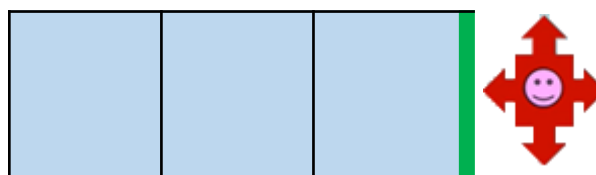
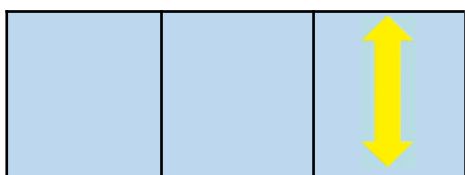
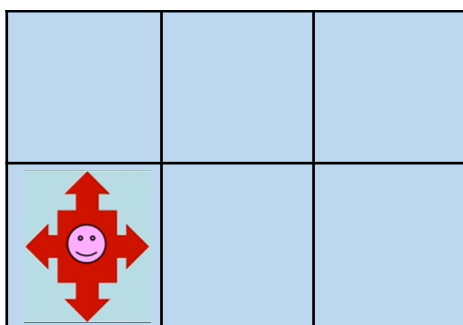


### **Guía de Actividades N° 5: Problema del Escape para un Agente Inteligente utilizando Reglas de Producción**

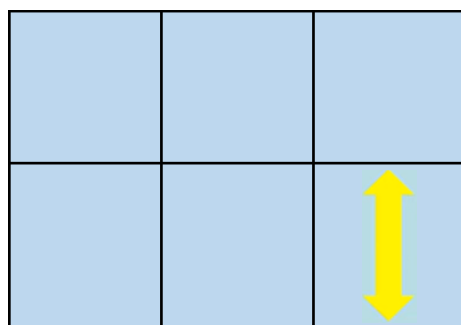
|                           |  |
|---------------------------|--|
| Contenidos                | Eje temático 1 y 2: Agentes inteligentes - Resolución de problemas de Búsqueda y Planificación.  |
| Objetivos                 | <ul style="list-style-type: none"> <li>• Representar los elementos del problema (objetivo, estado inicial, estados intermedios, estado final) en un papel y en un lenguaje de programación.</li> <li>• Resolver el problema del escape utilizando reglas de producción.</li> <li>• Identificar y definir los operadores válidos para el problema del escape</li> </ul> |
| Modalidad de desarrollo   | Individual o parejas   |
| Teoría de aprendizaje     | Aprender haciendo  |
| Tipo de actividad         | Análisis reflexivo; desarrollo práctico  |
| Recursos tecnológicos     | Lenguaje de programación a elección del estudiante   |
| Recursos materiales       | <ul style="list-style-type: none"> <li>• Bibliografía base y complementaria, apuntes de clase, Internet, IA generativas.</li> <li>• Papel y lápiz para representación del problema, recursos de programación, acceso a internet</li> </ul>   |
| Instrumento de evaluación | Presentación oral y escrita del proceso de resolución y del código desarrollado  |
| Criterios de evaluación:  | <ul style="list-style-type: none"> <li>• Correcta representación de los elementos del problema.</li> <li>• Identificación adecuada de los operadores válidos.</li> <li>• Calidad del código desarrollado.</li> <li>• Habilidad para resolver problemas y comunicar la solución.</li> <li>• Responsabilidad y cumplimiento en tiempo y forma.</li> </ul>                |

Desarrollar un agente inteligente capaz de resolver el problema del escape en un tablero de juego similar al que se muestra en la imagen, utilizando técnicas de búsqueda y resolución de problemas en la IA. El desafío del agente es: Utilizando tu inteligencia y habilidades de resolución de problemas, ¿podrás encontrar la estrategia óptima para mover la ficha roja objetivo hasta la casilla de salida (verde) y escapar del tablero de juego antes de que se agote el tiempo?, considerando las reglas del juego y las restricciones establecidas.





