En nuestra red bayesiana podemos identificar tres grupos de nodos:

- Nodos de partidos:
    - **Identificador**: round1_1v2 es el partido de la primera ronda entre el equipo 1 y 2. round3_2v3 es el partido de la tercera ronda entre el equipo 2 y el 3…
    - **Arcos**: los nodos de partidos siempre tienen dos arcos de salida que van a los respectivos nodos de memoria de la ronda correspondiente y tienen 2 arcos de entrada que vienen de los nodos de memoria de la anterior ronda. La única excepción a esto es la primera ronda que no tienen nodos de entrada.
    - **CPTs**: Las tablas de probabilidad de los nodos las podemos dividir en dos subgrupos:
        - Primera ronda: hemos considerado las probabilidades de victoria de los equipos a ser equiprobables, aunque si se quisiera en un escenario dado se podría distribuir las probabilidades siguiendo el consejo de un grupo de expertos.
        - Subsecuentes rondas: a partir de la primera, las siguientes rondas alteran su probabilidad de victoria base (50%-50%) en función de las victorias que tengan los equipos en esta liguilla. Lo que se ha hecho ha sido por cada victoria de diferencia que tenga un equipo sobre otro, otorgar a dicho equipo una probabilidad de victoria de un 20% superior. Evidentemente al escalar esta red se tendría que ponderar el incremento a la probabilidad de victoria según el número de victorias que puedan darse. Nuevamente, las probabilidades base se modificarían acorde a lo que creyera un grupo de expertos.
- Nodos de “memoria” de victorias:
    - **Identificador**: team1_wins_round1 representa el número de victorias que tiene el equipo 1 después de la primera ronda. team4_wins_round3 representa el número de victorias que tiene el equipo 4 después de la tercera ronda.
    - **Arcos**: los nodos de “memoria” reciben como arcos de “entrada” los partidos de la ronda correspondiente y como salida tienen los partidos de la siguiente ronda y los nodos de memoria de la siguiente ronda.
    - **CPTs**: los nodos de “memoria” tienen CPTs muy simples ya que se usan solo para contar victorias. Ejemplo: Si el input de los nodos de memoria de la ronda anterior es 1 y entra una nueva victoria desde la ronda correspondiente, el output será 2.
- Nodos de recuento de resultados:
    - **Identificador**:
        - Nodos intermedios: Son nodos que se usan para evitar tener tablas exponencialmente grandes, por cada par de partidos debería haber un nodo de este tipo se llaman wins_compare1 para los nodos de “primer nivel”, si hubieran más “niveles” de nodos intermedios se llamarían wins_compare2, wins_compare3, etc…
        - Nodo de output: Es el nodo llamado winner que nos indica que equipo ha ganado la liguilla.
    - **Arcos**:

        - Nodos intermedios: Los nodos intermedios de primer nivel reciben los inputs de las células de memoria de última ronda de dos de los equipos. Como el output de los nodos intermedios puede ir o bien a un nodo intermedio de un nivel superior o al nodo Winner.
        - Nodo de output: Recibe de input los dos nodos intermedios de más alto nivel. No tiene ningún arco saliente.
    - **CPTs**:
        - Nodos intermedios:
            - Primer nivel: en el primer nivel los nodos intermedios reciben la información sobre el número de wins que han obtenido dos equipos. Como output sacan el equipo que tiene más victorias de los dos así como el número de victorias que este ha tenido. Si los dos equipos tienen un número de victorias que sabemos que no les va a dar la victoria de la liguilla directamente saca como output “nowinner”.
            - Siguientes niveles: en los siguientes niveles, se obtiene el input de dos nodos intermedios de un nivel inferior (o de un nodo intermedio y un número de victorias en el caso de que el número de equipos no sea potencia de 2). Dicho input consta de un equipo con su respectivo número de victorias, y como output saca el equipo de los dos que tenga más victorias. También puede sacar “nowinner” si no hay ninguno de los dos equipos que tenga posibilidad de ganar o si las dos entradas son “nowinner”.
        - Nodo de output: El nodo de output da como resultado las probabilidades que tienen los equipos de ganar la liguilla. Para calcular las probabilidades usa como input los dos nodos intermedios de más alto nivel y le da la victoria al que tenga más victorias de esos dos.

Funcionalidades añadidas:

- Se ha añadido la funcionalidad de dar las probabilidades de victoria según las victorias que haya tenido cada equipo a lo largo de la liguilla.
- Las probabilidades del ganador se actualizan al añadir evidencias.

Optimizaciones aplicadas:

La escalabilidad ha sido el principal eje alrededor del que se han tomado las decisiones de diseño que han llevado a la red resultante.

- Las variables de “memoria” de victorias se han dividido a una por equipo y ronda en vez de tener solo una por equipo para evitar tener tablas exponencialmente grandes al aumentar la n.
- Se ha añadido un sistema de nodos intermedias para el cómputo del ganador con el objetivo de evitar tener que recurrir a la opción fácil para evitar tener tablas exponencialmente grandes al aumentar la n. Para implementar este sistema de nodos intermedios nos ha llevado a aumentar el número de variables que éstos

tienen, aunque no supone un gran problema como explicamos en la sección de escalabilidad.
- Se ha añadido el concepto de “nowiner” para resumir los casos en los que ningún equipo de los comparados va a tener posibilidades de victoria. Con esto nos ahorramos $Celing\left(\frac{n-1}{2}\right) - 1$ variables.

Escalabilidad de la solución:

No hay ningún nodo que recibe inputs de un número de nodos que crezca en función de n. Esto nos ha permitido evitar tener tablas exponencialmente grandes.
Aunque, como hemos comentado en la sección anterior, hemos tenido que aumentar el número de variables de los nodos intermedios, aunque esto es mejor que la solución evidente exponencial. Este aumento tiene su mayor efecto en la variable winner, aunque en este caso el número de columnas a rellenar es descrito por la siguiente función:

$$\left(2 * Ceiling\left(\frac{n-1}{2}\right) + 1\right)^2$$

Dicha función siempre será mejor a la alternativa a nuestra solución:

$$(n-1)^n$$

Para escalar la estructura de la solución, habrán:
- $\frac{n*(n-1)}{2}$ nodos partido
- $n * (n-1)$ nodos de memoria
- $n - 1$ nodos de resultado(contando los intermedios) en el peor caso (n potencia de 2)

RB resultante: