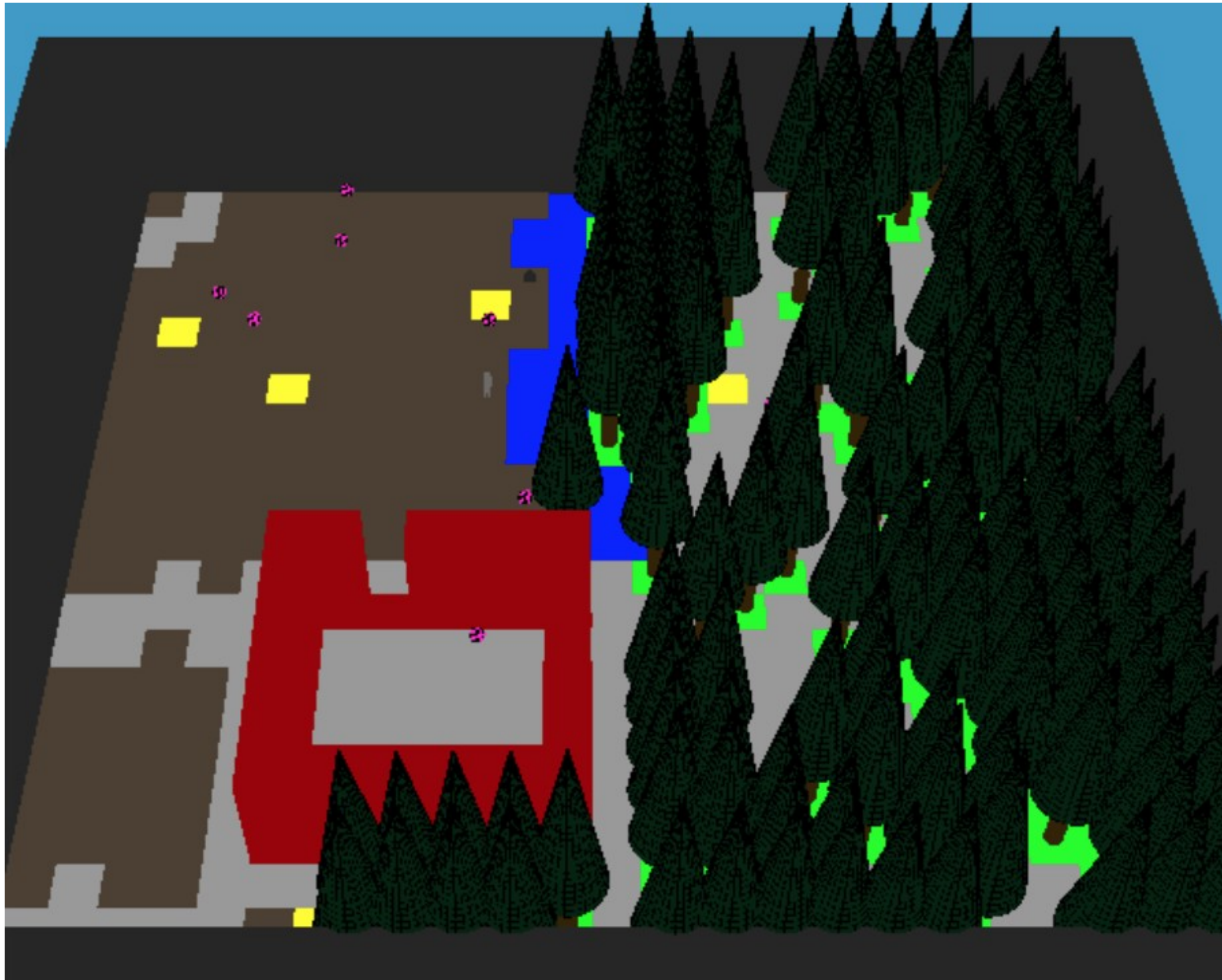


**PROBLEMAS  
DE  
BUSQUEDA**

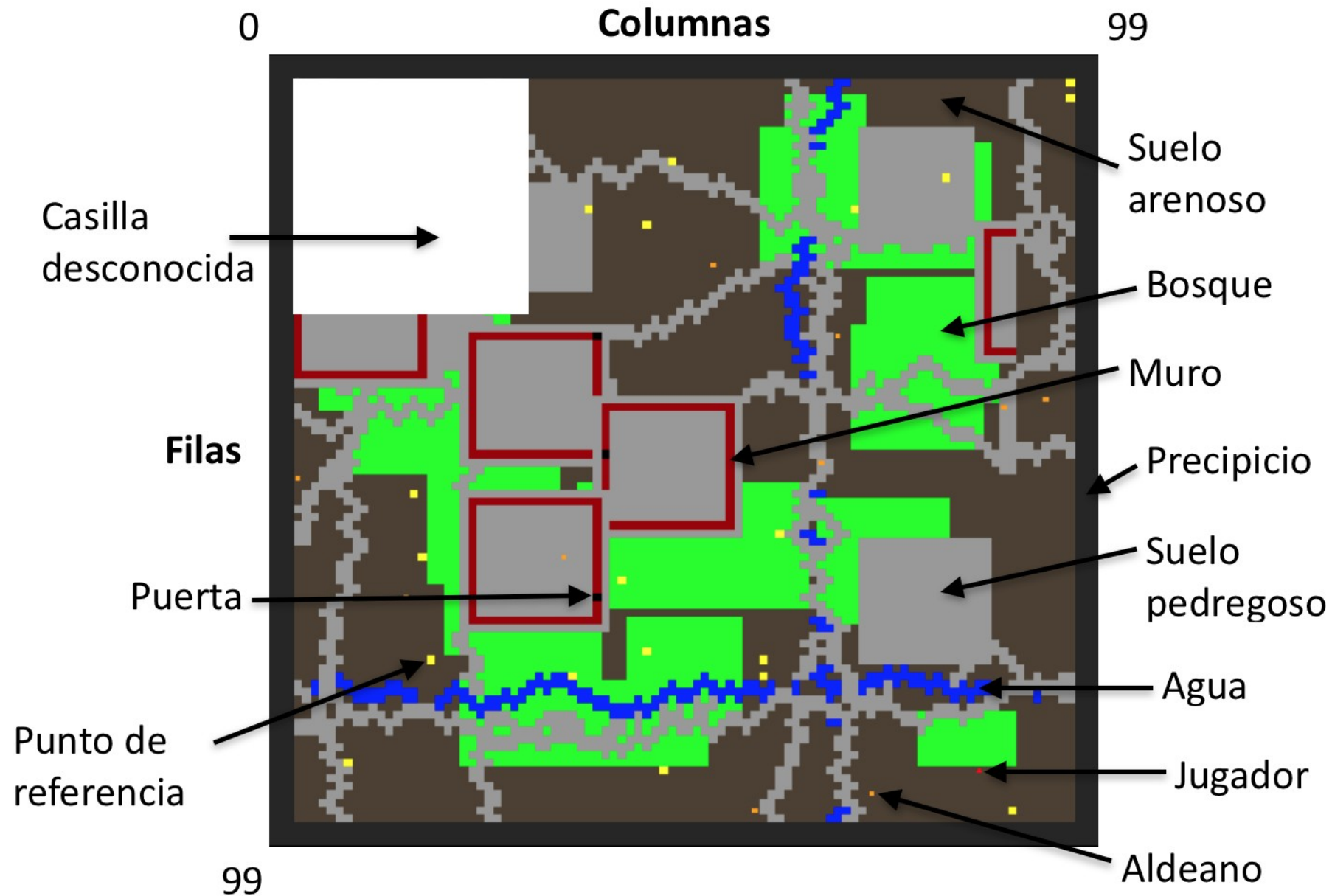


- Introducción
- Presentación del problema
- **Los extraños mundos de Belkan**
- Objetivos
- Software
- Método de evaluación y entrega de prácticas

