

# Proyecto Inteligencia Artificial II- Salida del Laberinto

Freen Esprint Jimenez Torres\*  
Universidad de la Amazonia, Florencia

Noviembre 01, 2012

## 1 Descripción del problema

Por definición, un laberinto no es más que una estructura formada por calles y encrucijadas, normalmente compleja, que intenta conseguir la confusión en quien en ella se adentra. El nombre de laberinto proviene del latín “labyrinthus” y del griego “labýrinzos”. (Isabel Hernández Fernández, 2010)

Desde ese enfoque, la resolución de problemas se describe a menudo como una búsqueda en un enorme laberinto de posibilidades, un laberinto que describe el entorno. Para resolver exitosamente un problema se requiere explorar el laberinto de forma selectiva y con ello reducirlo a proporciones manejables. Pero los problemas que surgen a la hora de implementar una solución basada en búsqueda de laberintos son los siguientes entre otros:

- Las máquinas aún no pueden reducir automáticamente los problemas a proporciones manejables, es por ello que los seres humanos tienen que formular los problemas y proponer estrategias para encontrar su solución.
- La complejidad que amerita encontrar la salida de un laberinto, donde su estructura es desconocida.

Debido a estos problemas, se debe hacer uso de técnicas de inteligencia artificial que proporcione las herramientas necesarias para la solución de este tipo de inconvenientes.

## 2 Estado del arte

A continuación se plantean diferentes soluciones ya realizadas por algunos autores, en la resolución de laberintos por medio de la inteligencia artificial.

---

\*Estudiante X semestre de ingeniería de sistemas

## 2.1 TÉCNICAS DE INTELIGENCIA ARTIFICIAL PARA LA SOLUCIÓN DE LABERINTOS DE ESTRUCTURA DESCONOCIDA.

En este trabajo se presentan las técnicas de búsqueda en amplitud y en profundidad, las cuales son técnicas de inteligencia artificial, para la solución de un problema de gran complejidad matemática como lo es la solución de un laberinto de estructura desconocida.

El primer requisito que debe cumplir una buena estrategia de control es que cause algún cambio, las estrategias de control que no causen cambio de estado nunca alcanzan la solución. El segundo requisito que debe cumplir una buena estrategia de control es que sea sistemática.

Para tener una buena claridad de las estrategias de control en la solución de laberintos, se presenta en la Figura 1, a modo de ejemplo un laberinto que un móvil, representado por un círculo negro, desea solucionar; siendo la solución del laberinto el triángulo negro. Las paredes del laberinto son cuadros grises y las posiciones en el laberinto se representan por una letra acompañada por un número, de esta forma la posición inicial se localiza en E9 y la posición final en H1. En la figura 2 se presenta un árbol de búsqueda para el laberinto de la Figura 1. En la raíz del árbol se encuentra el estado inicial, todas las ramificaciones de la raíz se generan al aplicar cada una de las reglas al estado inicial. Cada bifurcación en una rama representa un punto del laberinto con varios caminos y una rama de la que no sale ninguna bifurcación puede ser un callejón sin salida o la solución.

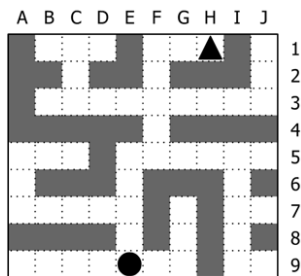


Figure 1: Laberinto

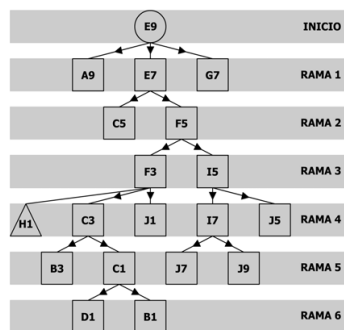


Figure 2: Árbol de búsqueda

## SOFTWARE DESARROLLADO

Se desarrolló utilizando Wx Dev C++ un software para el estudio de los diferentes métodos de solución de laberintos llamado Teseo. En la Figura 3 se presenta la interfaz gráfica de Teseo con la cual se pueden construir diferentes tipos de laberintos como el que se muestra a continuación:

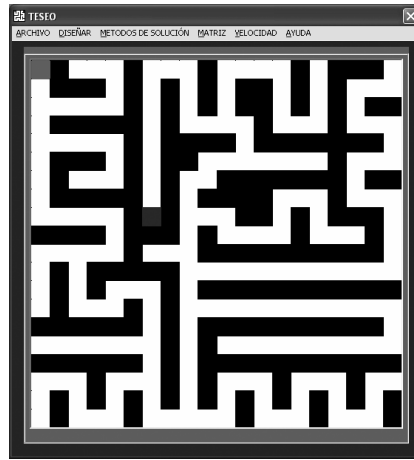


Figure 3: Laberinto

El método de búsqueda en amplitud obtiene en promedio los mejores tiempos de solución a pesar de tener que recorrer todo el laberinto empleando más memoria para realizar la búsqueda, por otro lado, la búsqueda en profundidad es más aleatoria en cuanto al tiempo de solución, no obstante puede obtener mejores tiempos que la búsqueda en amplitud y menor cantidad de memoria, pero en promedio sus tiempos de búsqueda son mayores. Dependiendo del tipo de laberinto ingresado, cada método empleado entrega una solución mejor que la otra.