**Estado inicial**



**Estado Meta**

Reglas del problema:

- El tablero está formado por una cuadrícula de 3 x 3 casillas
- En el tablero hay 2 fichas, cada una con un color diferente: una ficha roja (objetivo) que debes mover hasta la casilla de salida (marcada borde verde) para escapar y una ficha amarilla (bloqueo) que sirve de obstáculo dependiendo su posición
- Las fichas solo pueden moverse a lo largo de las líneas horizontales y verticales del tablero a casillas vacías.
- Una ficha no puede saltar sobre otra ficha que se encuentre en su camino, es decir puede moverse a una casilla que esté vacía.
- No puedes mover fichas en diagonal ni en sentido contrario.
- Debes tener cuidado de no quedar bloqueado por la ficha amarilla, ya que esto te impediría escapar.
- Tienes un número limitado de movimientos para completar el desafío (salir por la puerta verde)

### **Consignas:**

Diseñar un agente inteligente capaz de percibir el entorno, razonar, actuar y aprender. Considerar cómo el agente puede planificar su escape en función de los elementos de la búsqueda objetivo, las restricciones y la información percibida. Integrar las capacidades del agente inteligente en la implementación del problema y asegurarse de que el agente pueda tomar decisiones efectivas para un escape eficiente

#### **1. Análisis del Problema:**

- a. Leer detenidamente el problema del escape

- b. Identificar los elementos del problema (estado inicial, estado intermedio, el estado final, y restricciones).
- c. Definir los operadores válidos para el problema, teniendo en cuenta las restricciones proporcionadas.

**2. Representación del Problema:**

- a. Realizar una representación esquemática del problema (tablero, posiciones de las fichas, movimientos posibles)
- b. Reflexionar sobre cómo modelar este problema en un lenguaje de programación

**3. Modelar el proceso** de resolución del problema (estructural y funcionalmente), incluyendo la representación de los elementos, la definición de operadores y el código desarrollado.

- a. Completar Los descriptores que permitirán modelar el comportamiento del agente:
  - i. Tabla PAMA (Percepción – Acción – Medios fines/meta – Ambiente)
  - ii. Tabla REAS (Rendimiento – Entorno – Actuadores – Sensores)
  - iii. Tabla P→A (Percepción → Acción)
- b. A partir del análisis anterior, definir un conjunto de reglas de producción para el agente (convirtiendo la información tácita en explícita). Las reglas deben tener la forma:

**SI (condición), ENTONCES (acción)**

**4. Implementación.**

- a. Codificar el problema del escape en el lenguaje de programación elegido, utilizando las reglas de producción definidas.
- b. Incluir en el código las reglas de producción definidas previamente.
- c. Ejecutar y probar el programa en al menos 6 escenarios distintos (juegos).
- d.

**5. Evaluación y mejora del Agente Inteligente:**

- a. Analizar si el agente puede percibir, razonar, actuar y aprender en función del entorno.
- b. Evaluar la eficiencia de la solución y proponer mejoras.
- c. Entregar el código comentado y una breve presentación escrita u oral del proceso de resolución.
- d.

**6. Entrega y Evaluación:**

- a. Subir al aula la documentación del proceso realizado (análisis, modelado, implementación, validación, mejora, reflexiones).

# Análisis del problema

## Elementos del problema

| Elemento          | Ficha roja  |
|-------------------|---|
| Tipo de Agente    | <p><b>Agente basado en objetivos</b></p> <p>Según la teoría (Russell &amp; Norvig, y los documentos que usás en la carrera) un agente basado en objetivos es aquel que:</p> <p><i>“Toma decisiones y ejecuta acciones en función de una meta que desea alcanzar, evaluando si las acciones lo acercan o alejan del cumplimiento de ese objetivo.”</i></p> <p>Este tipo de agente:</p> <ul style="list-style-type: none"> <li>• Tiene una meta definida.</li> <li>• Selecciona sus acciones en función de esa meta.</li> <li>• No actúa al azar, ni por simples reacciones sin rumbo.</li> </ul> |
| Estado inicial    | <p><b>Posición de la ficha Roja: (2, 0), (FILA, COLUMNA)</b><br/><b>(Esquina inferior izquierda).</b></p> <p><b>Posición de la ficha amarilla: (0, 2) (Esquina superior derecha).</b></p>   |
| Estado intermedio | <p>Posiciones intermedias durante el recorrido en la grilla.</p> <p>Cualquier configuración del tablero en la que la ficha roja se haya movido al menos una vez desde su posición inicial y aún no haya alcanzado la casilla de salida</p> <p>(1,0) – Un paso hacia arriba desde el inicio</p> <p>(2,1) – Un paso a la derecha desde el inicio</p> <p>(0,0) – Dos pasos hacia arriba desde el inicio</p>  |