### Los extraños mundos de BelKan

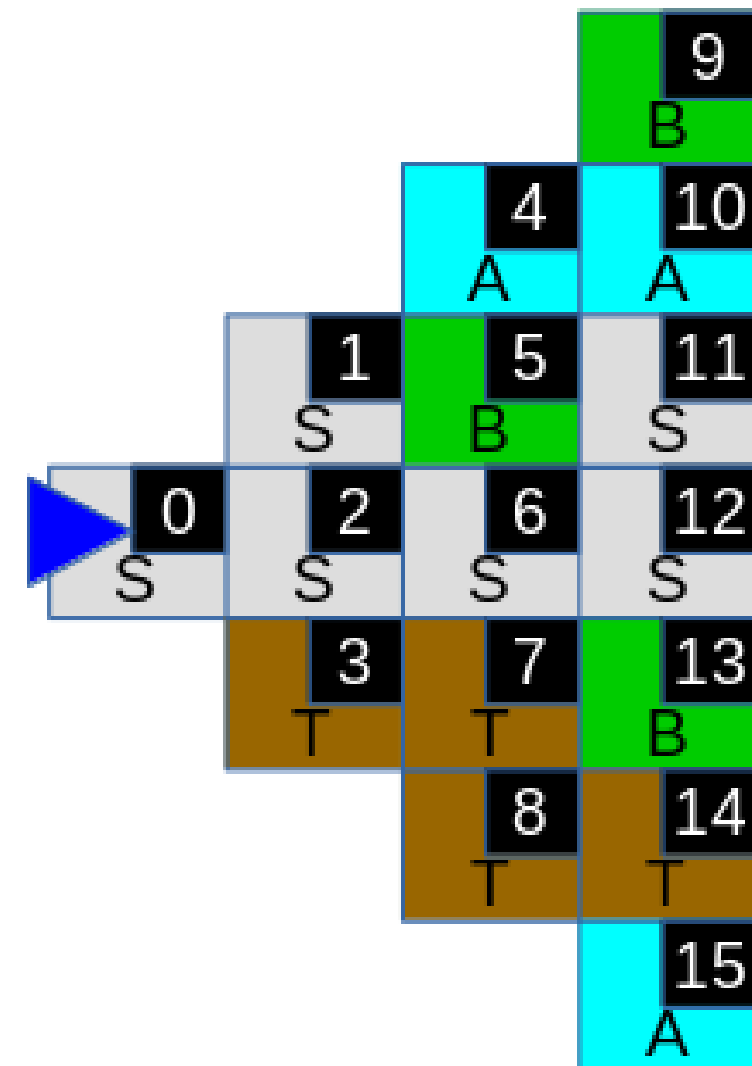






- El personaje muere cuando cae por un precipicio y no puede atravesar los muros.

- Sensor de choque (colision)
- Sensor de vida (reset)
- Sensores de mensaje (mensajeF, mensajeC)
- Sensor de destino (destinoF, destinoC)
- Sensor de comportamiento (nivel)
- Sensor de tiempo consumido (tiempo)



TERRENO / SUPERFICIE



Nuestro personaje puede realizar varias acciones distintas durante el juego:

- *actFORWARD*: le permite avanzar a la siguiente casilla del mapa siguiendo su orientación actual. Para que la operación se finalice con éxito es necesario que la casilla de destino sea transitable para nuestro personaje.
- *actTURN\_L*: le permite mantenerse en la misma casilla y girar a la izquierda 90º teniendo en cuenta su orientación.
- *actTURN\_R*: le permite mantenerse en la misma casilla y girar a la derecha 90º teniendo en cuenta su orientación.
- *actIDLE*: pues como su nombre indica, no hace nada.

El coste de las acciones es de 1 medido en unidades de tiempo o instantes de simulación como criterio general, pero la acción de avanzar tiene un coste superior sobre algunos tipos de terreno, en concreto

- Coste 2 si el suelo es arenoso.
- Coste 5 si es bosque.
- Coste 10 si es agua.

- Introducción
- Presentación del problema
- Los extraños mundos de Belkan
- **Objetivo**
- Software
- Método de evaluación y entrega de prácticas



- El objetivo de esta práctica es dotar de un comportamiento inteligente a nuestro personaje usando un agente reactivo/deliberativo para definir las habilidades que le permitan dentro del juego según el nivel seleccionado.
- Para que sea más fácil resolver esta práctica, se han diseñado dos niveles con 4 comportamientos:
  - **NIVEL 1: Agente deliberativo**
    - **Comportamiento 1: Resolver el nivel usando búsqueda en profundidad**
    - **Comportamiento 2: Resolver el nivel usando búsqueda en anchura**
    - **Comportamiento 3: Resolver el nivel usando búsqueda de coste uniforme**
  - **NIVEL 2: Agente reactivo/deliberativo**
    - **Comportamiento 4: Resolver el nivel**

