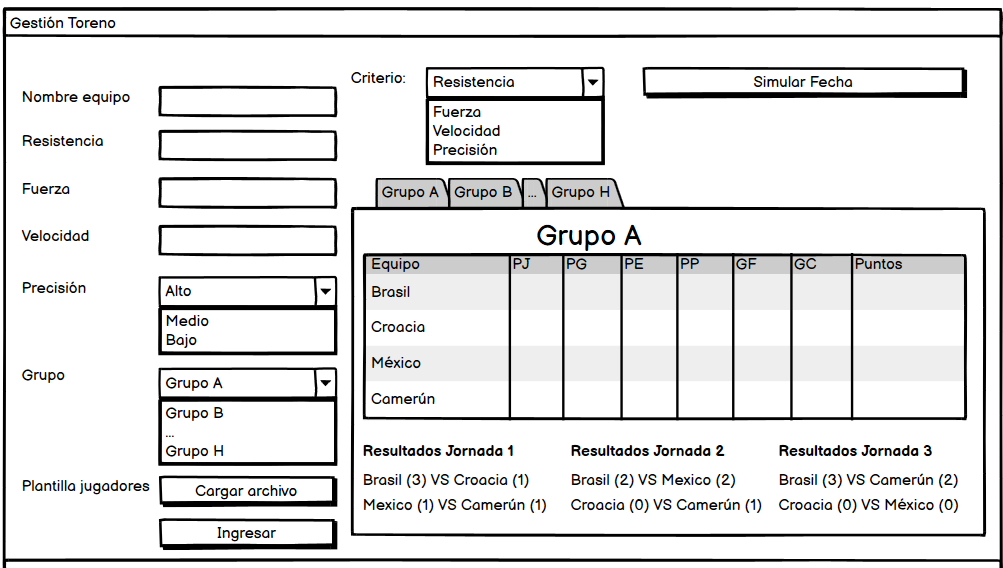
# ENFERMOS POR EL FUTBOL

Se le pide desarrollar un proyecto de software en el cual se puedan simular los partidos, de manera que se pronostiquen resultados y se proyecte como va a ser toda la copa. Para tal tarea usted debe recolectar información de cada equipo, en qué grupo quedó y las características principales que permitan la simulación, como resistencia, fuerza, velocidad, precisión en diferentes terrenos discriminados como altos, medios y bajos. Tenga en cuenta que no es el mismo desempeño de un equipo que acostumbra a obtener buenos resultados en terrenos bajos que los resultados que obtendría jugando en un terreno alto.

**Primera fase del torneo**

En primer lugar, se debe construir una interfaz gráfica la cual permita ingresar los datos de los equipos (descritos anteriormente) y poder asignarlos a los grupos correspondientes en los que jugaran la primera ronda del torneo. Es de aclarar que la máxima cantidad de equipos por grupo es de 4, en caso de querer violar esta regla se debe mostrar una alerta correspondiente para que el usuario elija otro grupo.



**NOTA: La interfaz anteriormente mostrada sirve como referencia, pero pueden cambiar el diseño y componentes gráficos a utilizar.**

Como referencia para su programa, la figura 1, (por cortesía de Johnson & Johnson) ilustra la configuración de los grupos.



Cada vez que se ejecuta el botón “Jugar Fecha” se toma como referencia el “criterio de juego”, una lista desplegable que permite elegir entre las opciones de resistencia, fuerza, velocidad y precisión. Según sea el caso se aplican las siguientes reglas para asignar el resultado de un partido, basta en principio con solo analizar la primera consideración, si hay un empate se procede a mirar la segunda, y si persiste el empate el partido queda empatado.

|  |  |  |
| --- | --- | --- |
| **Criterio** | **Primera consideración** | **Segunda Consideración** |
| Resistencia | Gana el equipo con mayor resistencia | Gana el equipo con mayor fuerza |
| Fuerza | Gana equipo con mayor fuerza | Gana equipo con mayor velocidad |
| Velocidad | Gana equipo con mayor velocidad | Gana equipo con precisión más alta |
| Precisión | Gana equipo con mayor precisión | Gana equipo con mayor resistencia |

