



* * IA EN LOS VIDEOJUEGOS: NPCs INTELIGENTES Y ÁRBOLES DE DECISIÓN

Cesar Villegas

Valentina Rosas

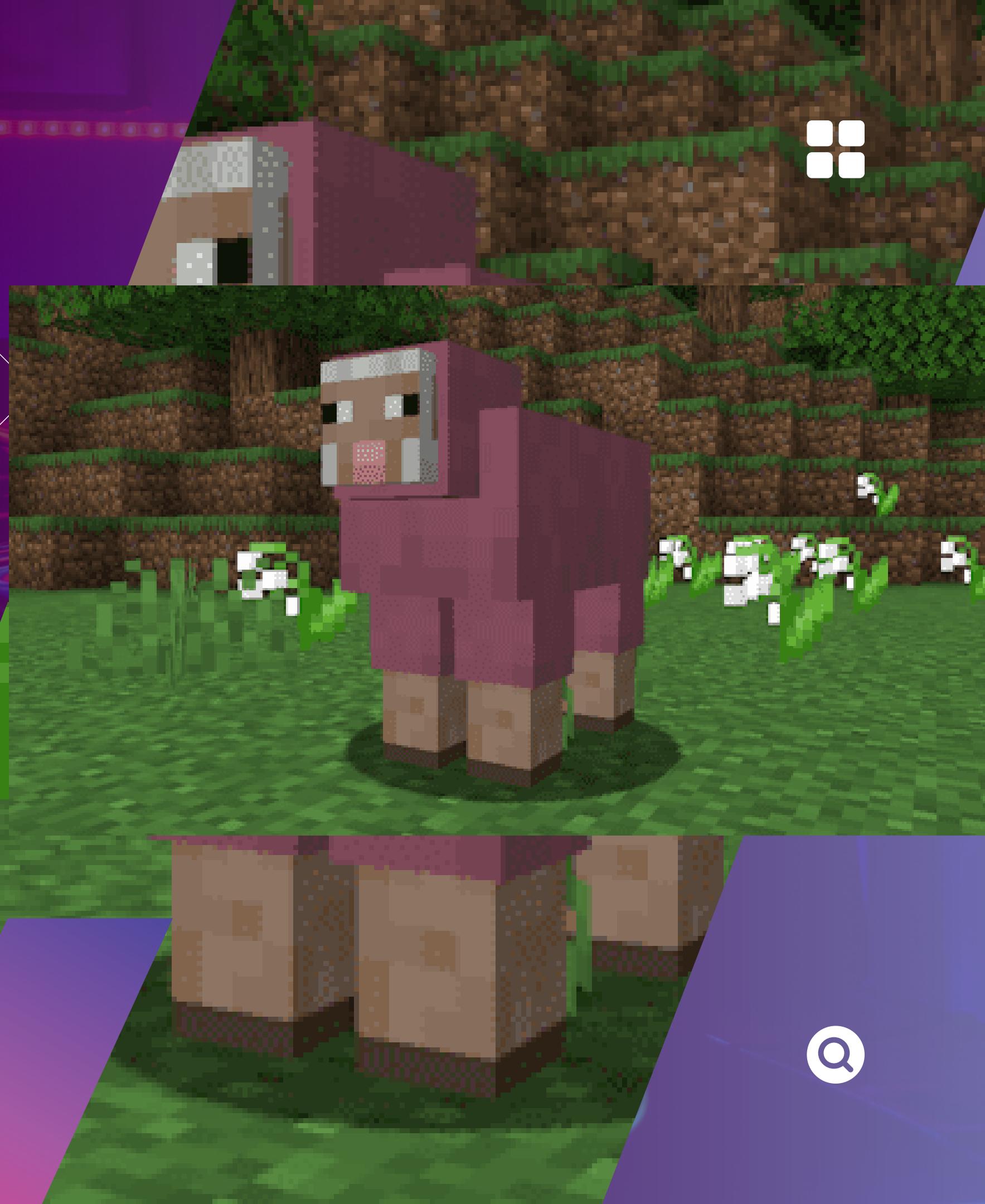
Empezar →





¿QUÉ ES UN NPC?

