



---

## Introducción

La selección argentina llegó al mundial con lo justo y el DT (preocupado por la situación) mando a llamar a los estudiantes de algoritmos 3 de la carrera de Ciencias de la Computación de la FCEyN de la UBA para que le ayuden a desarrollar estrategias que le permitan llegar a la final y ganarla. Frente a este problema, los alumnos propusieron hacer un modelo computacional de los partidos de fútbol, equipos y jugadores para luego correr algoritmos genéticos sobre esos modelos y obtener estrategias buenas.

## Enunciado

El objetivo del trabajo práctico es modelar computacionalmente el fenómeno descripto, parametrizar propiedades de dicho modelo que sean controlables e implementar distintas técnicas algorítmicas que permitan conseguir buenos valores para dichos parámetros. Estos parámetros serán los generadores de distintas soluciones al problema modelado.

Se debe:

1. Describir el fenómeno a modelar y dar distintos ejemplos.
2. Definir un modelado computacional que modele el fenómeno correctamente y que permita aplicar las distintas técnicas heurísticas sobre los mismos de forma tal que una solución para el modelo sea también una solución para el problema real.

Luego, por cada método de resolución:

3. Explicar de forma clara, sencilla, estructurada y concisa, las ideas desarrolladas para la resolución del problema. Para esto se pide utilizar pseudocódigo y lenguaje coloquial combinando adecuadamente ambas herramientas (**¡sin usar código fuente!**). Se debe también justificar por qué el procedimiento desarrollado sigue la técnica pedida, para esto, se debe detallar las distintas partes del algoritmo y explicar a que lugar de la estructura general de la heurística (que se está implementando) corresponde.
4. Deducir una cota de complejidad temporal del algoritmo propuesto (en función de los parámetros que se consideren correctos) y justificar por qué el algoritmo desarrollado para la resolución del problema cumple la cota dada. **Nota:** No siempre la complejidad puede darse respecto del input ya que las técnicas a implementar pueden depender de criterios de convergencia que afectan la complejidad de los algoritmos.

5. Dar un código fuente claro que implemente la solución propuesta.

El mismo no sólo debe ser correcto sino que además debe seguir las buenas prácticas de la programación (comentarios pertinentes, nombres de variables apropiados, estilo de indentación coherente, modularización adecuada, etc.).

Por último:

6. Realizar una experimentación computacional para medir la performance del programa implementado así como sus distintas propiedades. Para ello se deben preparar distintos conjuntos de casos que permitan analizar diferentes aspectos generales como los tiempos de ejecución en función de los parámetros, las soluciones obtenidas, etc. También deberán desarrollarse experimentos que permitan medir los aspectos evolutivos de las soluciones obtenidas mediante algoritmos genéticos y que permitan comparar las distintas técnicas utilizadas en el trabajo.

**Solo se permite utilizar `c++` como lenguaje para resolver los problemas.** Se pueden utilizar otros lenguajes para presentar resultados.

La entrada y salida de los programas **deberá hacerse por medio de la entrada y salida estándar del sistema.**

Deberá entregarse un informe impreso que desarrolle los puntos mencionados.

## Problema

En este trabajo práctico, se pretende modelar un partido de fútbol donde dos equipos de 3 jugadores compiten para ver cual de los dos obtiene más goles en un período de tiempo. El equipo con mayor cantidad de goles será el ganador. En caso de que ambos equipos tengan la misma cantidad de goles realizados, el resultado será un empate. Una vez modelado el fenómeno se pide aplicar algunas técnicas algorítmicas para intentar conseguir buenas estrategias.

### Reglas del juego

El juego consiste en una cancha de fútbol de  $M$  filas y  $N$  columnas donde:

1.  $M \geq 3$  es impar.
2.  $N \geq 2$  es par  $N \geq 2M$ .
3. Cada posición es una tupla  $(fila, columna)$  donde  $fila \in [0, M)$  y  $columna \in [0, N)$
4. Un casillero puede contener:
  - Un jugador solo
  - Un jugador con una pelota
  - La pelota sola
  - Dos jugadores de equipos contrarios sin pelota

