

# Implementación de un simulador para la comparación de estrategias de resolución de problemas de un agente computacional basado en objetivos

## Resumen

El objetivo de este trabajo fue construir un agente inteligente basado en objetivos, para comprender como éste se relaciona con el mundo en el cual se desenvuelve y como utiliza las diferentes técnicas existentes para tomar las decisiones sobre las acciones que puede emprender. Se implementó un entorno de simulación, en donde observamos las diferentes estrategias que un agente puede llevar a cabo en laberintos predefinidos. Luego, sacar conclusiones sobre qué tipo de agente es mejor incluir en una u otra situación donde se lo quiera implementar.

## Introducción

Este trabajo surge en el marco de la Cátedra Inteligencia Artificial de la UTN Facultad Regional Santa Fe con el fin de aplicar los conocimientos adquiridos en un caso práctico.

El problema que se planteó consiste en desarrollar un agente inteligente capaz de desenvolverse en el escenario planteado, el cual consiste en 3 laberintos interconectados entre sí de tal modo que las salidas de un laberinto se corresponden con la entrada del siguiente, tal como se muestra en la siguiente figura.



Cada laberinto contiene caminos de los cuales al menos uno de ellos permite ir desde la entrada a la salida.

Las salidas del laberinto pueden tener puertas, las cuales pueden estar cerradas con candado. Para abrir una puerta el agente debe estar ubicado en la misma casilla en donde se encuentra, y si la misma está cerrada debe poseer la llave correspondiente para abrirla.

Para la resolución de los laberintos, Ronly utiliza dos estrategias de resolución de problemas: **Búsqueda y Cálculo Situacional** a fin de poder luego comparar los resultados obtenidos.

## Metodología

Cuando utilizamos búsqueda el agente percibe el laberinto que tiene que resolver en cada nivel (mundo totalmente observable), conociendo la posición exacta de las entradas, salidas, de cada pared, de las llaves y si las puertas están abiertas o cerradas.

Cuando utilizamos cálculo situacional, el agente recibe una percepción correspondiente a los elementos ubicados en la celda en la que está ubicado. Esto es, se le indican las paredes a su alrededor y los objetos (si los hay) que se encuentran en dicha celda.

A fin de resolver las consignas planteadas, utilizamos el lenguaje JAVA, un framework para simular agentes inteligentes denominado FAIA e IDEM-IA, una herramienta que permite modelar gráficamente el problema.

### Modelado Conceptual

Ronly
<ul style="list-style-type: none"> <li>- Posición: [fila, columna]</li> <li>- Orientación: {Norte, Sur, Este, Oeste}</li> <li>- Posee llave?: {Si, No}</li> <li>- Mapa del laberinto (Búsqueda)</li> <li>- Características de la celda actual (Cálculo Situacional)</li> </ul>
<ul style="list-style-type: none"> <li>+ Avanzar ()</li> <li>+ Girar 90° a la izquierda ()</li> <li>+ Girar 90° a la derecha ()</li> <li>+ Tomar la llave ()</li> <li>+ Abrir puerta ()</li> </ul>

Un modelo conceptual de nuestro agente, identificando su estado, posibles acciones, y la prueba de meta que le permitirá saber si ha alcanzado su objetivo.

### Implementación del Estado del Agente

En el caso de Búsqueda, el estado está definido por una clase en el modelo de objetos la cual no difiere más que en cuestiones de implementación de la representación conceptual.

En el caso de Cálculo Situacional, el estado está definido por un conjunto de predicados Prolog, los cuales se mantienen en una Base de Conocimiento dentro del objeto agente.

### Solver para Búsqueda

Ronly utiliza dos estrategias de búsqueda distintas: A\* y Costo Uniforme.

Para A\* se diseñó una heurística que elige la menor de las distancias entre el agente y cada una de las salidas. Si la salida está cerrada con candado y Ronly no posee la llave para abrirla, entonces, la distancia que se calcula es la de Ronly hasta la llave más la distancia de la llave hasta la salida con candado.

