

# Emulador de resultados del fixture del Mundial Qatar 2022

Estamos ante un mundial súper mega importante, ya que sería el primero que no tenemos a Maradona con nosotros, y el último de Messi... Por lo tanto, vamos a esforzarnos un poco en emular los resultados de este mundial con un poquito de suerte (el azar) y puede haber un jueguito con las probabilidades (un poco de estadística).

Empecemos...

En un campeonato de fútbol, lo primero que tenemos es la fase de grupos, que se ve de la siguiente forma:





GRUPO "A"		
FECHA 1	FECHA 2	FECHA 3
<b>Domingo 20-11 / 13:00 (ARG)</b>  <b>VS</b>  QATAR ECUADOR ESTADIO AL BAYT	<b>Viernes 25-11 / 10:00 (ARG)</b>  <b>VS</b>  QATAR SENEGAL ESTADIO AL THUMAMA	<b>Martes 29-11 / 12:00 (ARG)</b>  <b>VS</b>  QATAR PAÍSES BAJOS ESTADIO AL BAYT
<b>Lunes 21-11 / 13:00 (ARG)</b>  <b>VS</b>  SENEGAL PAÍSES BAJOS ESTADIO AL THUMAMA	<b>Viernes 25-11 / 13:00 (ARG)</b>  <b>VS</b>  ECUADOR PAÍSES BAJOS ESTADIO INTERNACIONAL KHALIFA	<b>Martes 29-11 / 12:00 (ARG)</b>  <b>VS</b>  ECUADOR SENEGAL ESTADIO INTERNACIONAL KHALIFA

- Tenemos una cantidad de grupos limitada, del grupo "A" al grupo "H"
- La cantidad de equipos por grupo son 4
- La cantidad de partidos por grupo son 6

Luego viene la situación de las estadísticas que definen quienes pasan y quienes no de cada grupo.

- Solo pasan los 2 primeros

Estadísticas de cada Equipo:

Group A								
Team	MP	W	D	L	GF	GA	GD	Pts
1  Qatar	0	0	0	0	0	0	0	0
2  Ecuador	0	0	0	0	0	0	0	0
3  Senegal	0	0	0	0	0	0	0	0
4  Netherlands	0	0	0	0	0	0	0	0

Cada equipo posee la siguiente estadística:

- **MP:** Matches played (partidos jugados)
- **W:** win (ganados)
- **D:** draw (empate)
- **L:** loss (perdidos)
- **GF:** goles a favor
- **GA :** goal against (goles en contra)
- **GD:** goal difference (diferencia de goles)
- **Pts:** puntos totales

