

UNIVERSIDAD  
IBEROAMERICANA

TSU SOFTWARE

ESTRUCTURA DE DATOS

PROYECTO FINAL

Mtro. Alfonso Gregorio  
Rivero Duarte

Isaac Vega Telésforo

# Documentación Proyecto

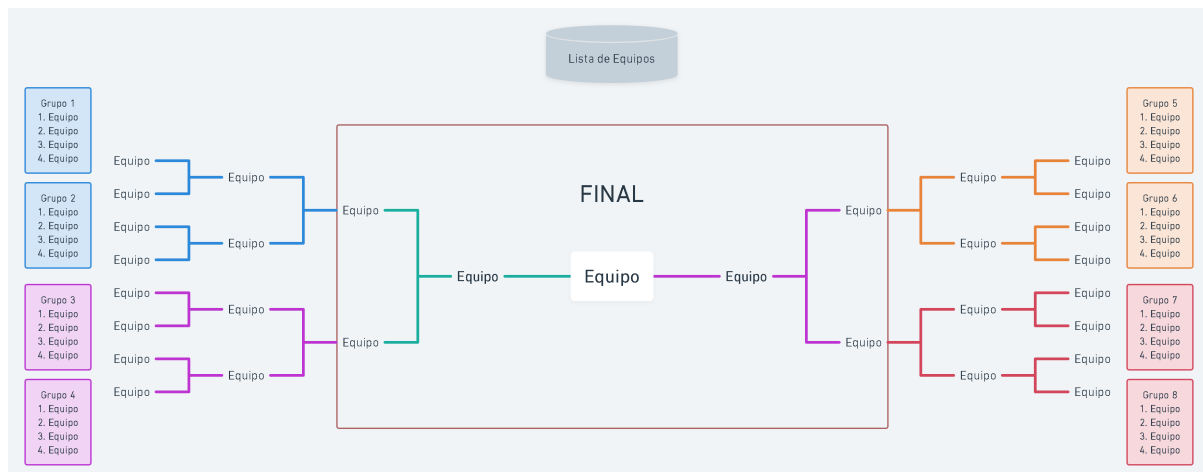
Universidad Iberoamericana - Estructura de datos - Isaac Vega Telésforo

## Problemática

Se quiere crear un programa que simule un torneo de football, implementando los temas vistos en clase de Estructura de Datos. Los temas que se deben implementar son listas, pilas, colas y/o árboles. Al igual que la lectura y/o escritura de archivos.

## Solución

Basandome en una breve investigación sobre como se realizan los juegos por la copa. Se crea una forma visual de poder entender el flujo que llevará el programa.



Como se puede observar, se cuentan con un total de 32 equipos, repartidos en 8 grupos, de los cuales, solo 2 equipos de cada grupo, se encuentran dentro del árbol general del juego.

Mi idea fue, convertir el archivo que contiene la lista de los nombre de los equipos, en una lista ligada, dentro de la cual, cada *bloque* se creará con la estructura en código de la derecha.

Ligando los *bloques* a modo de una **pila**.

Para los primeros datos solo utilizaré la posición, el nombre y los goles.

```
struct groups {
    char nombre[tam];
    int posGrp;
    int goles;

    // Pila, Cola
    groups *sig;

    // Árbol
    int posicion;
    groups *der;
    groups *izq;
    groups *padre;
};
```

```
struct grpClasificacion {
    int numGrupo;
    groups *g1;
    groups *g2;
    groups *g3;
    groups *g4;
    grpClasificacion *sig;
};
```

## Una ves creada la lista...

Para hacer que se vea más *interesante*, al crear los grupos, implementé una función que me devuelve un número aleatorio, busco ese número dentro de mi lista, y extraigo el “bloque” para insertarlo en la estructura en código de la izquierda.

Ligando los *bloques* a modo de una *cola*.

Una ves tengo todos los grupos creados, queda *jugar la clasificatoria*.

Para ello, bastará con colocar una cantidad  $n$  de goles a cada equipo; al terminar, una función de *ordenamiento*, mostrará los grupos con sus respectivos equipos, ordenados de mayor a menor puntaje. De los cuales solo los 2 primeros de cada grupo, son los que usaremos para el torneo.

## Construcción de los árboles...

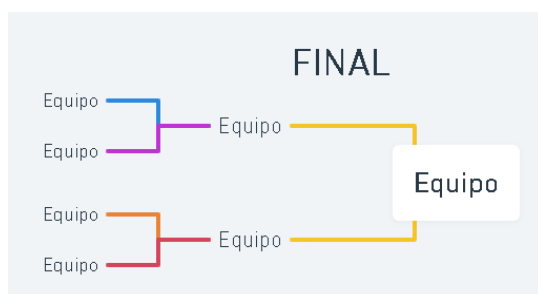
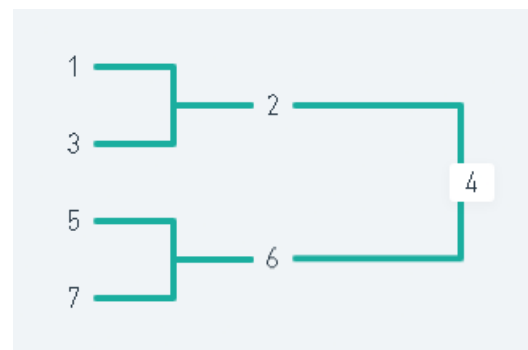
Como pudimos observar en la primer imagen, el torneo lo podemos crear como un solo árbol, o bien, lo que yo decidí, fue hacer 5 árboles pequeños.