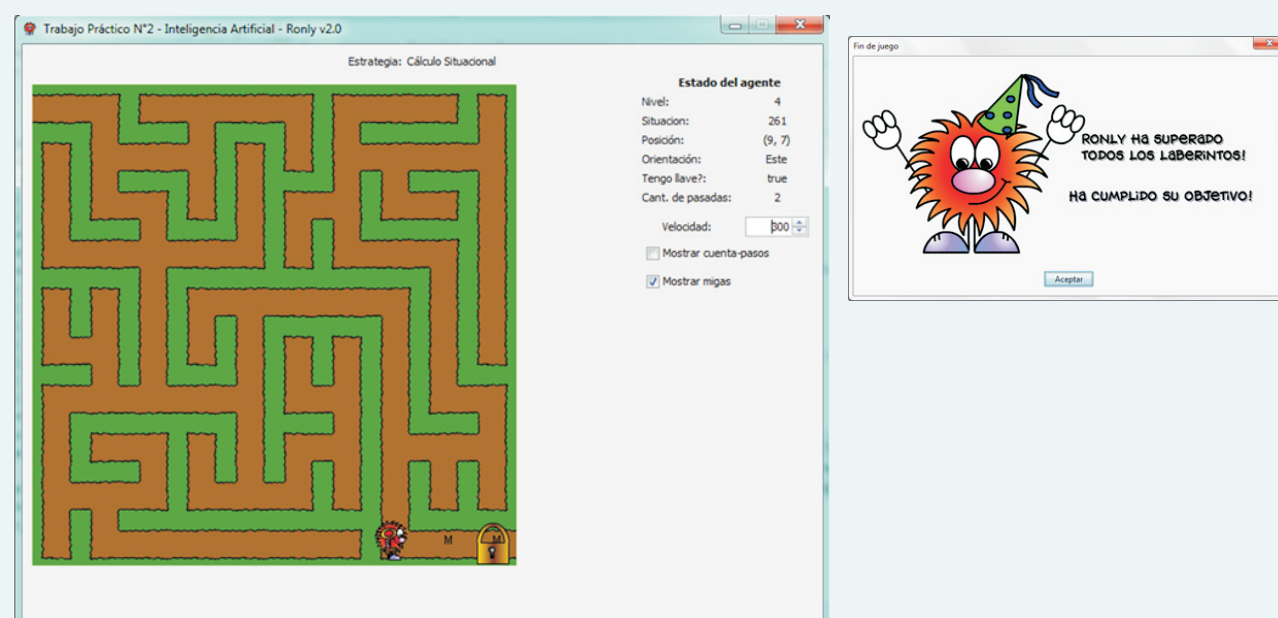
### Solver para Cálculo Situacional

Utilizamos distintas estrategias de valoración de acciones para decidir cuál será la próxima acción a tomar como "Celda menos visitada", "Regla de la Mano Derecha" y "Algoritmo de Hansel-Gretel".

### Representación del Estado del Ambiente

El ambiente almacena tanto el conjunto de laberintos contiguos como la ubicación de Ronly en el instante actual.

Cada vez que el agente decide ejecutar una acción o secuencia de acciones, una lista de acciones es pasada al ambiente por el simulador, y éste entonces actualiza su estado.



## Autores

**CORNIER, Maria de las Nieves**

mcornier@frsf.utn.edu.ar

**MANTOVANI, Valentino**

vale\_manto@hotmail.com

**SCHNIDRIG, Manuel**

mschnidrig@frsf.utn.edu.ar

## Nuestro Agente

Le hemos dado vida a nuestro agente resuelve-laberintos. Su nombre es Ronly, tiene 22 años, vive en la capital de la provincia de Santa Fe y tiene una amplia experiencia en la resolución de puzzles lógicos y laberintos.

Ronly puede desplazarse en las direcciones Norte, Sur, Este, Oeste. Para esto dispone de dos motores. Uno le permite girar sobre su eje 90° hacia la izquierda o hacia la derecha, mientras el otro le permite desplazarse hacia adelante en la dirección en la que se encuentra.



## Resultados

Debemos notar que a diferencia de todas las demás estrategias, A\* es una estrategia informada, y por tanto es intuitivamente la que resultará en el camino más corto frente a las demás.

Se hizo primero una comparación de las estrategias de búsqueda A\* y dos versiones de una estrategia de Costo Uniforme (donde las acciones avanzar y girar tenían diferentes costos).

En cuanto al Cálculo de Situaciones, la solución no es la óptima ya que a diferencia de A\*, Ronly no percibe el laberinto completo para poder decidir qué acciones conviene tomar para resolverlo. Además no se detiene al principio del laberinto a buscar una solución completa a su problema si no que va resolviéndolo a medida que avanza.



Estos arboles muestran los nodos que han sido expandidos en la búsqueda del camino a la salida.

## Conclusión

Este trabajo nos ha permitido experimentar las etapas básicas por las que debe pasar un equipo que se involucre en la construcción de un agente inteligente, desde el modelado conceptual hasta la implementación del mismo. Hemos comparado distintas técnicas de resolución de problemas basados en objetivos, conociendo cómo funciona cada una de éstas y pudiendo así sacar conclusiones tanto de performance como de conveniencia.