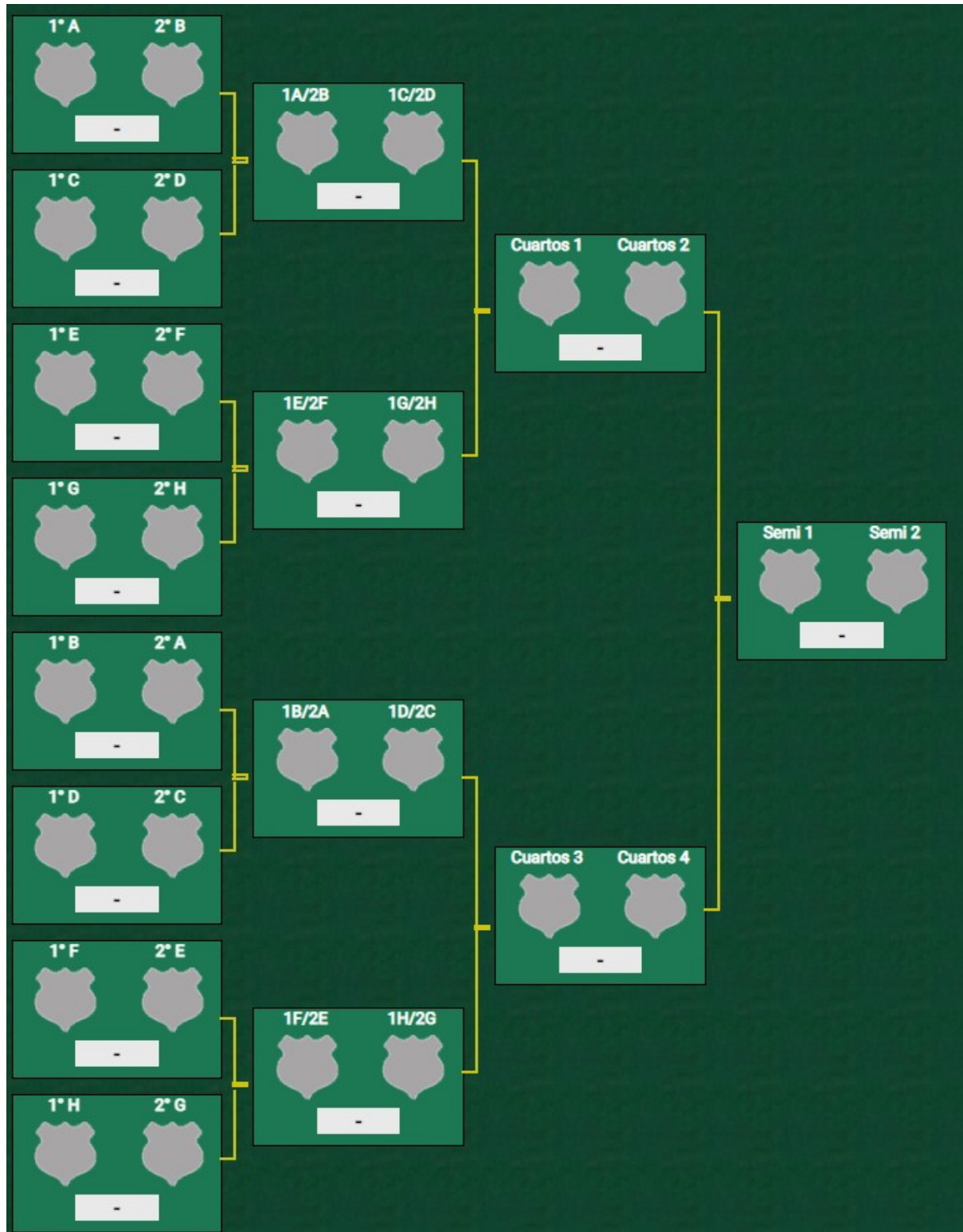
...Y ahora viene la parte compleja... ¿Quiénes pasan a la siguiente ronda?

- Pasan los 2 de cada grupo que tengan todos los partidos jugados y posea más puntaje.
- El puntaje total se calcula de la siguiente forma:
  - Por cada partido: **3pts si se gana, 1 si empata, 0 si pierde**
- Ahora, puede existir que empaten en puntos... y así sucesivamente.
  - Consideremos solo 3 situaciones, pasaran en el siguiente orden:
  - **1.** Por puntos
  - **2.** Si empatan en puntos, la diferencia de goles.
  - **3.** Si empatan, por goles realizados.

## Siguientes rondas

Para el resto de las llaves, se desglosa de esta manera:

**8vos de final** → **4tos de final** → **semifinales** → **final**



- Tener en cuenta que el orden en que van los equipos se debe respetar según quienes han salido 1ero o 2dos en la fase de grupos.

# Codificación

1. Cada equipo tendrá la siguiente estructura:

```
typedef struct {
    char nomEquipo[40];
    int mp; ///partidos jugados
    int gf; ///goles a favor
    int ga; ///goles en contra
    int win; ///partidos ganados
    int loss; ///partidos perdidos\
    int pts; ///puntos totales
}Equipo;
```

Primeramente, tenemos una lista simplemente enlazada de Equipos, donde los equipos los cargaremos con sus nombres y el resto de los datos en 0. Las referencias a los equipos una vez creados se trasladaran al resto de las estructuras donde se necesiten, es decir, sólo sus referencias. Entonces, en cualquier lado que se modifiquen sus campos se verá reflejado en todo el programa. No volveremos a crear un nuevo Equipo, siempre nos manejaremos con su referencia.

```
typedef struct
{
    Equipo equipo;
    struct nodoEquipo* siguiente;
}nodoEquipo;
```