|                        |  |
|------------------------|--|
|                        | <p>(1,1) – Está en el medio</p> <p>(0,1) – Paso previo a la meta ( si lo ves desde la perspectiva que está a la izquierda de la meta )</p> <p>(1,2) - Paso previo a la meta ( si lo ves desde la perspectiva que está abajo de la meta )</p> <p>(2,2) - Está en la esquina inferior derecha</p>  |
| Estado final<br>(meta) | (0,2) - La ficha roja llegó a la meta  |
| Restricciones          | <p>Movimiento</p> <ol style="list-style-type: none"> <li>1. Solo se puede mover una casilla por turno</li> <li>2. Solo puede desplazarse a casillas adyacentes en <b>vertical u horizontal</b> (no en diagonal).</li> <li>3. <b>No puede saltar casillas.</b> (No se puede pasar por encima de otra ficha ni atravesar casillas bloqueadas)</li> <li>4. No puede salirse del tablero ((0 a 2, 0 a 2))</li> <li>5. La casilla ocupada por la ficha amarilla se considera <b>bloqueada temporalmente</b>. (El rojo debe evitar ese lugar o esperar que se libere.)</li> <li>6. No puede moverse dos veces seguidas sin que actúe el otro agente</li> <li>7. No puede moverse si todas las casillas adyacentes están ocupadas o fuera del tablero (Si el agente está rodeado de bordes del tablero y/o del otro agente, puede quedar <b>temporalmente sin movimientos posibles</b>.)</li> <li>8. Solo toma decisiones basadas en su entorno inmediato (No puede adivinar o saber de antemano que va a hacer la amarilla)</li> </ol> |
| Entorno del ejercicio  | <p>Tablero 3x3 con una ficha roja (objetivo) y una ficha amarilla (bloqueo) colocadas en posiciones definidas. ( Nuestro caso)</p> <p>El entorno es <b>discreto, determinista, estático y parcialmente observable</b>:</p>   |

|  |  |
|--|--|
|  | <p><b>Discreto:</b> El tablero está compuesto por un número finito de casillas (9 en total).</p> <p><b>Determinista:</b> Cada acción tiene un único resultado conocido (por ejemplo, moverse una casilla hacia abajo siempre termina en la casilla inferior).</p> <p><b>Estático:</b> El entorno no cambia mientras el agente está tomando una decisión.</p> <p><b>Parcialmente observable:</b> El agente amarillo percibe solamente la posición actual del agente rojo en el tablero, pero no puede predecir sus movimientos futuros.</p> |
|--|--|

| Elemento       | FICHA AMARILLA  |
|----------------|---|
| Tipo de Agente | <p><b>Agente basado en objetivos</b></p> <p>Según la teoría (Russell &amp; Norvig, y los documentos que usás en la carrera) un agente basado en objetivos es aquel que:</p> <p><i>“Toma decisiones y ejecuta acciones en función de una meta que desea alcanzar, evaluando si las acciones lo acercan o alejan del cumplimiento de ese objetivo.”</i></p> <p>Este tipo de agente:</p> <ul style="list-style-type: none"> <li>• Tiene una meta definida.</li> <li>• Selecciona sus acciones en función de esa meta.</li> <li>• No actúa al azar, ni por simples reacciones sin rumbo.</li> </ul> |
| Estado inicial | <p>Posición de la ficha <b>Amarilla</b>: (0, 2), (FILA, COLUMNA)<br/>(Esquina superior derecha).</p> <p>Posición de la ficha <b>Roja</b>: (2, 0) (Esquina inferior izquierda).</p>  |