### Nivel 1: Agente puramente deliberativo

- Nuestro personaje se encuentra en un mundo inmutable y completamente conocido.
- Nuestro personaje aparecerá de forma aleatoria sobre un mundo de BelKan concreto conociendo su posición (a través de los sensores **mensajeF** y **mensajeC** que se activa cuando el jugador debe decidir su primera acción) y orientación sobre el mapa (siempre mira al norte al iniciar).
- El objetivo es construir un camino que le permita llegar hasta la posición de destino (marcada en los sensores destinoF y destinoC).
- Cada una de los tres comportamientos incluidos en este nivel implican resolver el problema usando tres algoritmos de búsqueda distintos.
- El comportamiento 1, búsqueda en profundidad, se encuentra incluido en el software inicial. Se incluye para que el estudiante lo use como referencia para la realización del resto de los comportamientos.

### Nivel 2: Agente reactivo/deliberativo con mapa desconocido

- En este nivel la cosa se complica ya que no nos han dado el mapa del mundo y no tenemos idea de donde estamos! Solo sabemos que miramos al norte.
- Deberemos ir descubriendo el mapa poco a poco. Pero para empezar debemos saber en que fila y columna estamos. Para ello debemos recorrer el mapa hasta encontrar una casilla PK (las casillas amarillas) que nos indicará la posición real en fila y columna y desde ese momento podremos ir construyendo nuestro propio mapa a medida que nos dirigimos al destino.
- En la parte deliberativa de este nivel podemos usar cualquiera de los algoritmos de búsqueda implementados para el nivel 1 o incluir uno nuevo exclusivamente para usarlo en este nivel.

- Para optar por este nivel es necesario tener hecho el nivel anterior. En este nivel cada vez que lleguemos al destino, éste cambiará de objetivo a alcanzar. Es decir, deberemos tratar de conseguir llegar a la mayor cantidad de destinos posible utilizando como máximo 3000 acciones o 300'' totales (lo que suceda antes).
- También es importante recordar que el jugador no debe morir ni chocarse con ningún objeto inmóvil del mapa.