- NPC: Non-Player Character o personaje no jugable.
- Presente en casi todos los videojuegos: aliados, enemigos, comerciantes, civiles.
- La IA permite que tengan comportamientos coherentes y adaptativos.



¿QUÉ ES UN NPC Y QUÉ LO HACE "INTELIGENTE"?

NPC NORMAL



NPC INTELIGENTE



- No reacciona a lo que hace el jugador.
- Solo sigue scripts simples

- Tiene IA incorporada para tomar decisiones.
- Puede reaccionar a lo que hace el jugador
- Su comportamiento puede ser dinámico o incluso aprender del entorno.

*JA EN LOS NPCS: ALGORÍTMOS COMUNES

1. Máquinas de Estados

Permiten que un NPC cambie entre comportamientos como patrullar, atacar o huir.

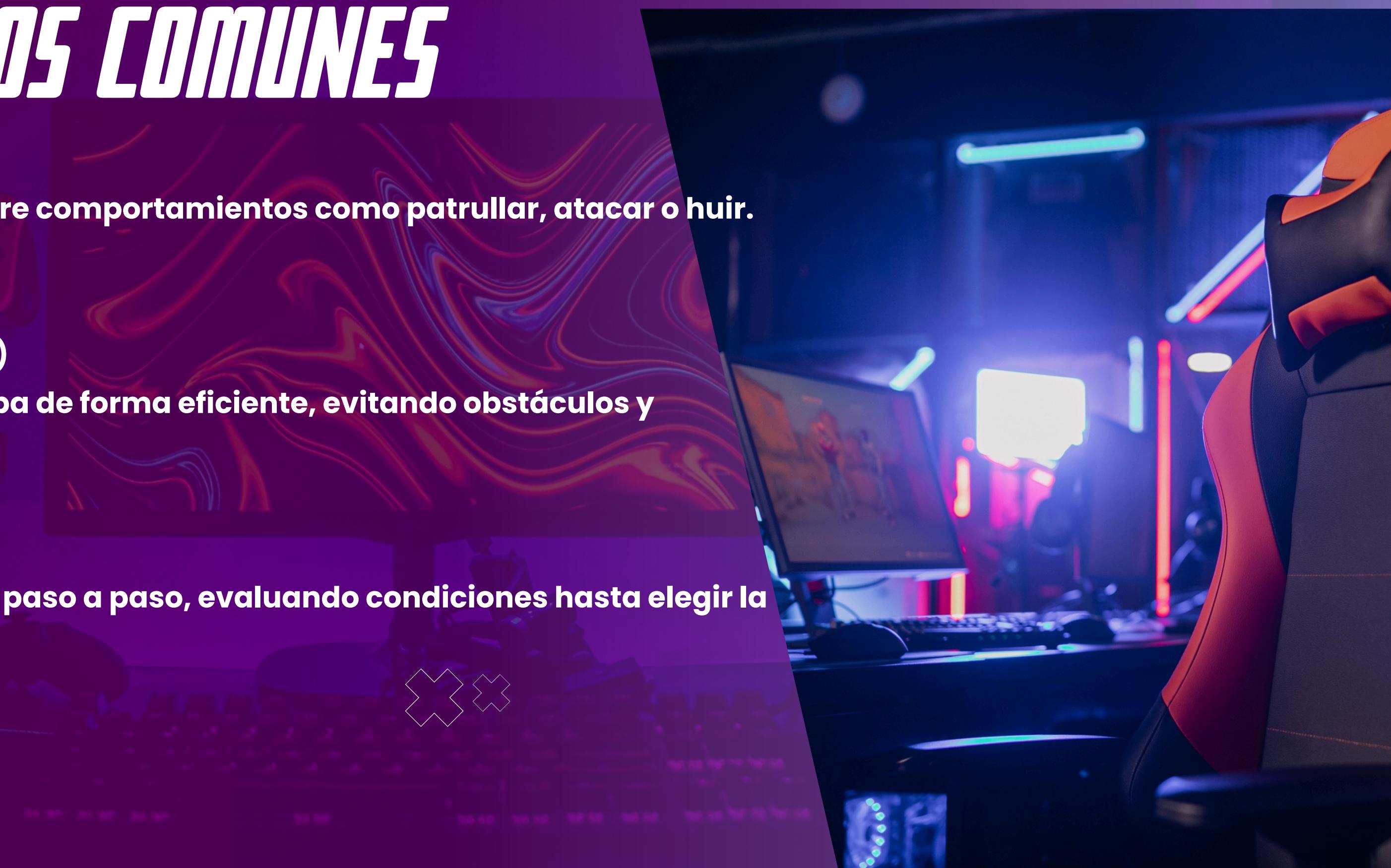
Son simples, pero muy usadas.