|                     |   |
|---------------------|---|
| Estado intermedio   | <p>Posiciones intermedias durante el recorrido en la columna 2 (la única por la que se puede mover).</p> <p>Cualquier configuración del tablero en la que la ficha amarilla se haya desplazado verticalmente y aún no se haya definido el resultado del juego.</p> <p>(1,2) - Baja desde la posición inicial para anticiparse a una entrada del rojo desde abajo ( Un paso hacia abajo desde su estado inicial)</p> <p>(2,2) - Baja hasta la base de la columna para evitar que el rojo ingrese desde la derecha inferior</p> <p>(0,2) - Vuelve al estado inicial (desde su primer momento)</p> |
| Estado final (meta) | <p>(0,2) - La amarilla ocupa directamente la casilla de meta y el rojo no puede acceder</p> <p>(1,2) - El rojo intenta entrar desde abajo y la amarilla lo bloquea desde arriba</p> <p>(2,2) - El rojo intentó entrar por la esquina inferior derecha, y se ve bloqueado desde ahí</p> <p>A diferencia del agente rojo, el amarillo <b>no tiene un único estado final fijo</b>, sino <b>varios posibles</b>, que dependen de la posición del rojo.</p>  |
| Restricciones       | <p>Movimiento</p> <ol style="list-style-type: none"> <li>1. Solo se puede mover <b>una casilla por turno</b>.</li> <li>2. Solo puede desplazarse a <b>casillas adyacentes en dirección vertical (<math>\uparrow</math> o <math>\downarrow</math>)</b> dentro de la <b>columna 2</b> (tercera columna del tablero).</li> <li>3. <b>No puede moverse en dirección horizontal ni diagonal.</b></li> <li>4. No puede salirse del tablero (</li> <li>5. No puede saltar casillas (El agente <b>no puede desplazarse más de una casilla por turno</b>, No puede “saltar” por encima del</li> </ol>    |

|                       |   |
|-----------------------|---|
|                       | <p>agente rojo, Cada movimiento debe ser <b>a una casilla libre y adyacente verticalmente.</b>)</p> <p>6. No puede moverse dos veces seguidas sin que actúe el otro agente</p> <p>7. No puede bloquear todos los caminos al mismo tiempo</p> <p>8. Solo toma decisiones basadas en su entorno inmediato</p>   |
| Entorno del ejercicio | <p>Tablero 3x3 con una ficha roja (objetivo) y una ficha amarilla (bloqueo) colocadas en posiciones definidas. ( Nuestro caso)</p> <p>El entorno es <b>discreto, determinista, estático y parcialmente observable:</b></p> <p><b>Discreto:</b> El tablero está compuesto por un número finito de casillas (9 en total).</p> <p><b>Determinista:</b> Cada acción tiene un único resultado conocido (por ejemplo, moverse una casilla hacia abajo siempre termina en la casilla inferior).</p> <p><b>Estático:</b> El entorno no cambia mientras el agente está tomando una decisión.</p> <p><b>Parcialmente observable:</b> El agente amarillo percibe solamente la posición actual del agente rojo en el tablero, pero no puede predecir sus movimientos futuros.</p> |

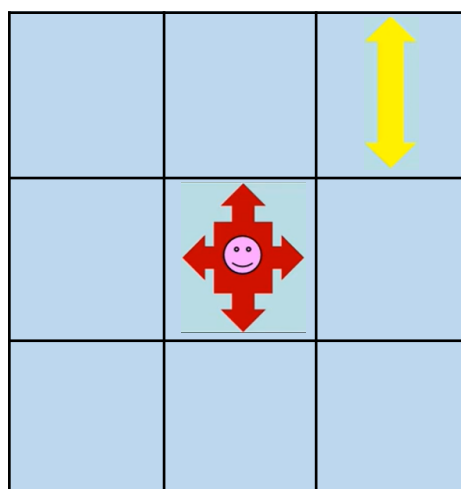
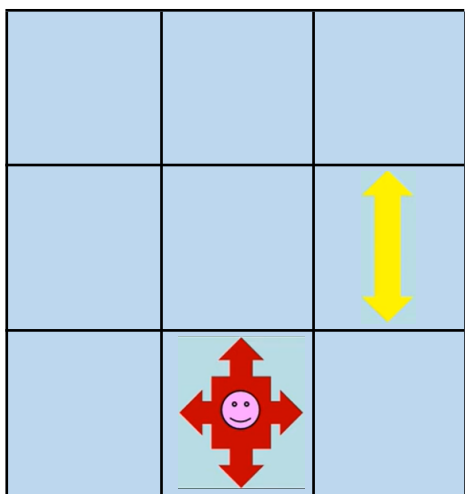
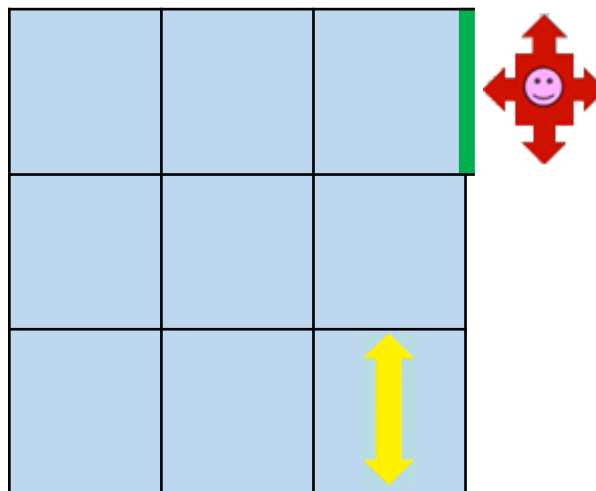
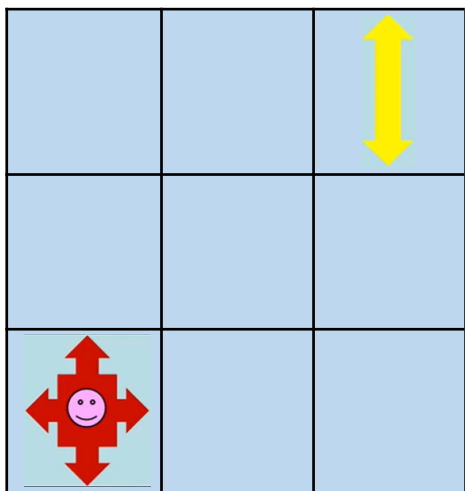
## Operadores válidos para el problema

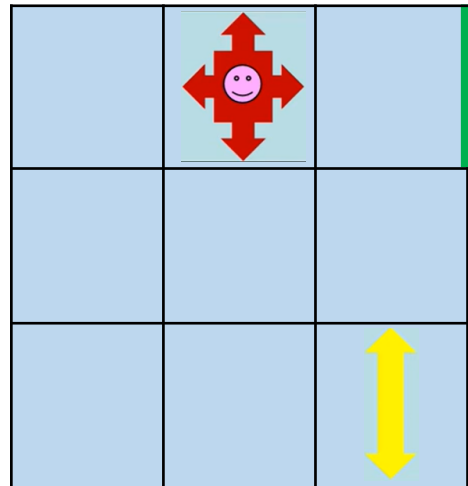
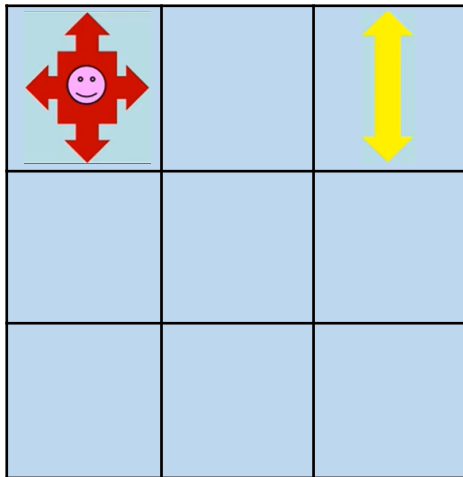
| Agente rojo  | Agente amarillo  |
|--|--|
| <ul style="list-style-type: none"> <li>Moverse a la derecha siempre y cuando la casilla esté disponible</li> </ul> | <ul style="list-style-type: none"> <li>Moverse a arriba siempre y cuando la casilla esté disponible</li> </ul> |



|   |   |
|---|---|
| <ul style="list-style-type: none"><li>● Moverse a la izquierda siempre y cuando la casilla esté disponible</li><li>● Moverse a arriba siempre y cuando la casilla esté disponible</li><li>● Moverse a abajo siempre y cuando la casilla esté disponible</li></ul> | <ul style="list-style-type: none"><li>● Moverse a abajo siempre y cuando la casilla esté disponible</li></ul> |
|---|---|

## 2. Posibles escenarios





## Reflexión acerca de cómo programar

- Preguntarse:
  - ¿Qué entidades hay y cómo se representan?
- ¿Cuál es el objetivo de cada agente?
- ¿Cómo luce un estado del juego?
- ¿Qué reglas hacen que un estado pase a otro?
- ¿Cómo es el turno a turno?
- ¿Qué condiciones marcan un movimiento válido?
- ¿Qué tipo de inteligencia va a tener cada agente?
- ¿Cuándo termina el juego?

## Modelar

### Identificar los elementos del sistema

#### Entidades clave:

- Un **tablero** (3x3) → espacio discreto
- Un **agente rojo** (jugador) → intenta alcanzar una meta
- Un **agente amarillo** (bloqueador) → intenta impedirlo
- Una **meta fija**:  $(0, 2)$  es el objetivo del rojo

#### Acciones posibles:

- Mover arriba, abajo, izquierda, derecha (según reglas)
- Turnos alternos entre los agentes

### Definir los objetivos

| Agente   | Objetivo principal                    |
|----------|---------------------------------------|
| Rojo     | Llegar a $(0, 2)$                     |
| Amarillo | Impedir que el rojo llegue a $(0, 2)$ |

Ambos son **agentes racionales**, cada uno con metas claras y comportamientos que deben orientarse a cumplirlas.

## Pensar en los estados

Un **estado** representa una configuración del juego:

- Posición del rojo
- Posición del amarillo

Ejemplo de un estado:

`(pos_rojo=(2,0), pos_amarillo=(0,2))`

La programación se puede pensar como una **exploración de posibles estados**, donde en cada turno se genera un nuevo estado dependiendo de las acciones.

## Pensar en las reglas de transición entre estados

Te preguntás:

¿Qué puede pasar a continuación desde el estado actual?

- ¿El rojo puede moverse?
  - ¿A dónde?
  - ¿La casilla está libre?
  - ¿Sigue dentro del tablero?
- ¿El amarillo puede bloquearlo?
  - ¿Debe quedarse en la meta o bajar?
  - ¿Anticipa o reacciona?

Así definís las **transiciones de estado**, que más adelante vas a codificar como funciones o estructuras condicionales.

## Pensar en el flujo del juego (turnos)

Mientras el juego no termine:

- si turno es del rojo:
  - calcular su próximo movimiento
- si turno es del amarillo:
  - decidir cómo bloquear
- verificar si alguien ganó o perdió:
  - cambiar turno

## Pensar en las restricciones

Esto es muy importante antes de programar:

- El rojo no puede salir del tablero.
- No puede ocupar la misma casilla que el amarillo.
- El amarillo solo se mueve en vertical dentro de la columna 2.
- Se mueven en turnos alternados.

Estas restricciones se traducen más tarde en condicionales y validaciones.