## 2.2 Buscador de caminos

Aquí tenemos un buscador de caminos, el ordenador podrá encontrar una salida de el, buscara el mejor camino para salir y lo mostrara en pantalla, se puede modificar el punto de salida y el punto de llegada, osea el inicio del laberinto y el final, ademas en este buscador de caminos se puede modificar el laberinto, agregando o quitando muros, casillas.

El buscador de caminos consiste en un laberinto con una serie de obstaculos, se define un punto inicial y un punto final, y la inteligencia artificial buscara el camino mas optimo para llegar desde un punto a otro, la innovacion de este laberinto es que se puede modificar, cambiando por completo su estructura.

El buscador de caminos esta programado en flash, siendo la funcion de busqueda completamente action script.

## :: Buscador de caminos v.01 ::

vea más en: [diseñador web flash \[ www.cerotec.net \]](http://diseñador.web.flash[www.cerotec.net])

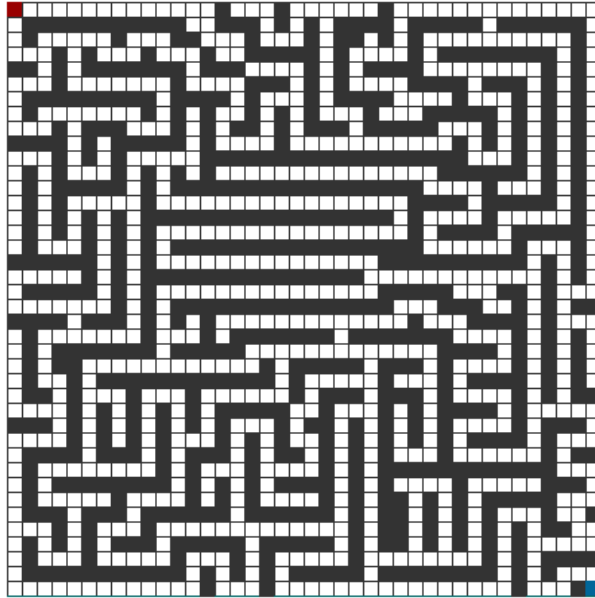


Figure 4: Buscador de caminos

### 3 Propuesta

Como propósito de la clase de Inteligencia Artificial II, se ha propuesto desarrollar un juego que implique el uso de técnicas computacionalmente inteligentes, dicho juego está planteado de la siguiente forma:

1. El juego va constar de un laberinto y un personaje.
2. El laberinto va tener una entrada y una salida respectivamente.
3. Como mínimo, el laberinto debe tener un camino libre de obstáculos, que conecte la entrada con la salida.
4. El personaje tiene que estar en la capacidad de salir del laberinto automáticamente.
5. El juego tiene la posibilidad de generar nuevos laberintos, donde la entrada y la salida se puedan mover a decisión del usuario.
6. Por motivos de optimización, el laberinto va a tener un ancho y alto predefinido de manera estática.

Para desarrollar este juego se piensa hacer uso de técnicas computacionalmente inteligentes como las “Redes neuronales supervisadas, altamente conexionistas”, ya que las redes neuronales son las más apropiadas, porque se basan en un aprendizaje continuo que a partir de unos parámetros de entrada y de salida, pueden ser entrenadas de manera adecuada, para así satisfacer el requerimiento deseado. Además deben ser altamente conexionistas para permitir un flujo de información libre entre todas las neuronas, dando así un mejor aprendizaje por parte de la red.

La red neuronal que se piensa a implementar es una red perceptron multicapa, ya que es una de las redes más comunes a implementar y posee una gran capacidad de aprendizaje y adaptación en problemas computacionalmente complejos.

Además, La red neuronal va a tener tres capas implementadas, una capa de entrada, una capa oculta y una capa de salida, tal como lo muestra la siguiente figura:

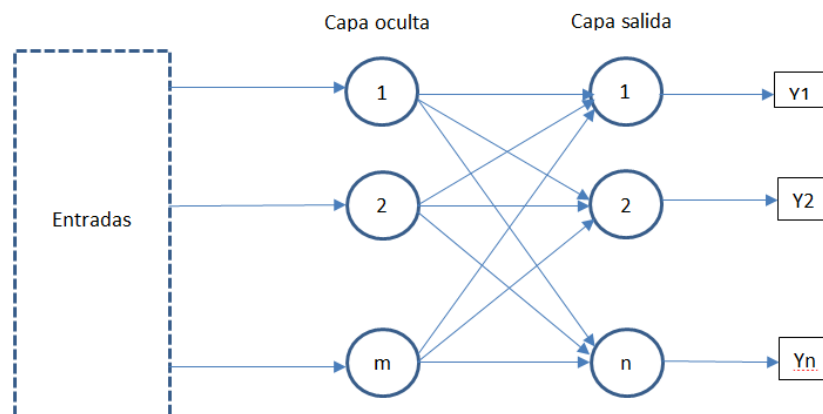


Figure 5: Arquitectura red neuronal

En la capa de entrada van a ir todos los posibles caminos por los cuales el personaje puede pasar.

En la capa oculta dependiendo el entrenamiento, el número de iteraciones y los resultados obtenidos, se ira añadiendo o quitando neuronas, inicialmente se tiene la intención de arrancar con 5 neuronas en esta capa.

En la capa de salida se piensa hacer uso de métodos de búsqueda por medio de inteligencia artificial, para encontrar la salida del laberinto más óptima.