Estudiante:

Luis Alejandro Salas Rojas

Carne:

2014010742

Profesora:

María Mora Cross

Curso:

Inteligencia Artificial

Proyecto programado 1:

Conecta 4

Sede Interuniversitaria de Alajuela

II semestre del año 2018

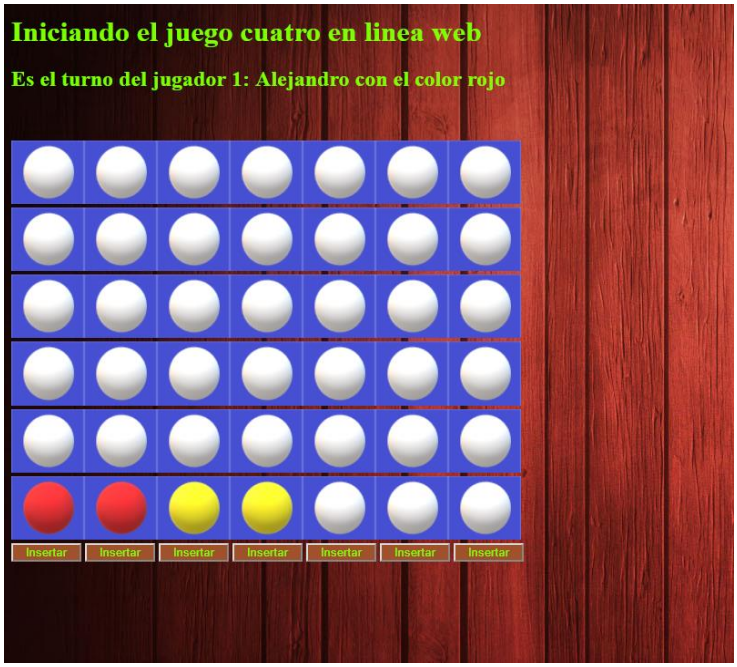
Lunes 08 de octubre de 2018

Descripción del sistema:

El juego conecta cuatro es un juego que permite al usuario pensar de una manera más realista la estrategia que desea aplicar para llegar a ganar en este, en este sistema el juego del cuatro en línea se va desarrollar de una manera con java web y para tener una mejor visión de este, nos vamos a guiar con la siguiente imagen.

Basados en esta imagen que es la pantalla principal o donde inicia el juego podemos ver como se nos pide que introduzcamos nuestro nombre esto para que el juego referencia el nombre de la persona que está jugando contra la computadora pero si el usuario no digita ningún nombre igual la aplicación continua y simplemente lo referencia con un mensaje de “Turno de jugador SIN NOMBRE”, también se muestra un tablero vacío en donde se ve como es el juego con su tablero totalmente vacío y por último en esta pantalla se nos pide que elijamos mediante un botón entre “Fácil”, “Medio” o “Difícil”, esto lo que va a definir el grado de profundidad que va tener el algoritmo que estamos utilizando, que la fácil implica una profundidad de 4, la medio implica una profundidad de 6 y la difícil implica una profundidad de 8.

Por otra parte, al elegir el grado de dificultad que nosotros deseamos este se va dirigir a la pantalla principal que es donde se concentra la mayor parte del juego y para entender mejor como funciona nos vamos a guiar con la siguiente imagen.



En esta pantalla se logra apreciar donde el juego indica el turno de cada uno de los jugadores arriba del tablero, y con cada uno de los botones que se encuentran debajo del tablero se puede ir insertando en cada una de las filas que deseemos, el color rojo es para el jugador 1 y el color amarillo es para el jugador de la computadora, también cabe mencionar que apenas se logre un cuatro en línea(cuatro fichas seguidas del mismo color) este se va redirigir para una pantalla de gane que se muestra en la siguiente imagen y si se da el empate(todo el tablero lleno y nadie logro el cuatro en línea) se envía a una pantalla donde se imprime el tablero lleno y se imprime un mensaje que se dio un empate entre los dos jugadores.

En esta pantalla de gane se muestra como el jugador 1 perdió contra la computadora y se imprime el tablero justo como quedo con el gane, también se hizo un botón que reinicia la aplicación por si el usuario quiere volver a jugar.

Descripción de la arquitectura:

JSP's creados:

- ✓ PaginaWEB: pagina encargada de arrancar la aplicación en donde se pide el nombre del usuario, y en donde el usuario puede elegir la dificultad del juego y por último en esta pantalla se muestra al usuario como va ser el tablero principal ya que este se imprime totalmente vacío.
- ✓ Jugadas: pantalla en donde se concentra la mayoría del juego, en este cada turno va pidiendo al jugador que elija la columna donde desea insertar, al recibir la columna este lo que hace es poner la posición del usuario y pedir a la computadora que haga su movimiento para así volver a imprimir el tablero actualizado con cada movimiento.
- ✓ PantallaEmpato: pantalla en donde se muestra el tablero y un mensaje indicando que hubo un empate, también se vuelve a imprimir el tablero para que el usuario lo vea completamente lleno, por último, se creó un botón en donde se reinicia el juego, ósea este vuelve a la página principal y reinicia la matriz en cero y algunas variables importantes.
- ✓ PantallaGano: pantalla en donde se muestra el jugador que gano, y se muestra un mensaje para que este lo identifique, por último, se vuelve a imprimir el tablero con una imagen diferente en donde se formó el cuatro en línea para que el usuario note donde fue que gano, por último, se creó un botón en donde se reinicia el juego, ósea este vuelve a la página principal y reinicia la matriz en cero y algunas variables importantes.

En el paquete Servlet:

- ✓ InformacionMovimientos: este .java es el servlet de este proyecto, lo que hace es recibir los datos por medio de el "post", y lo que puede recibir es; el nombre del usuario, el id de las columnas, y el id de los botones cuando el usuario decide reiniciar, este servlet lo que hace es obtener los datos y hace comparaciones para saber si el usuario decidió reiniciar o si el usuario digito una columna para llamar a las funciones de insertar en el tablero, también después de cada inserción en la matriz del tablero hace las comparaciones respectivas para saber si el usuario gano al hacer esa inserción en el tablero.

En el paquete Lógica:

- ✓ Controlador: en esta clase lo que se hace es recibir el id de la columna que el usuario elige y mediante la función de moverFichaUsuario que recibe la columna lo que se hace es insertar en la respectiva posición en la matriz del tablero, después de que la bandera de que el usuario movió se activa le permite a la computadora hacer su respectivo movimiento llamando a la función moverFichaCompu que lo que hace es llamar al minimax e insertar con la columna que devuelve el minimax.

- ✓ Minimax: en esta clase se define la parte de la inteligencia artificial que lo que hace es devolver una posición en la que la el turno de la computadora va insertar en la matriz lógica, para tener un mejor entendimiento vamos a explicar un poco mejor cada una de las funciones que contiene esta clase:
    - Inicio(): esta función es la que inicia el método del minimax, inicia la variable columna en -1, llama a la función minimax() y al final retorna la columna que devuelve el minimax.
    - Minimax(): esta función recibe lo que la profundidad inicial y un turno que es más que todo para manejar la lógica dentro del algoritmo, básicamente lo que hace es preguntar si alguno de los jugadores ya gano, sino compara si la profundidad inicial es igual a la profundidad máxima del algoritmo y si son iguales llama a la función de evaluar(), si esto no se da lo que se hace es recorrer un for para el largo de las columnas y se va insertando en cada una de las posiciones de la matriz un 2 que corresponde a la ficha de la máquina, también se obtiene la puntuación evaluando el minimax nada más que ahora la profundidad inicial se le suma uno, después de esto se verifica si la puntuación es mayor que el resultado máximo y así se asigna la columna que devuelve el minimax. Lo de los turnos es para ir agregando para los dos jugadores y así ir haciendo posibles movimientos para ver la mejor opción.
    - Remover(): esta función lo que hace es recorrer la columna e ir poniendo 0 donde no lo hay.
    - calcularResultado(): recibe el contador de la máquina y el número de campos vacíos para la columna y se aplican unas fórmulas para dar un resultado al ser multiplicado dependiendo de los datos ingresados.
    - Evaluar(): esta función recibe una referencia de tipo tablero para poder tener acceso a este, seguidamente se crea dos for para recorrer cada una de las posiciones de la matriz, después se recorre desde un distinto punto de la matriz hacia la dirección derecha, arriba, diagonal izquierda y diagonal derecha, en cada una de las iteraciones le va sumando a un contador que lleva el numero de 2 que hay seguidos para después enviarlo a una función que calcula el resultado y asigna un puntaje.
- ✓ Tablero: en esta clase se crea una matriz de Strings y el método que inicializa esta clase se hace mediante un método de singleton para que al volver a inicializar esta clase no se cree la matriz de nuevo ni tampoco se reinicien las variables que están en esta. También aparte del constructor que recibe el número de filas y numero de columna este tiene las siguientes funciones:

- Insertar(pos): esta función recibe la columna en la que se va insertar y la recorre de abajo hacia arriba para saber dónde hay un cero, al encontrarlo compara cual es el jugador y si es 1 inserta un 1 en esa posición, también esta función cambiar el jugador actual y el jugador anterior para llevarlo a la hora de validar, también se guarda la posición actual mediante filaActual y columnaActual y si en esta inserción la hizo el jugador 1 pone esa variable en 0 para dar el turno a la computadora.
- Limpiar(): esta función lo que hace es reiniciar la matriz en 0 y reiniciar los jugadores actual y anterior.
- verMatriz(): lo que hace es recorrer la matriz e ir imprimiendo cada una de las posiciones para verificar que al insertar se está cambiando la matriz.

✓ Validaciones: en esta clase lo que se hace es ir recorriendo la matriz para ir realizando las respectivas validaciones, dentro de esta clase existen las funciones de:

- hayGanador(): esta función recorre la matriz lógica y va comparando si existen 4 fichas del mismo tipo, realiza el recorrido mediante for que van sumando a un contador llamado contadorMaquina y contadorUsuario y después compara cuál de los dos es 4 devuelve la misma ficha(1 o 2) si alguno de los jugadores gano. También se va llenando una lista para recordar cuales posiciones formaron el 4 en línea y así imprimir diferente.
- filaValida()/columnaValida(): en esta función se analiza si la fila o columna no se sale de los rangos de la matriz y también valida si esta es diferente de 0, esto quiere decir que la ficha solo es válida si su columna es mayor o igual a 0 y menor que 7, si su fila es menor que 6 y mayor o igual a 0; también es válida si su valor es 1 o 2.
- llenarMatriz(): esta función lo que hace es cambiar las posiciones de la matriz en donde el usuario formo el cuatro en línea para que esta se pueda imprimir de una forma diferente.

Descripción del agente y sus componentes:

Medidas de desempeño: rapidez al mover cada ficha según la profundidad y el número de movimiento que es, busca la mejor solución al realizar movimiento de la máquina, gana de una manera eficaz.

Entorno: Tablero.

Sensores: botones.

Actuadores: algoritmo minimax, inserción en la matriz.

- ✓ Estados: matriz[i][j]=”1”, matriz[i][j]=”2”, matriz[i][j]=”3” o matriz[i][j]=”0”.
- ✓ Estado inicial: matriz[i][j]=”0”.
- ✓ Acciones: insertar columna[i]; i=0,1,2,3,4,5,6. Reiniciar, Fácil, Medio, Difícil.
- ✓ Modelo de transición:
    - Para el usuario: insertar(columna)=matriz con la columna modificada.
    - Para la maquina: insertar(minimax())= matriz con la columna modificada.
- ✓ Prueba de objetivo: mediante la función de ganar() y la de empatar() verifica si el juego ya termino.
- ✓ Costo de la trayectoria: cada movimiento tiene un costo igual para el usuario, pero para la computadora varía dependiendo del número de jugadas que hayan pasado y además de la profundidad del algoritmo.

Descripción de la función evaluar del algoritmo minimax:

Esta función recibe una referencia de tipo tablero para poder tener acceso a este, seguidamente se crean dos for para poder recorrer cada una de las posiciones de la matriz.