2. Los grupos deben estar representados mediante un **Arreglo de listas simplemente enlazadas de tipo Equipo**, donde cada celda del arreglo es de tipo Grupo

```
typedef struct
{
    char letra; ///'A','B'....
    struct nodoGrupoEquipo* equipos; ///siempre va a haber 4
}Grupo;
```

```
typedef struct
{
    Equipo* equipo;
    struct nodoGrupoEquipo* siguiente;
}nodoGrupoEquipo;
```

3. Los partidos se representan mediante la siguiente estructura:

```
typedef struct
{
    int id; ///opcional
    int fecha; ///eligen ustedes el criterio, el tipo tambien
    Equipo* equipo1;
    Equipo* equipo2;
    int golesEq1;
    int golesEq2;
    int penales1; ///en la fase de grupos no hay penales
    int penales2; ///pueden setearlo en 1 y 0 (como un boolean)
}Partido;
```

Donde equipo1 como equipo2 son referencias a Equipo de la lista de equipos.

**¿Qué hago cuando debo cargar los partidos antes de que se jueguen?**

Los partidos pueden traerlos de archivo en alguna estructura distinta (con sólo la fecha, la fase y el grupo(de cada equipo, 1G, 2H..) al que pertenecen.

4. **¿Cómo se representan los partidos?**

Los partidos de la fase de grupos es el más especial ya que será un arreglo de listas de partidos único.

```
typedef struct
{
    Partido partido;
    struct nodoPartido* siguiente;
}nodoPartido;
```

```
typedef struct
{
    char letra;
    nodoPartido* partidos;
}GrupoPartido;
```

Luego, octavos de final, como 4tos, la semifinal, tercer puesto y final, estarán contenidos en un *arreglo de fases* de la siguiente forma:

- Fases[0]///**octavos**  
Contiene una lista de 8 partidos
  - Fases[1]///**4tos**  
Contiene una lista de 4 partidos
  - Fases[2]///**semifinales**  
Contiene una lista de 2 partidos
  - Fases[3]///**tercer puesto**
-

- Contiene una lista de 1 partido
- Fases[4]///final
- Contiene una lista de un solo partido

```
typedef struct
{
    int idFase;
    nodoPartido* partidos;
}fase;
```

### OBJETIVO

- El objetivo final de nuestro emulador es que el usuario ingresando (o no) un resultado (“clasifica” o “no clasifica”) **en fase de grupos** para Argentina o para cualquier país o países, se deduzca el resto de los resultados.
  - ¿Cómo lo hacemos? puede ser todo a través del azar, o a través de las probabilidades: <https://oddspedia.com/football/world/world-cup>.  
<https://predict.fotmob.com/en/wc2022>
  - Ustedes elijan la forma en que más les gustaría resolverlo. Pueden generar una estructura “Probability” que tenga la probabilidad de ganar de un equipo frente a otro.
  - Se debe poder visualizar el resultado de un partido o de una fase total.
  - La muestra final de cada uno de los resultados debe visualizarse por pantalla (como un arbolito), decidan cómo logran el gráfico.

Entrega final: Lunes 28 de noviembre (ojo porque no tiene reentrega, aconsejo entregar el 25)

### **Reglas del TP**

1. El código debe estar correctamente modularizado por cada estructura de datos.
2. Las variables como las funciones deben poseer nombres coherentes a su objetivo.
3. Los tipos de datos definidos por el usuario deben respetar la estructura mencionada.
4. Al momento de la entrega, el programa no puede poseer errores que impidan su ejecución. (esto se puede reever)

5. La exposición es grupal y todos los integrantes del grupo deben participar, la nota es un promedio entre la nota individual del TP y la grupal.

Condiciones	Puntos	Obtenido
<ul style="list-style-type: none"> <li>• Correcta modularización</li> <li>• Utilización de librerías</li> <li>• Nombres significativos en funciones y variables utilizadas</li> <li>• Compilación del proyecto sin errores</li> </ul>	30	
<ul style="list-style-type: none"> <li>• Creación de la s estructura s compuestas solicitadas y la carga.</li> </ul>	35	
<ul style="list-style-type: none"> <li>• Correcta resolución del emulador</li> </ul>	35	

**Tabla de puntuación:**

Puntaje	10	20	30	40	50	60	70	80	90	100
Nota	1	2	3	4	5	6	7	8	9	10
Condición	Desaprobado					Aprobado				