## Elegir el tipo de inteligencia o lógica

Tenés que decidir:

- ¿Querés que los agentes usen reglas fijas? (como un **SI . . . ENTONCES**)
- ¿Querés que el rojo busque siempre el camino más corto?
- ¿Querés que el amarillo solo reaccione o que también anticipe?

### Pensar en condiciones de finalización

Antes de arrancar a programar, tenés que saber cuándo termina el juego:

- El rojo llega a  $(0, 2) \rightarrow$  gana
- El amarillo bloquea la meta y el rojo no puede avanzar más  $\rightarrow$  gana el amarillo
- Opcional: límite de turnos, empate, etc.

3.

PAMA

| Ficha    | Percepción  | Acción  | Medios/Fines             | Ambiente  |
|----------|---|---|--------------------------|---|
| Roja     | Posición propia, posición actual y futura de ficha amarilla | Mover a celda segura y disponible en las cuatro direcciones considerando la más cercana a la meta | Escapar sin colisionar   | Tablero cuadrado, ficha amarilla como obstáculo móvil |
| Amarilla | Posición propia, posición actual y futura de ficha amarilla | Mover a celda disponible en el eje vertical considerando la posición de la ficha roja             | bloquear a la ficha roja | Tablero cuadrado, ficha amarilla como obstáculo móvil |

**Tabla REAS (Rendimiento – Entorno – Actuadores – Sensores)**

| Ficha           | Rendimiento   | Entorno  | Actuadores   | Sensores  |
|-----------------|---|--|--|---|
| <b>Roja</b>     | <ul style="list-style-type: none"> <li>- Tiempo o número de movimientos para alcanzar la salida</li> <li>- Éxito al escapar sin colisionar</li> </ul> | <ul style="list-style-type: none"> <li>- Tablero cuadrado con casillas ocupadas o libres</li> <li>- Presencia de la ficha amarilla como obstáculo móvil</li> </ul> | <ul style="list-style-type: none"> <li>- Movimiento en las 4 direcciones cardinales (↑ ↓ ← →)</li> </ul> | <ul style="list-style-type: none"> <li>- Detección de posición actual</li> <li>- Detección de celdas adyacentes libres u ocupadas</li> <li>- Detección posición de la ficha amarilla</li> </ul> |
| <b>Amarilla</b> | <ul style="list-style-type: none"> <li>- Eficiencia en bloquear a la ficha roja</li> <li>- Tiempo que logra demorar su salida</li> </ul>              | <ul style="list-style-type: none"> <li>- Mismo tablero cuadrado</li> <li>- Interacción constante con el movimiento de la ficha roja</li> </ul>                     | <ul style="list-style-type: none"> <li>- Movimiento vertical (↑ ↓)</li> </ul>                            | <ul style="list-style-type: none"> <li>- Detección de posición propia</li> <li>- Posición de la ficha roja actual y proyectada</li> <li>- Cercanía de la ficha roja con la salida</li> </ul>    |



**Tabla P→A (Percepción → Acción)**

| Percepciones   | Acciones  |
|--|---|
| 1. Posición actual de la ficha roja y la ficha amarilla          | 1. Evaluar dirección de movimiento posible hacia la meta (salida)               |
| 2. Celdas adyacentes disponibles (vacías)                        | 2. Mover ficha roja hacia la celda vacía más cercana a la salida                |
| 3. Detección de ficha amarilla en celda adyacente                | 3. Evitar movimiento hacia la celda ocupada por la ficha amarilla               |
| 4. Cálculo de trayectoria bloqueada o no hacia la meta           | 4. Planificar movimiento alternativo para esquivar bloqueo                      |
| 5. Detección de celda fuera del tablero                          | 5. No ejecutar movimiento en esa dirección                                      |
| 6. Proximidad entre ficha amarilla y ficha roja                  | 6. Ficha amarilla se mueve verticalmente para interponerse o bloquear el avance |
| 7. Detección de que la ficha roja llegó a la meta                | 7. Detener movimiento y marcar éxito (estado de escape alcanzado)               |
| 8. Detección de que no existen movimientos posibles sin colisión | 8. Marcar situación de fracaso (sin escape posible)                             |