- Dos jugadores de equipos contrarios donde uno de los dos jugadores tiene la pelota
  - Dos jugadores de equipos contrarios y la pelota sin posesión
5. Los jugadores pueden desplazarse moviéndose de un casillero a otro siempre y cuando estos casilleros compartan un borde o una esquina.
  6. En caso de que un jugador se encuentre en el mismo casillero que la pelota pero no tiene la posesión de la misma, si no hay otro jugador en dicho casillero, entonces el jugador pasa automáticamente a poseer la pelota.
  7. En caso de que dos jugadores se encuentren en el mismo casillero y uno de ellos tiene la pelota, automáticamente el jugador sin la pelota intenta quitarle la pelota al otro jugador y tiene la probabilidad mostrada en (1) de lograrlo.
  8. En caso de que dos jugadores se encuentren en el mismo casillero y la pelota también está en ese casillero pero sin posesión, automáticamente pasan a disputarse la pelota. Para esto, se calcula si uno de los dos jugadores se quedó con la pelota utilizando la probabilidad mostrada en (2), si lo logra el se queda con la pelota, si no lo logra el otro jugador se queda con la pelota.
  9. El juego tiene una duración en cantidad de pasos. Cada equipo debe decidir en cada paso que movimiento realiza cada uno de sus jugadores. La duración del partido está dada por la variable  $S$ .
  10. Los movimientos posibles para un jugador son:
    - Moverse a un casillero vecino dentro del tablero, para lo cual, debe indicar una dirección de las indicadas en la Figura 1.
    - Tirar la pelota en una dirección con una cierta fuerza (medida en cantidad de pasos). La dirección de la pelota es una de las 8 direcciones mostradas en la figura 1 y la cantidad de pasos indica durante cuanto tiempo la pelota va a moverse, terminados esos pasos la pelota quedará quieta en el lugar donde se encuentre. **Nota:** La pelota se mueve en cada paso dos casilleros en la dirección indicada en vez de un solo casillero como los jugadores, además no está permitido indicar una cantidad de pasos mayor a  $\frac{N}{2}$  ni una cantidad de pasos que dejen a la pelota fuera del tablero.
  11. Un jugador intercepta una pelota libre solo si en algún momento ambos (jugador y pelota) se encuentran en un mismo casillero siguiendo las reglas marcadas en 6 y 8 según corresponda. También intercepta la pelota si la misma va de un casillero A a un casillero B y el jugador se encuentra en el casillero intermedio tanto cuando la pelota está en el casillero A como cuando la pelota está en el casillero B. En otras palabras, el jugador está en el camino de la pelota y el mismo se queda quieto mientras la pelota pasa sobre él. Si hay dos jugadores en el casillero intermedio, se utiliza la formula 2 para ver quien se queda con la pelota.
  12. El juego comienza con un marcador en 0 para ambos equipos y cada vez que la pelota entra en un arco, se suma 1 punto en el marcador del equipo contrario al arco. Cuando termina el partido, el equipo con más puntos gana, en caso de tener ambos equipos el mismo puntaje, el partido es un empate.

13. Los arcos se encuentran en las posiciones  $(\lfloor \frac{M}{2} \rfloor - 1, -1), (\lfloor \frac{M}{2} \rfloor, -1), (\lfloor \frac{M}{2} \rfloor + 1, -1)$  para un equipo y  $(\lfloor \frac{M}{2} \rfloor - 1, N), (\lfloor \frac{M}{2} \rfloor, N), (\lfloor \frac{M}{2} \rfloor + 1, N)$  para el otro equipo.
14. Los jugadores arrancan en la posición que el equipo quiera siempre que esté en su mitad de la cancha.
15. El equipo que comienza (aquel cuyo arco está en la columna  $-1$ ) tiene un jugador en la posición  $(\lfloor \frac{M}{2} \rfloor, \frac{N}{2} - 1)$  que tiene posesión de la pelota.
16. Cada vez que un equipo mete un gol, los jugadores vuelven a la posición inicial pero el equipo al que le metieron un gol tiene la pelota y por ende tiene un jugador en la posición  $(\lfloor \frac{M}{2} \rfloor, \frac{N}{2} - 1)$  si el gol fue en el arco de la columna  $N$  o  $(\lfloor \frac{M}{2} \rfloor, \frac{N}{2})$  si el gol fue en el arco de la columna  $-1$ .

$$P_{quitar}(p_{pelota}, p_{quitador}) = X \leq \frac{P_{quite}(p_{quitador})}{P_{quite}(p_{quitador}) + (1 - P_{quite}(p_{pelota}))}, X \sim U(0, 1) \quad (1)$$

$$P_{ganar}(p_1, p_2) = X \leq \frac{P_{quite}(p_1)}{P_{quite}(p_1) + P_{quite}(p_2)}, X \sim U(0, 1) \quad (2)$$

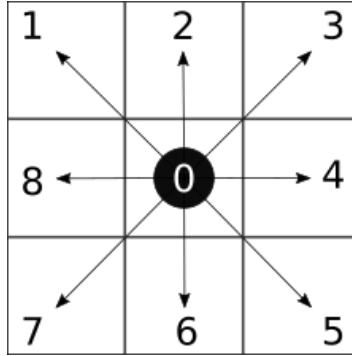


Figura 1: Cada movimiento está asignado a un número, el movimiento 0 indica que el jugador se queda quieto.

### Puntos a desarrollar:

1. Implementar un equipo goloso que dado un estado, mueva a sus jugadores de la mejor forma posible respecto a una función puntuadora de estados, para esto:
  - a) Implementar una función parametrizable que dado un estado cualquiera de un tablero, la misma le asigne un puntaje. La complejidad de esta función debe ser como máximo  $\mathcal{O}(N^2M^2)$ .
  - b) Implementar un equipo que utilice la función del punto 1a como comparador de los estados resultantes para los distintos movimientos posibles y que elija aquel movimiento que maximice dicho puntaje. **Nota:** Dado que en cada paso ambos equipos hacen movimientos a la misma vez se desconoce el movimiento del oponente, por lo que hay que tomar una decisión respecto a como modelarlo. Una opción es evaluar el tablero resultante para cada movimiento posible del oponente, y asumiendo que juega de forma óptima, quedarse con el peor de todos los puntajes obtenidos para ese movimiento.
2. Optimización de los parámetros:
  - a) Utilizar la técnica de *grid-search* de forma de conseguir buenos valores para los parámetros de la heurística propuesta en el punto 1. Dado que el espacio de búsqueda (tamaño de la grilla) es considerablemente grande, se pide implementar heurísticas de búsqueda local y grasp como mecanismos para recorrer dicha grilla.
  - b) Utilizar la técnica de algoritmos genéticos para optimizar los parámetros de la función desarrollada en 1. Para esto definir a los cromosomas como la parametrización anterior o una codificación de la misma e implementar lo siguiente:
    - 1) Una población (conjunto de cromosomas).
    - 2) Al menos dos funciones de *fitness*
    - 3) Al menos dos operaciones de *crossover*
    - 4) Al menos una operación de mutación
    - 5) Al menos dos métodos de selección distintos
3. Investigar el estado del arte de los algoritmos genéticos, principales áreas de aplicación y limitaciones. En base a lo investigado, escribir un resumen.
4. Utilizar los parámetros óptimos obtenidos en el punto 2 y dar un equipo (con los parámetros del algoritmo genético) que sepa jugar en un tablero de  $M = 10, N = 5, S = 250$ . El día 29 de Junio se llevará a cabo una competencia para la cuál cada grupo compite con el equipo provisto, con premios a determinar por la cátedra.

### Programa a desarrollar:

Cada equipo es representado por un programa que lee la configuración inicial y las jugadas de su rival por la entrada estándar y escribe sus propias jugadas en la salida estándar.

La cátedra provee un programa que hace de juez (Elizondo.py). Este programa debe ser provisto de dos binarios y se encarga de ejecutarlos, comunicarles los estados del tablero,

recibir sus jugadas y coordinar el partido, además, cuenta con la opción de mostrar mediante mediante archivos de logs y una interfaz gráfica el estado del juego. También provee algunos jugadores sin una estrategia inteligente al solo efecto de facilitar las pruebas.

La primera línea de la entrada estándar leída por un jugador en cada partido será la configuración inicial en forma de 3 enteros  $N$ ,  $M$  y  $S$  y un string *lado* separados por espacios.  $N$  es la cantidad de columnas,  $M$  es la cantidad de filas y  $S$  es la duración del partido en cantidad de movimientos, *lado* es uno de DERECHA o IZQUIERDA e indica de que lado juega el equipo. Luego le siguen 6 líneas con la información correspondiente a cada uno de los 3 jugadores del equipo y de los 3 jugadores rivales (en ese orden), en cada línea hay 2 números *id*  $p_{quite}$ , donde *id* es el id de jugador y  $p_{quite}$  es la probabilidad de quite correspondiente al jugador, dicha probabilidad es la utilizada en las fórmulas 1 y 2.

Luego el programa debe enviar la posición inicial de sus jugadores como 3 triplas (una por línea) con el siguiente formato:

id\_jugador fila columna:

1. id\_jugador es el id del jugador.
2. fila  $\in [0, M)$  es la fila en la cual se encuentra el jugador.
3. columna  $\in [0, N)$  es la columna en la cual se encuentra el jugador.

**Nota:** Si al equipo le toca comenzar, el programa va a colocar al jugador con id 0 en la posición central del tablero con la posesión de la pelota.

A partir de ahí, cada línea leída de la entrada estándar será un estado del tablero y cada línea escrita en la salida estándar será un conjunto de movimientos que representan como se mueven los jugadores del equipo. Cuando el partido termine, el juez enviará algunos de los mensajes *PERDISTE*, *GANASTE* o *EMPATARON* indicando que el partido terminó y el resultado del mismo.

El estado del tablero está formado por 6 cuádruplas (una por línea) que representan los las posiciones de los jugadores y si tienen o no la pelota. Cada cuádrupla está conformada de la siguiente manera:

id\_jugador fila columna posesión:

1. id\_jugador es el id del jugador.
  2. fila  $\in [0, M)$  es la fila en la cual se encuentra el jugador.
  3. columna  $\in [0, N)$  es la columna en la cual se encuentra el jugador.
  4. posesión  $\in [\text{CON\_PELOTA}, \text{SIN\_PELOTA}]$  indica si el jugador tiene o no la pelota.
- Nota:** a lo sumo un solo jugador puede tener la pelota.

**Nota:** Siempre los primeros 3 jugadores son los del equipo del programa y los siguientes 3 son los del oponentes. El programa le envía a cada equipo los jugadores en el orden correcto.

**Nota:** Si ningún jugador tiene la pelota, entonces hay una línea más que contiene 2 números indicando la posición de la pelota libre en el tablero.

El equipo envía los movimientos a realizar por sus jugadores como 3 triplas que representan los movimientos de cada jugador, cada tripla está conformada de la siguiente forma:

id\_jugador tipo\_movimiento valor\_movimiento

- `tipo_movimiento`  $\in$  [MOVIMIENTO, PASE]
- `valor_movimiento`:
  - Si `tipo_movimiento` = MOVIMIENTO entonces el valor es un número entre 0 y 8 inclusive como muestra la Figura 1
  - Si `tipo_movimiento` = PASE entonces el valor son 2 números, el primero es la dirección (entre 1 y 8 ya que 0 no es válido) y el siguiente es la fuerza del pase en cantidad de movimientos, el mismo puede ser un número positivo cualquiera siempre que la pelota no termine fuera del tablero.

**Nota:** Tener cuidado con los mensajes ya que los mismos son *case-sensitive*

### Ejemplos de ejecución del programa:

Los parámetros del programa Elizondo.py están explicados utilizando el flag `--help`.

Ambos equipos son el mismo ejecutable y las probabilidad de quite de todos los jugadores es 0.5:

```
python Elizondo.py --team_A ./random_player --quites_A 0.5 0.5 0.5
--team_B ./random_player --quites_B 0.5 0.5 0.5
--columns 10 --rows 5 --steps 50 --first A
```

Equipo A es el provisto por la cátedra y el equipo B un jugador ejecutable con distintas probabilidades de quite:

```
python Elizondo.py --team_B ./random_player --quites_B 0.5 0.5 0.5
--columns 10 --rows 5 --steps 50 --first A
```

### Ejemplo de input/output:

A continuación se muestra un ejemplo de una entrada y su posible salida. Dado que la entrada y salida corresponde a un protocolo de comunicación entre el juez y un jugador, los mismos se muestran de forma intercalada para expresar la temporalidad en que cada proceso envía sus mensajes.

Entrada de ejemplo	Posible salida esperada de ejemplo
10 5 50 DERECHA 0 0.5 1 0.5 2 0.5 0 0.5 1 0.5 2 0.5	0 1 1 1 2 1 2 3 1
0 2 4 CON_PELOTA 1 2 1 SIN_PELOTA 2 3 1 SIN_PELOTA 0 1 9 SIN_PELOTA 1 2 9 SIN_PELOTA 2 3 9 SIN_PELOTA	0 PASE 1 3 1 MOVIMIENTO 3 2 MOVIMIENTO 4
0 2 4 SIN_PELOTA 1 1 2 SIN_PELOTA 2 3 2 SIN_PELOTA 0 1 8 SIN_PELOTA 1 2 8 SIN_PELOTA 2 3 8 SIN_PELOTA 4 6 . . .	. . .
GANASTE	



### Fechas de entrega

- Formato Electrónico: Jueves 28 de Junio de 2018, hasta las **23:59 hs**, enviando el trabajo (informe + código) a la dirección algo3.dc@gmail.com. El subject del email debe comenzar con el texto [TP3] la lista de apellidos de los alumnos.
- Formato físico: Viernes 29 de Junio de 2018, a las 18 hs. en la clase de laboratorio.

**Importante:** El horario es estricto. No se considerarán los correos recibidos después de la hora indicada.