Los siguientes pasos se hacen para recorrer la fila a la derecha, la columna hacia arriba, la fila hacia la izquierda, la diagonal arriba derecha, y por último la diagonal arriba izquierda:

> El siguiente paso es ir recorriendo cada posición de una manera en la que se forme tres posiciones seguidas que tengan “2”, por cada una de las posiciones que se va encontrando un dos se suma a un contador y su se encuentra un “1” se reinicia el contador, y también el contador que verifica las celdas vacías.
>
> Después de este paso, lo que se hace es que si se encontraron espacios en blanco se va ir a recorrer cada una de las celdas para ver cuántos espacios alrededor de esa ficha y a un largo de 3 máximo son válidos para esta jugada.
>
> Después de ver si los campos vacíos son mayores que 0 se hace en el if un puntaje y después de esto se reinicia el contador de la maquina (el número de “2”). El puntaje del que se habla se obtiene de la función calcular resultados que uno le manda el contador de la maquina (el número de “2”) en la respectiva iteración y además de esto se le manda el numero de 0’s que esta iteración tiene. Cabe mencionar que esta función se da después de que el algoritmo haya alcanzado la profundidad máxima.

Después de cada uno de estos pasos se puede decir que la puntuación de cada jugada va dada por la multiplicación que se realiza en la función de calcularResultado ya que en esta se multiplica la cantidad de posibles movimientos dependiendo de si es 1 se multiplica por 1, si es 2 se hace por 10, si es 3 se hace por 100 y sino se retorna 1000.

También cabe mencionar que en el primer movimiento de la computadora se aplicó una heurística para que inserte en la columna del centro para así mejorar la velocidad del juego y aumentar el rendimiento.

Gráficos de pruebas variando la profundidad:

Gráfico de profundidad 2:

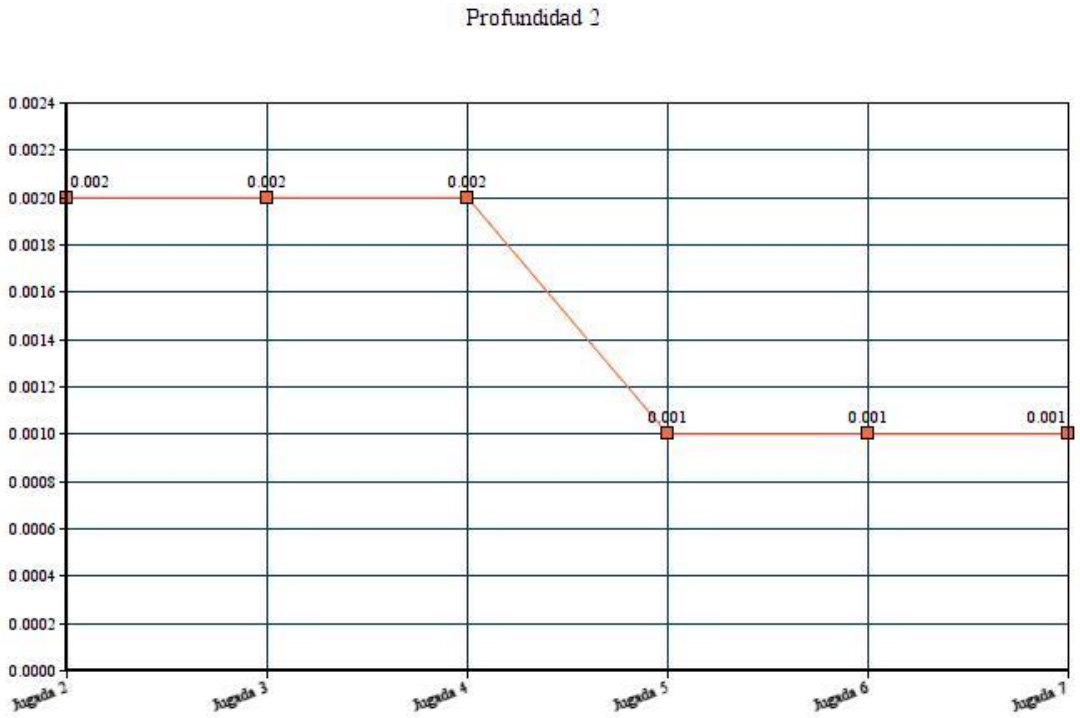


Gráfico de profundidad 4:

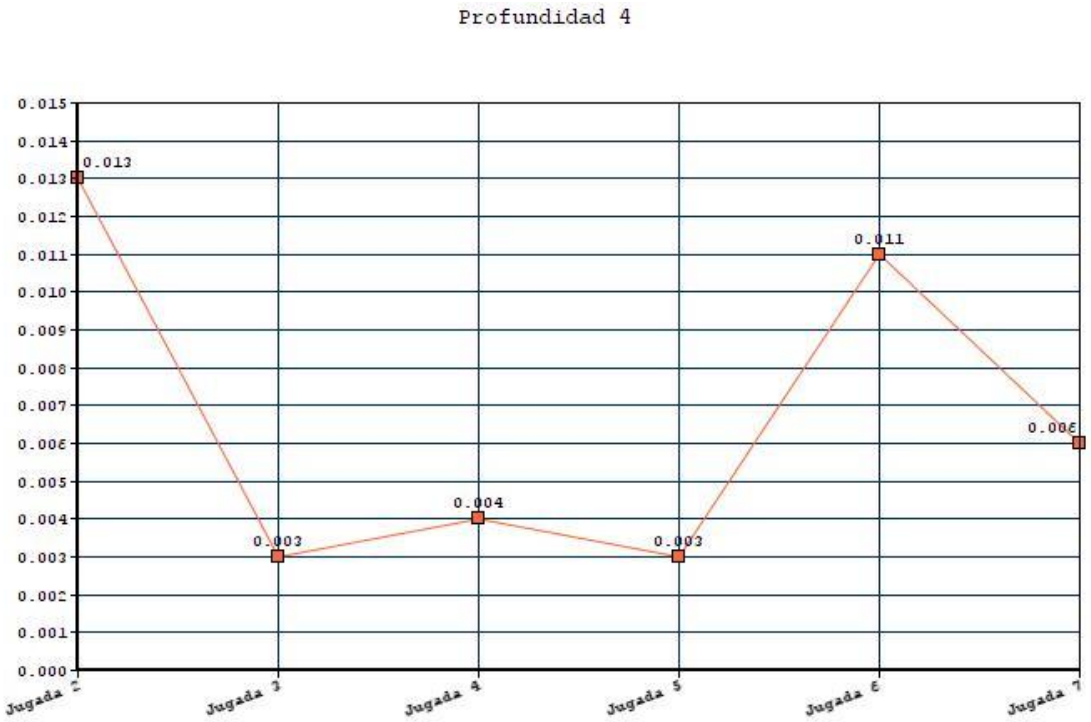


Gráfico de profundidad 6:

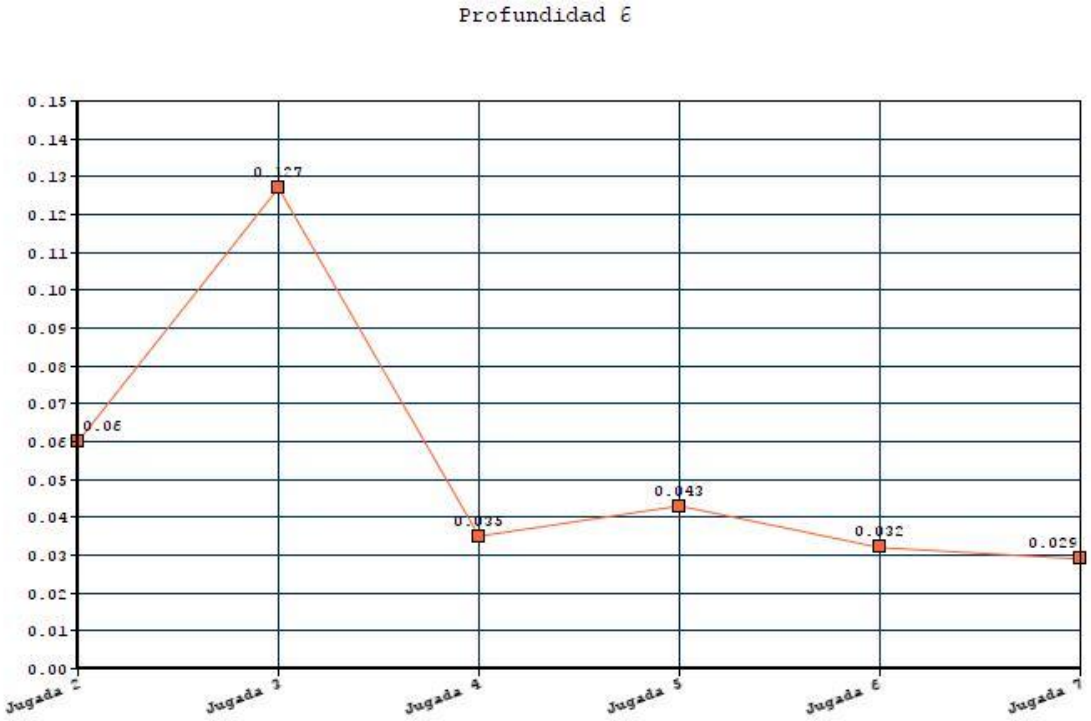


Gráfico de profundidad 8:

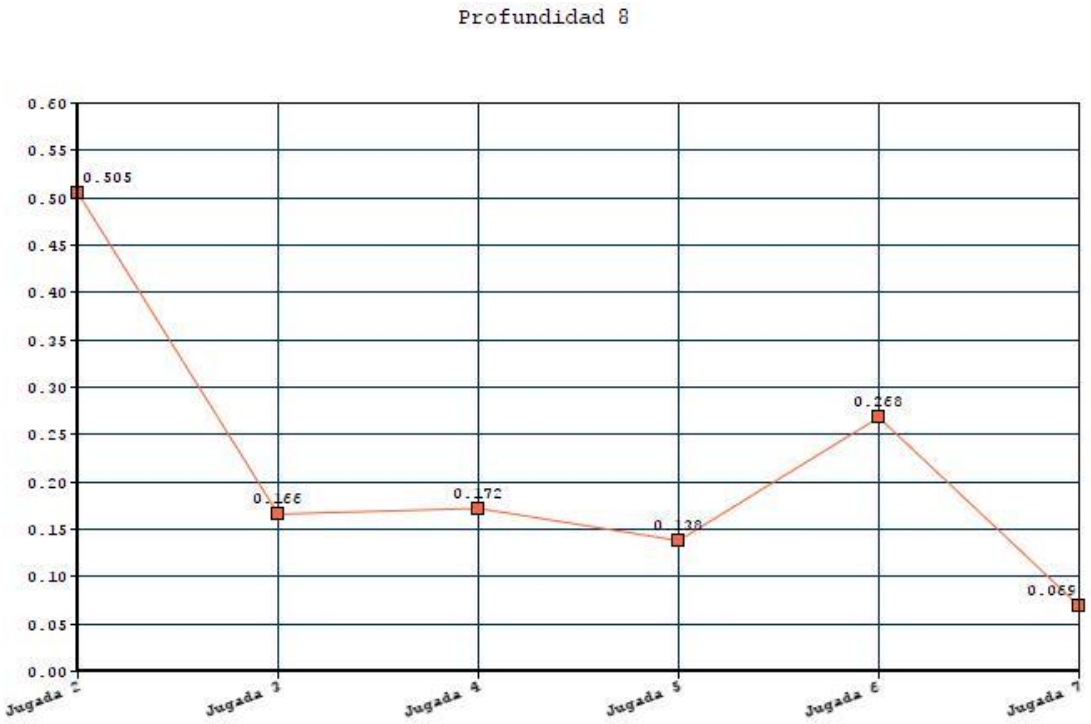


Gráfico de profundidad 9:

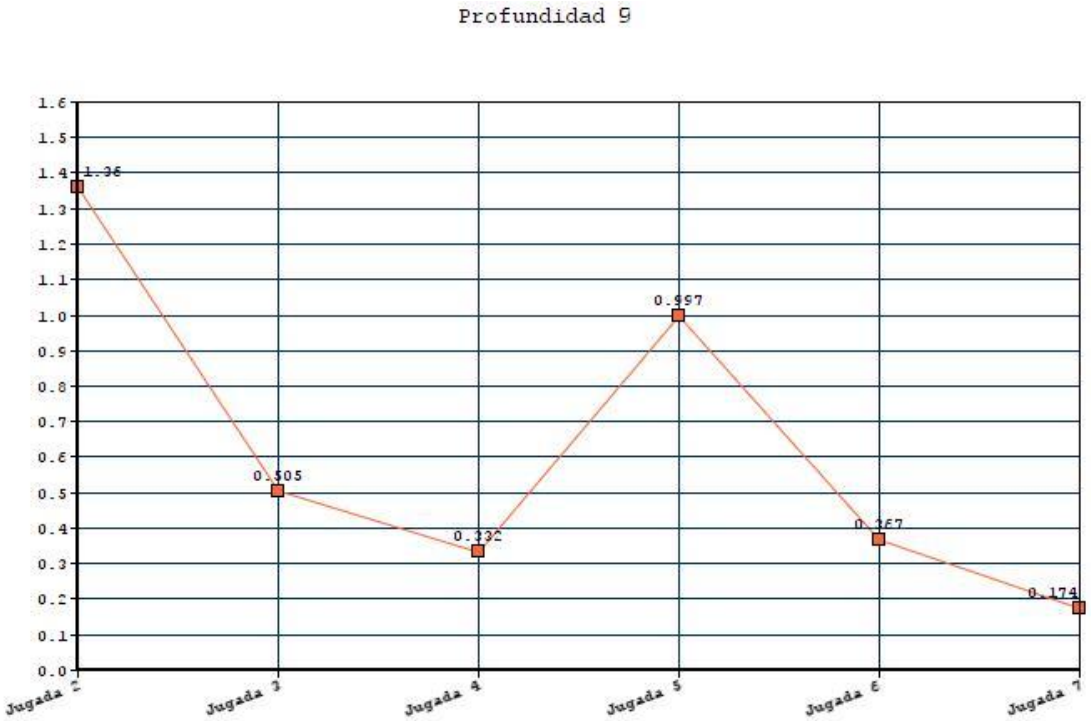
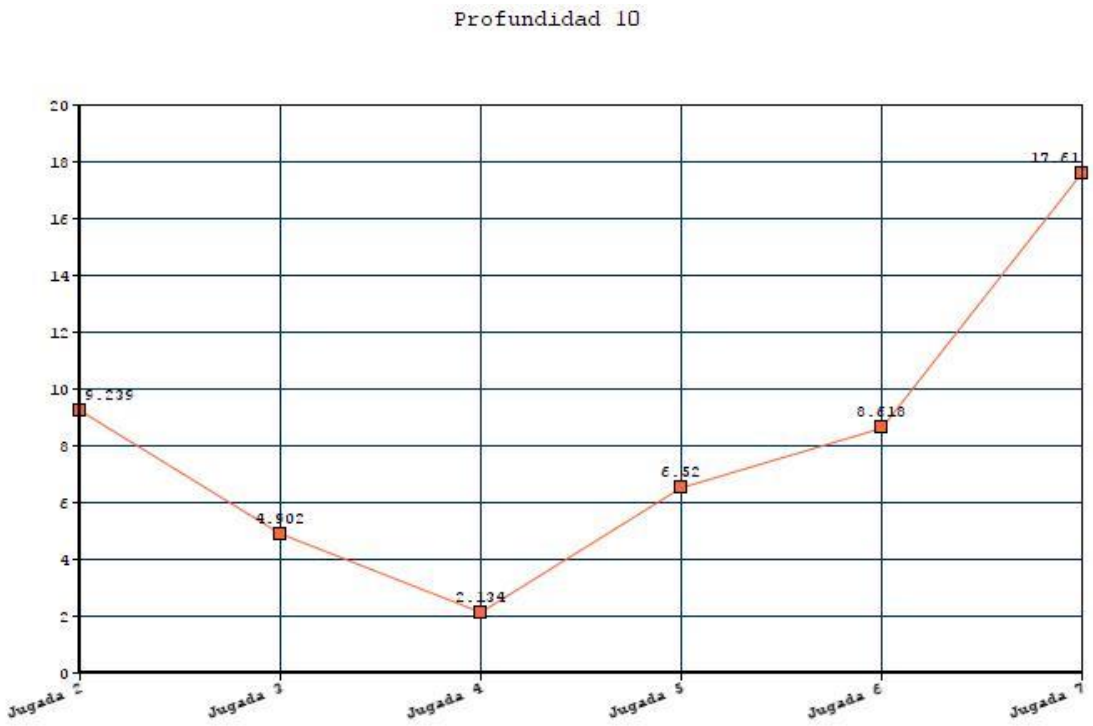


Gráfico de profundidad 10:



Referencias

- Solving Connect 4: how to build a perfect AI. (2018). Solving Connect 4: how to build a perfect AI. Recuperado el 07 de octubre de: http://blog.gamesolver.org/
- En.wikipedia.org. (2018). Connect Four. [online] Recuperado el 07 de octubre de: https://en.wikipedia.org/wiki/Connect_Four