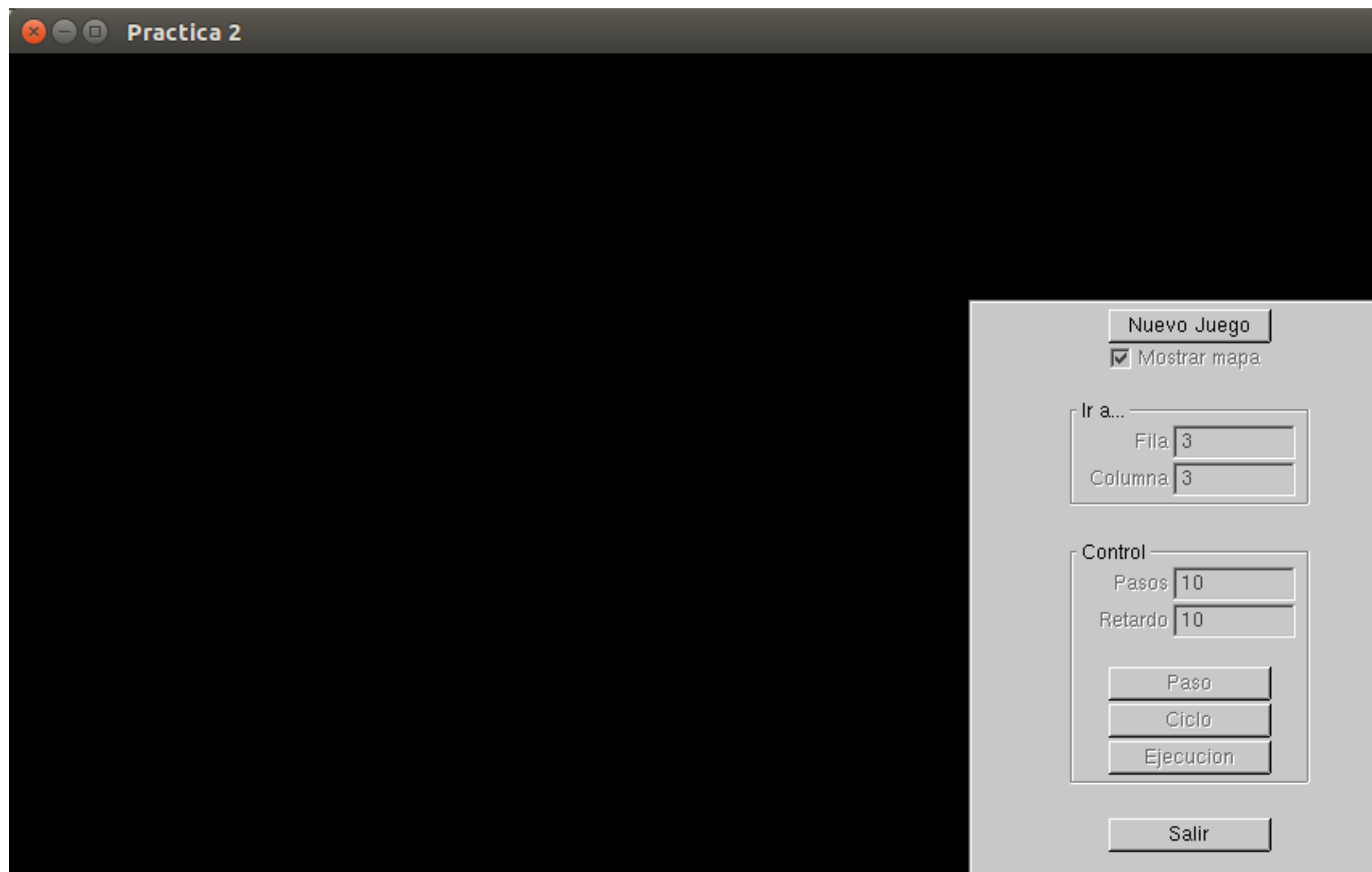


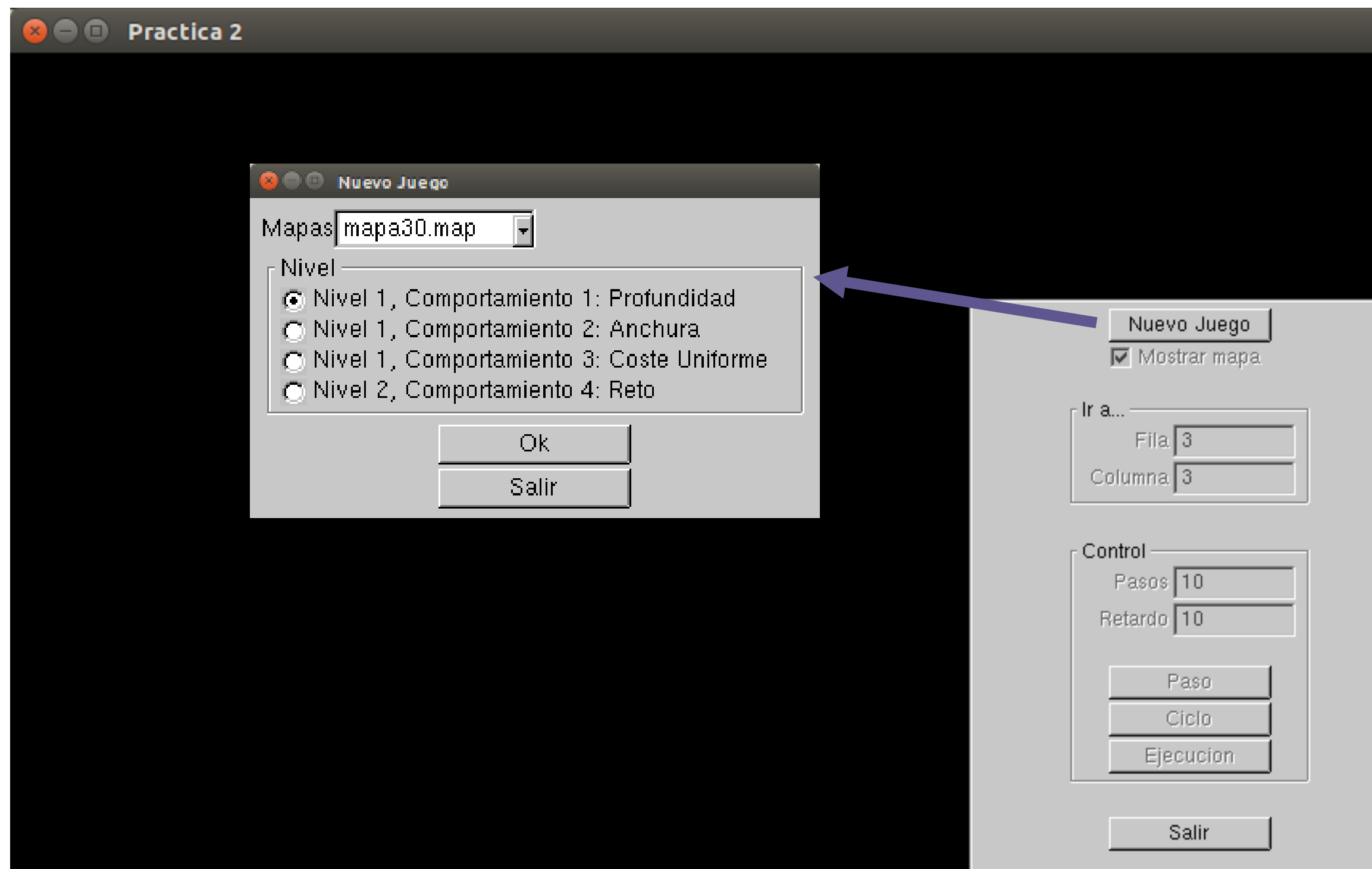
- Introducción
- Presentación del problema
- Los extraños mundos de Belkan
- Objetivo
- **Software**
- Método de evaluación y entrega de prácticas