2. Algoritmos de Navegación (A*)

Les ayudan a moverse por el mapa de forma eficiente, evitando obstáculos y encontrando rutas.

3. Árboles de Decisión

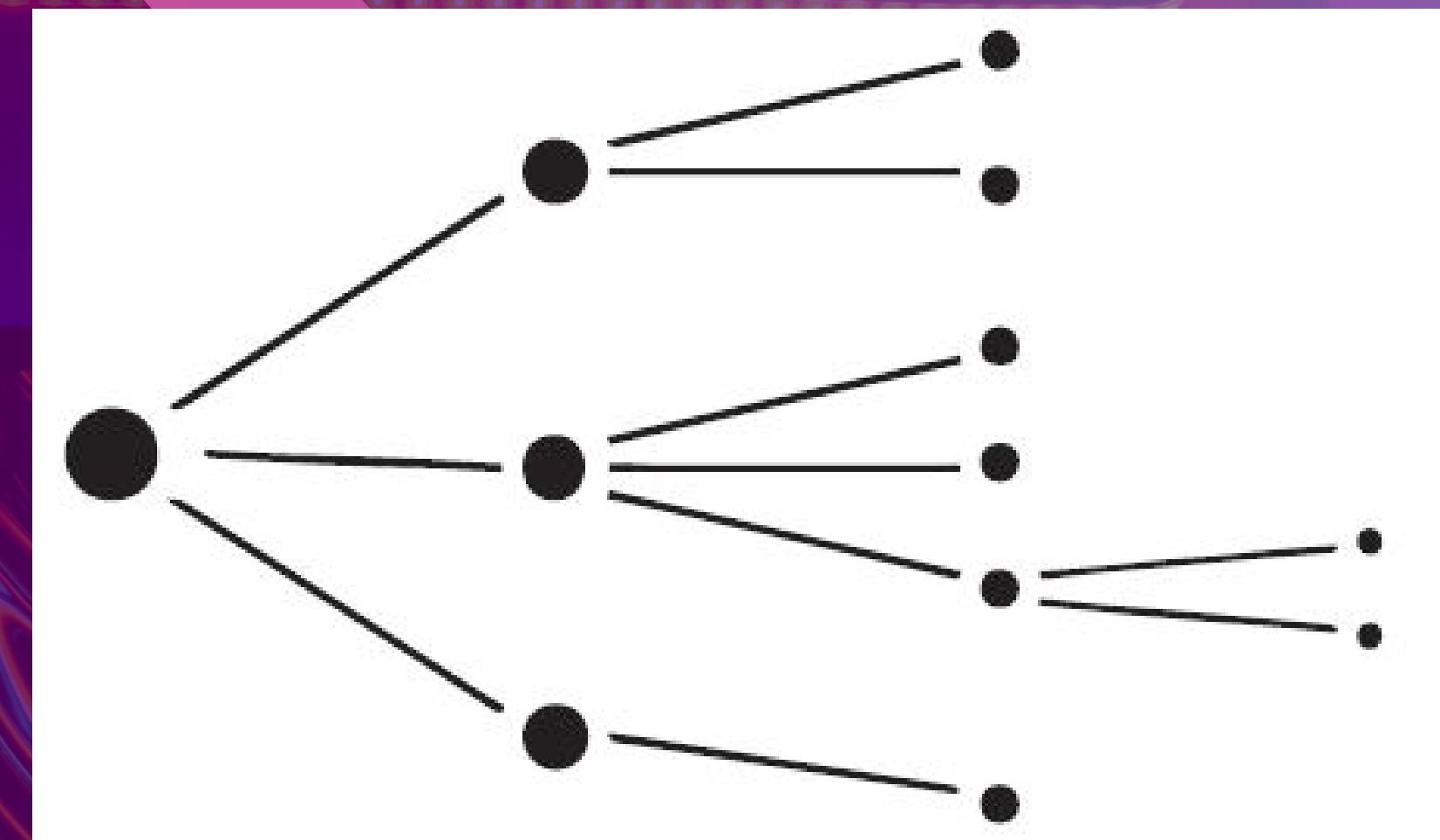
Organizan decisiones complejas paso a paso, evaluando condiciones hasta elegir la mejor acción.





