

# Informe TP1 “Reversi”

Materia: Algoritmos y Programación I 75.40

Catedra: Wachenchauzer

Primer cuatrimestre 2016

## 1- Dificultades a afrontar:

Luego de analizar el enunciado del trabajo practico, se encontraron las siguientes dificultades, las cuales será necesario analizar cómo resolver:

1. Generar un tablero en memoria que se pueda almacenar datos en cada casillero
2. Modificar cada casillero individualmente
3. Mostrar el tablero en pantalla
4. Comer las fichas del oponente

## 2- Resolución:

Para poder generar un tablero en pantalla que se pueda modificar cada casillero individualmente lo que hay que hacer es utilizar una lista, ya que al ser una estructura mutable se puede modificar cada elemento por separado sin ningún problema. Ya que en este caso se necesitaban de filas y columnas se necesitaría una estructura de for anidado para crear la lista que funcionaria del tablero, tiene que ser una lista de dos dimensiones.

Luego para mostrar el tablero en pantalla lo que hice fue ir imprimiendo posición por posición de la lista del tablero, al igual que para generar el tablero se necesitó de una estructura de for anidado para obtener la posición de fila y columna a imprimir.

En este momento surgió el problema de como el usuario nos va a decir en qué posición del tablero desea posicionar su ficha, para eso habría que imprimir una letra por columna y un número por fila. Para las columnas cree una tupla, ya que no necesitaba que la estructura sea mutable, de lo contrario, mejor sea inmutable, el abecedario no tiene por qué modificarse. Solo quería que estén todas las letras del abecedario, una en cada posición de la tupla. Con la misma dimensión del tablero se recorre la tupla y se imprime una letra por columna.

Los numero de las filas los imprimí en el primer for del for anidado, que justamente el primer for es la columna y el segundo for imprime la fila.

Después llega el momento en que el usuario debe ingresar la ficha, para eso decidí que el usuario me ingrese las coordenadas de la posición donde desea ubicar la ficha en dos líneas diferentes, la columna y la fila. La columna me la va a dar con una letra y la fila con un número. La letra la transformo en un número por la posición de esa letra en la tupla abecedario.

Luego para ver si la posición ingresada es válida me fijo que la columna sea una letra del abecedario, la fila un número y que ambos estén dentro de la dimensión del tablero. Si la coordenada ingresada no existe, que ingrese devuelta otras coordenadas.

Para imprimir las cuatro fichas iniciales lo que hice fue asignarle a las posiciones del centro del tablero de ocho por ocho las letras N y B, como lo indicaba la guía del trabajo practico.

Ahora viene la parte más difícil del programa, la parte de ver si hay una jugada valida en la posición que ingreso el usuario, para luego poder comer la ficha del oponente. Para eso hay que verificar en las ocho direcciones posibles (en los bordes del tablero hay menos de ocho direcciones posibles, eso hay que tenerlo en cuenta) si se puede comer una ficha del oponente desde donde yo quiero posicionar mi ficha.

Cada vez que el jugador desee posicionar su ficha, debo preguntar por todas las direcciones si puedo comer fichas del oponente, porque puede ser pueda comer en más de una dirección a la vez.

Básicamente la idea de las ocho funciones que verifican una jugada valida en la dirección es la misma:

Primero se pone la condición que restringe el dominio en esa dirección, para esa condición se tiene en cuenta que pasa en los extremos del tablero, en cada extremo en qué dirección no se puede comer. Después lo que hago es recorrer desde la siguiente posición de la coordenada ingresada, que se fija que haya una ficha del contrario, y que siga habiendo una ficha del contrario hasta que encuentre una de mí mismo color.

Para ahorrar líneas me conviene utilizar la misma función para ver si puedo comer fichas del oponente y para comerlas, para eso tengo que poner una variable que determine si es momento de verificar si puedo ubicar la ficha en esa posición para comer fichas del oponente o si es momento de comer fichas.

Para comer la ficha del oponente recorre los casilleros que están entre donde quiero ubicar la ficha y dónde está mí otra ficha, los reemplaza todos por mis fichas.

Luego de esto tuve que resolver algunos detalles del funcionamiento del programa, como por ejemplo ver si hay una jugada posible que lo que hace es recorrer los casilleros vacíos del tablero y preguntar si hay una jugada posible para el jugador que está evaluando en ese momento, lo que hace es suponer que el jugador puede poner su ficha en cualquier espacio vacío del tablero, entonces evalúa las potenciales jugadas.

Siguiendo, debí resolver el problema de posicionar la ficha en el tablero, esto es básicamente poner la ficha en la posición que te indicó el usuario, siempre que se pueda comer por lo menos una ficha del oponente en ese lugar, y comer las fichas que correspondan.

Para finalizar tuve que resolver el problema de como decidir quién ganó la partida, o si empataron, para eso hay que contar todas las fichas que hay en el tablero, las blancas por un lado y las negras por el otro. Básicamente es recorrer e ir contando las fichas.