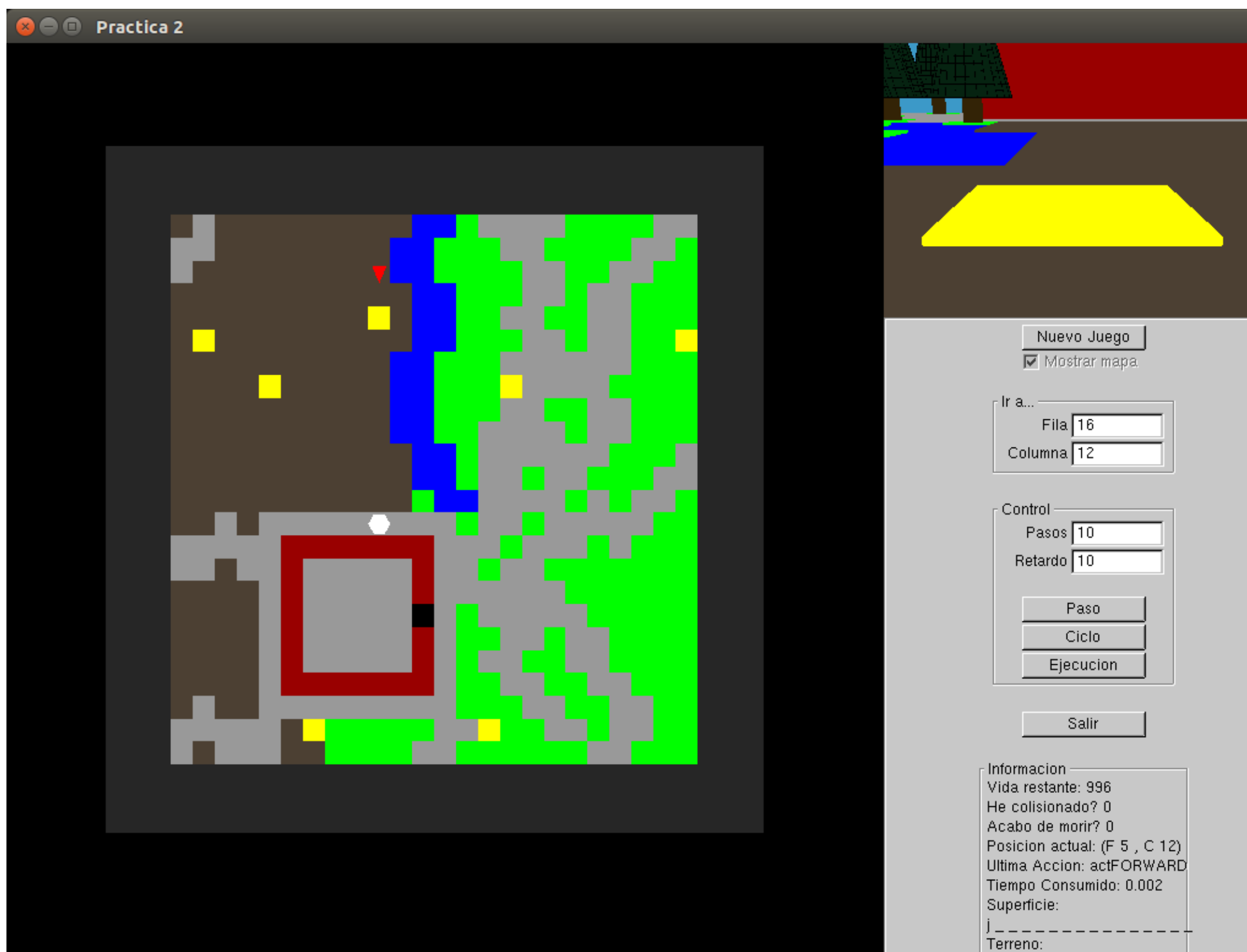


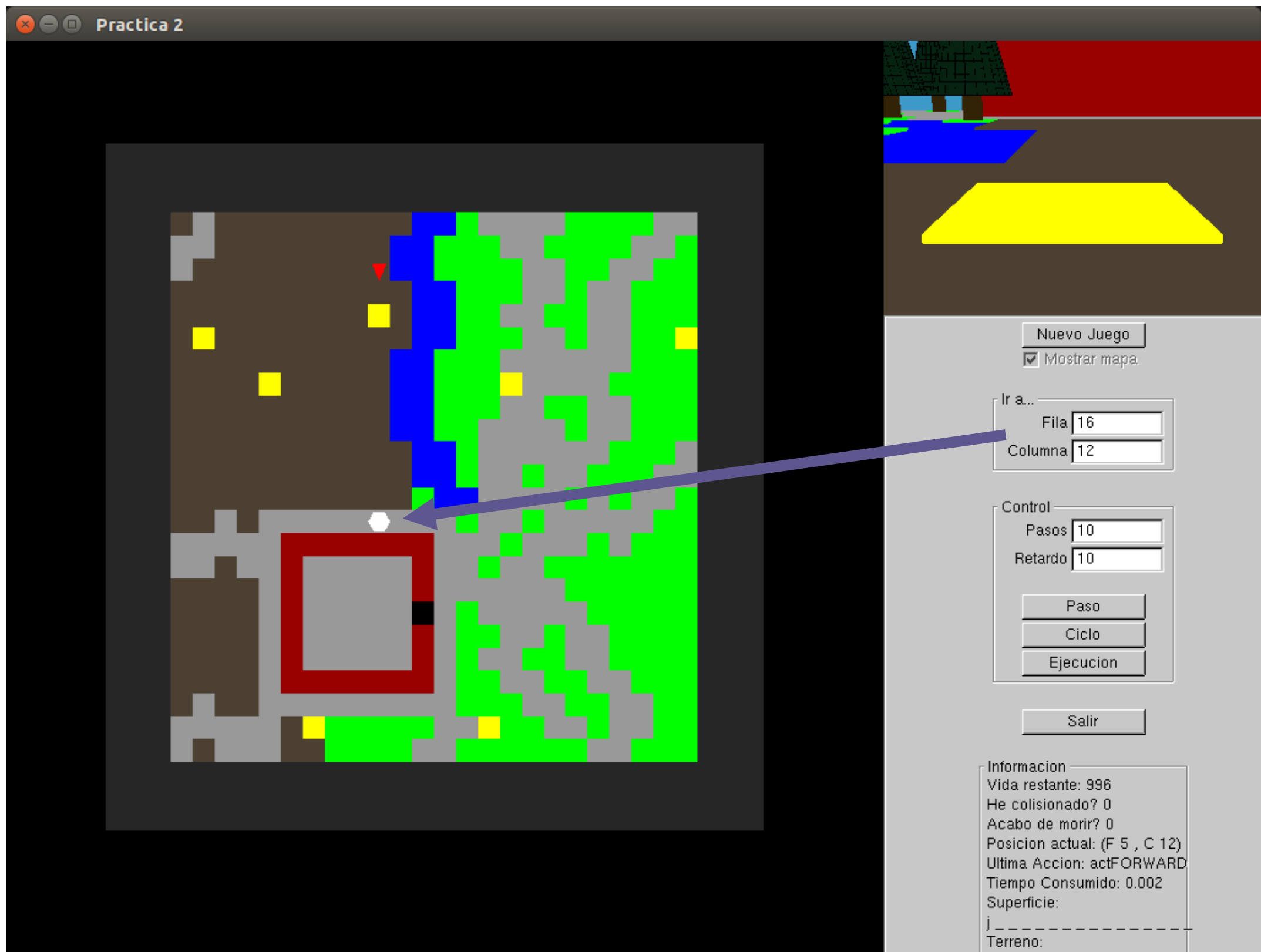
- Se proporciona el software para el sistema operativo Linux (Distribución Ubuntu). Dicho software se puede encontrar en el apartado de “Material de la Asignatura” dentro de la plataforma docente de DECSAI.
- Tras compilar el software, podemos encontrar dos ficheros ejecutables: **Belkan** y **BelkanSG**. El primero corresponde al simulador con interfaz gráfica, mientras que el segundo es un simulador en modo *batch* sin interfaz.
- Todos los detalles y explicación de la instalación, uso y detalles las variables necesarias para su desarrollo se encuentra en el guion de prácticas asociado a esta presentación.