¿QUÉ ES UN ÁRBOL DE DECISIÓN?

- Cada nodo representa una pregunta o condición.
- Cada rama representa una respuesta posible (por ejemplo, "sí" o "no").
- Las hojas al final del árbol indican la acción que el NPC va a realizar.





EJEMPLO EN UN VIDEOJUEGO:

Un guardia debe decidir entre atacar, patrullar o huir:

¿Jugador está cerca?

— Sí → **¿Jugador tiene arma?**

— | — Sí → **Huir**

— | — No → **Atacar**

— — No → **Patrullar**



Search Presentation



Read More

CÓDIGO EN PYTHON

```
1  def decidir_accion(jugador_cerca, jugador_armado):
2      if jugador_cerca:
3          if jugador_armado:
4              return "Huir"
5          else:
6              return "Atacar"
7      else:
8          return "Patrullar"
9
10 # Bloque principal que se ejecuta al correr el script
11 if __name__ == "__main__":
12     casos = [
13         (False, False),
14         (False, True),
15         (True, False),
16         (True, True)
17     ]
18
19     for cerca, armado in casos:
20         accion = decidir_accion(cerca, armado)
21         print(f"Jugador cerca: {cerca}, Armado: {armado} → Acción: {accion}")
22
```

Search Presentation

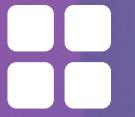


Read More

TABLA DE VERDAD

A (Cerca)	B (Armado)	Acción
0	0	Patrullar
0	1	Patrullar
1	0	Atacar
1	1	Huir

[Read More](#)



BOOLEANAS

ESPORT GAME



Usamos funciones para cada acción:

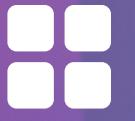
- $f_{\text{Atacar}}(A, B) = A \wedge \neg B$
- $f_{\text{Huir}}(A, B) = A \wedge B$
- $f_{\text{Patrullar}}(A, B) = \neg A$