**NOTAS:**

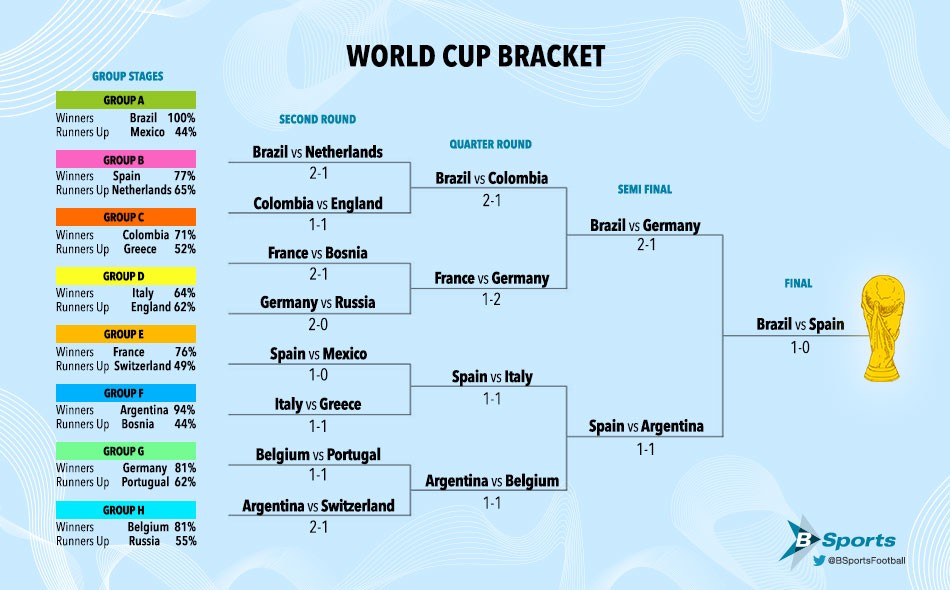
1. Cuando existe un empate, el número de goles es un valor aleatorio entre 0 y 3.
2. Cuando existe un ganador, el número de goles (tanto para el local como el visitante) se generan de manera aleatoria entre un rango de 1 y 5. Donde el que gana por obvias razones tiene un número mayor al que pierde.
3. Cada vez que se juega una fecha, debe aparecer los equipos enfrentados, con el marcador obtenido, y la tabla de posiciones se debe actualizar automáticamente.

Una vez finaliza la tercera fecha, se genera un árbol binario que representa la fase desde dieciséis octavos de final, donde los clasificados corresponden a los 2 primeros de cada grupo.

**Segunda fase del torneo**

La figura 2, para su referencia, muestra la manera en cómo debe conformarse el árbol de encuentros en el que a partir de los grupos se realizará la simulación indicando el pronóstico de la copa, a partir de los partidos inicialmente programados, de todos los resultados hasta llegar al partido final y mostrar cual es el campeón siguiendo los criterios anteriormente mencionados para seleccionar el ganador del encuentro.

Se debe poder visualizar en otra interfaz el comportamiento del árbol como va quedando fecha tras fecha.



# Requerimientos:

1. En cualquier momento puede ingresar a la simulación un evento extraordinario, como por ejemplo, la descalificación inmediata de un equipo en el torneo, que puede suceder aun si éste es ganador
2. Su programa debe ofrecer la posibilidad de mostrar todas las simulaciones posibles con base en la permutación de todos los equipos programados en los primeros encuentros de la copa.
3. Se debe graficar estadísticamente el desempeño de cada equipo en cada una de las simulaciones. La estadística corresponde a un diagrama de barras que representa las ganancias obtenidas por pasar a cada fase. (Los valores son acumulables cada vez que se pasa de ronda.

|  |  |
| --- | --- |
| **Fase** | **Valor (USD)** |
| Dieciseisavos de final | 20.000 |
| Cuartos de final | 25.000 |
| Semifinal | 35.000 |
| Final | 40.000 |
| Campeón | 40.000 |

1. Cada equipo se debe identificar por la bandera del país respectivo, el director técnico y los jugadores, donde la ficha técnica de un jugador consta de: nombre, edad (en años), altura (en metros), peso (en kilogramos), salario (en millones de dólares por año), posición en la que juega (delantero, centrocampista, defensa o arquero) y número de goles de la temporada. Estos datos los pueden almacenar en un archivo texto con el fin de agilizar la carga inicial de los datos por parte del usuario, y, tenga en cuenta que estos datos pueden ser ficticios. La estructura del archivo por cada equipo será en formato CSV separados por “;”

**Métodos a implementar**

1. Resalte en color amarillo la ruta que siguió un equipo seleccionado por medio de una lista desplegable (en la segunda fase del torneo)
2. Determine la cantidad de partidos en el que un equipo seleccionado por parámetro ganó por diferencia de 2 goles en la segunda fase
3. Pinte en color rojo por cada una de las llaves del torneo (pares de equipos), el equipo con menos promedio de edad
4. Dado un nivel del árbol por parámetro, se requiere marcar en color azul el equipo con menor cantidad de goles anotados durante todo el torneo (suma de goles de la primera fase + segunda fase)
5. Calcule la cantidad total de goles anotados en la segunda fase
6. Brinde una bonificación de 10.000 por cada partido que ganó el campeón del torneo cuando jugó de visitante.
7. Obtenga el equipo con mayor promedio de altura de todos los equipos que jugaron la segunda fase.
8. Obtenga el equipo que tiene el jugador con mayor salario de todos los equipos que jugaron la segunda fase y que quedaron en segunda posición en la primera fase.
9. La visualización del árbol se puede dar de 2 maneras horizontal o vertical.

**Consideraciones**

* Grupos máximo de 2 personas
* La implementación debe ser totalmente gráfica utilizando el lenguaje de programación Python.
* Entrega y sustentación 30 de abril de 2024