## ./Belkan









- Sistema *batch*:

`./BelkanSG mapas/mapas100.map 1 3 94 64 15 15 7 7`

- fichero de mapa
- semilla para inicializar el generador de números aleatorios
- comportamiento (1, 2, 3 o 4)
- fila origen
- columna origen
- pares de (fila, columna) destino

En caso de quedarse sin destinos, los elegirá al azar.



- Al finalizar la ejecución nos el tiempo consumido, la cantidad de fallos que hemos cometido al descubrir el mapa (para el nivel 2), el número de colisiones que hemos sufrido (por obstáculos u otros agentes), la cantidad de muertes por caídas a precipicios y la cantidad de destinos alcanzados.

```
Tiempo consumido: 3.25805
Fallos: 0
Colisiones: 0
Muertes innecesarias: 0
Objetivos encontrados: 15
```

- Introducción
- Presentación del problema
- Los extraños mundos de Belkan
- Objetivo
- Software
- **Método de evaluación y entrega de prácticas**

- Se pide desarrollar un programa (modificando el código de los ficheros del simulador ‘jugador.cpp’ y ‘jugador.hpp’) con el comportamiento requerido para el agente.
- Estos dos ficheros deberán entregarse mediante la plataforma web de la asignatura, en un fichero ZIP (de nombre “practica2.zip” que no contenga carpetas.
- El archivo ZIP deberá contener sólo el código fuente de estos dos ficheros con la solución del alumno así como un fichero de documentación en formato PDF que describa el comportamiento implementado con un máximo de 5 páginas.

- A partir del comportamiento entregado por el alumno en los ficheros antes mencionados y del nivel elegido por el mismo, se realizarán varias simulaciones con diferentes mapas (de diferente tamaño, pero con un máximo de 100x100) y con diferentes destinos. El simulador no debe “colgarse” en ningún momento y debe, por supuesto, compilar sin problemas.
- Se valorará la cantidad de destinos alcanzados en cada simulación (en el nivel 1 solo se puede alcanzar un destino, en el nivel 2 no hay límite, salvo el dado por la cantidad de pasos y el tiempo máximo) y restará puntos cualquier colisión o muerte innecesaria.

- La nota final se calculará de la siguiente manera:

$$\text{Nota final} = \text{Nota Practica} * \text{Defensa}$$

donde “Defensa” es un valor en  $[0,1]$  indicando en que medida el alumno ha sabido defender los comportamientos implementados en su agente en el proceso de defensa.

- Si se opta por el Nivel 1, la nota máxima alcanzable será de 5.
- Si se opta por el Nivel 2, debe haber superado el nivel 1 y la nota máxima alcanzable será de 10.

Grupo de prácticas	Fecha límite entrega	Fecha defensa
<b>TODOS</b>	28 de Abril hasta las 23:00 horas	Semana del 6 de Mayo

## Esta práctica es **INDIVIDUAL**

- En el caso de detectar prácticas copiadas, los involucrados (tanto el que se copió como el que se ha dejado copiar) tendrán suspensa la asignatura.