Search Presentation

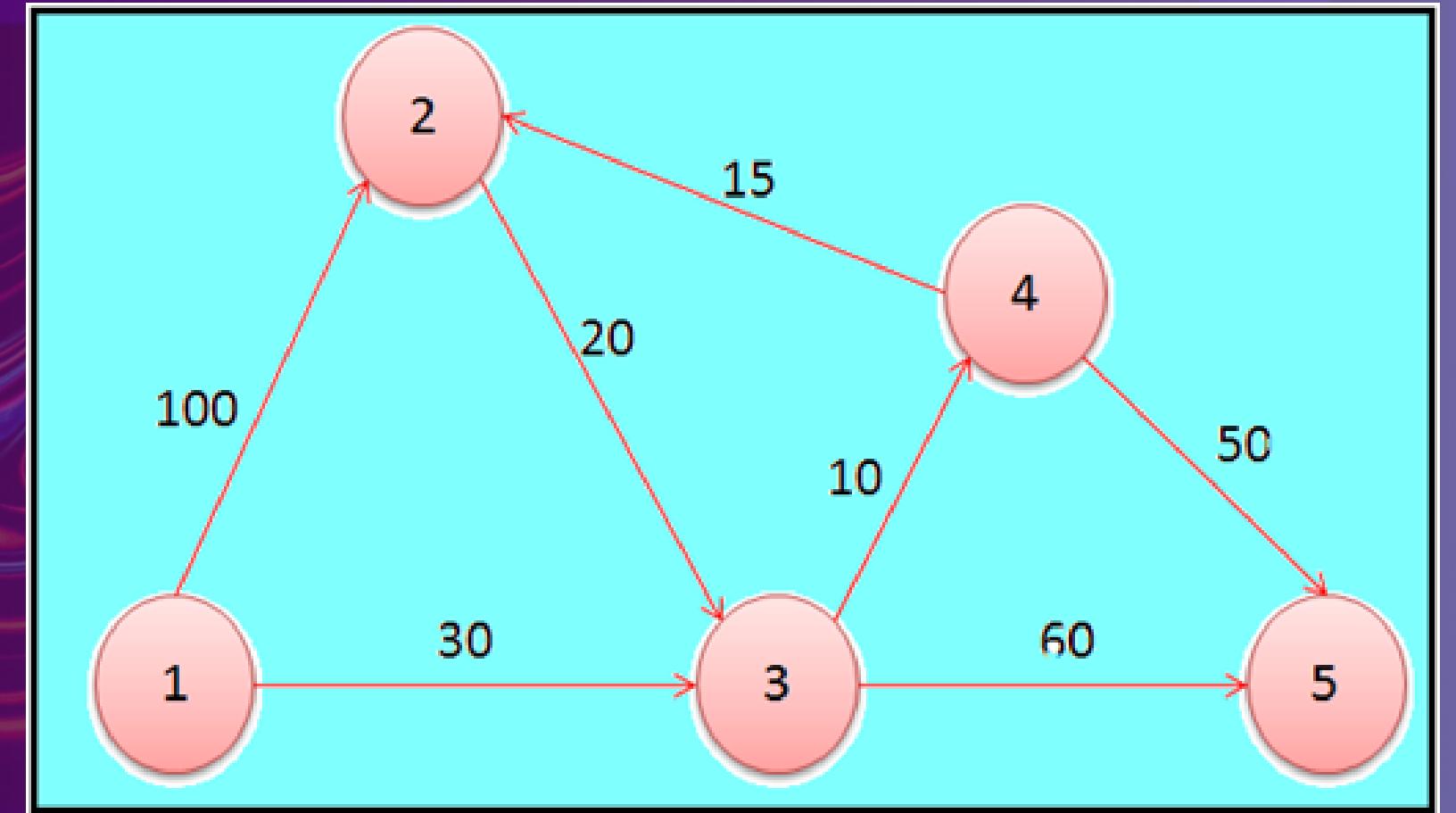


Read More

ALGORITMO A* (A ESTRELLA)



El algoritmo A* (A estrella) se usa para encontrar la ruta más corta y eficiente entre dos puntos en un mapa. Lo utilizan NPCs para moverse de forma inteligente, evitando obstáculos y eligiendo el mejor camino posible.



Search Presentation



Read More



¿ CÓMO FUNCIONA?



A* se basa en la siguiente función de costo:

$$f(n) = g(n) + h(n)$$

Donde:

$g(n)$: es el costo real desde el punto de inicio hasta el nodo n .

$h(n)$: es una heurística (una estimación del costo restante desde n hasta el destino).

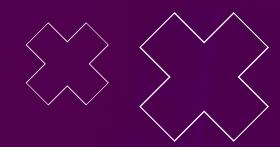
$f(n)$: es el costo total estimado pasando por ese nodo



Search Presentation



Read More



MUCHAS
GRACIAS