Pero, ¿como voy a construir los árboles?

La forma más facil que se me ocurrió es hacer *árboles binarios*, ya que puedo colocar en ramas específicas los datos que yo quiera.

Como se observa en la imagen a la derecha, las posiciones por donde deberian comenzar los equipos, es en las *hojas*.

Pues bien, la manera que yo implemente fue crear una función que creara un árbol binario de manera automatica, pero para lograrlo, los números tambien deben de tener cierto orden.



Es por ello que cree un arreglo con en el orden adecuado para crear dicho árbol :

```
arregloPos[7] = { 4, 2, 1, 3, 6, 5, 7 }
```

De este modo, solo al iterar sobre una posición *hoja*, se guarda el nombre de los equipos.

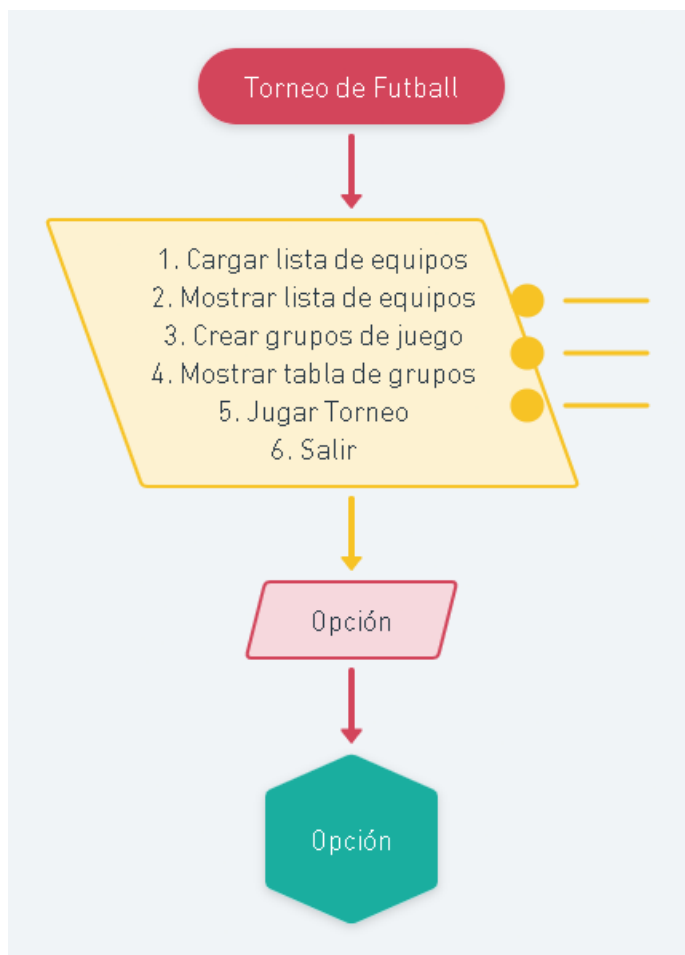
Al final me quedán 4 árboles binarios. Y tomando el resultado de cada uno de ellos, se construye el último árbol.

## ¿Como es que funciona el árbol del torneo?

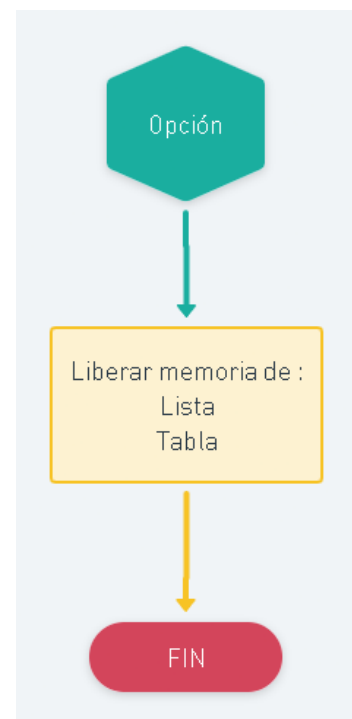
La manera más fácil que se me ocurrió, fue comparar los goles en cada partido, y copear, mediante el enlace *padre* al ganador. Una vez jugados todos los partidos, se termina el juego, libera memoria de los `árboles` y la `cola`, la lista, no se limpia, por si el jugador quiere usar esa misma lista para generar otro torneo; solo se libera la memoria de todo, al finalizar el programa.

Retomando el tema de las lista de los equipos, en caso de se quiera cambiar de lista se puede hacer sin ningún problema. De hecho, el programa ya cuenta con 3 listas con diferentes equipos, si ninguna lista agrada al jugador, se implemento la posibilidad de que se puedan crear listas personalizadas, introduciendo el comando clave `#new_` seguido del nombre de la lista que se creará, a su vez, se cuenta con la posibilidad de borrar alguna lista que ya no quiera usar, usando el comando clave `#delete_` seguido del nombre de la lista que se borrará.

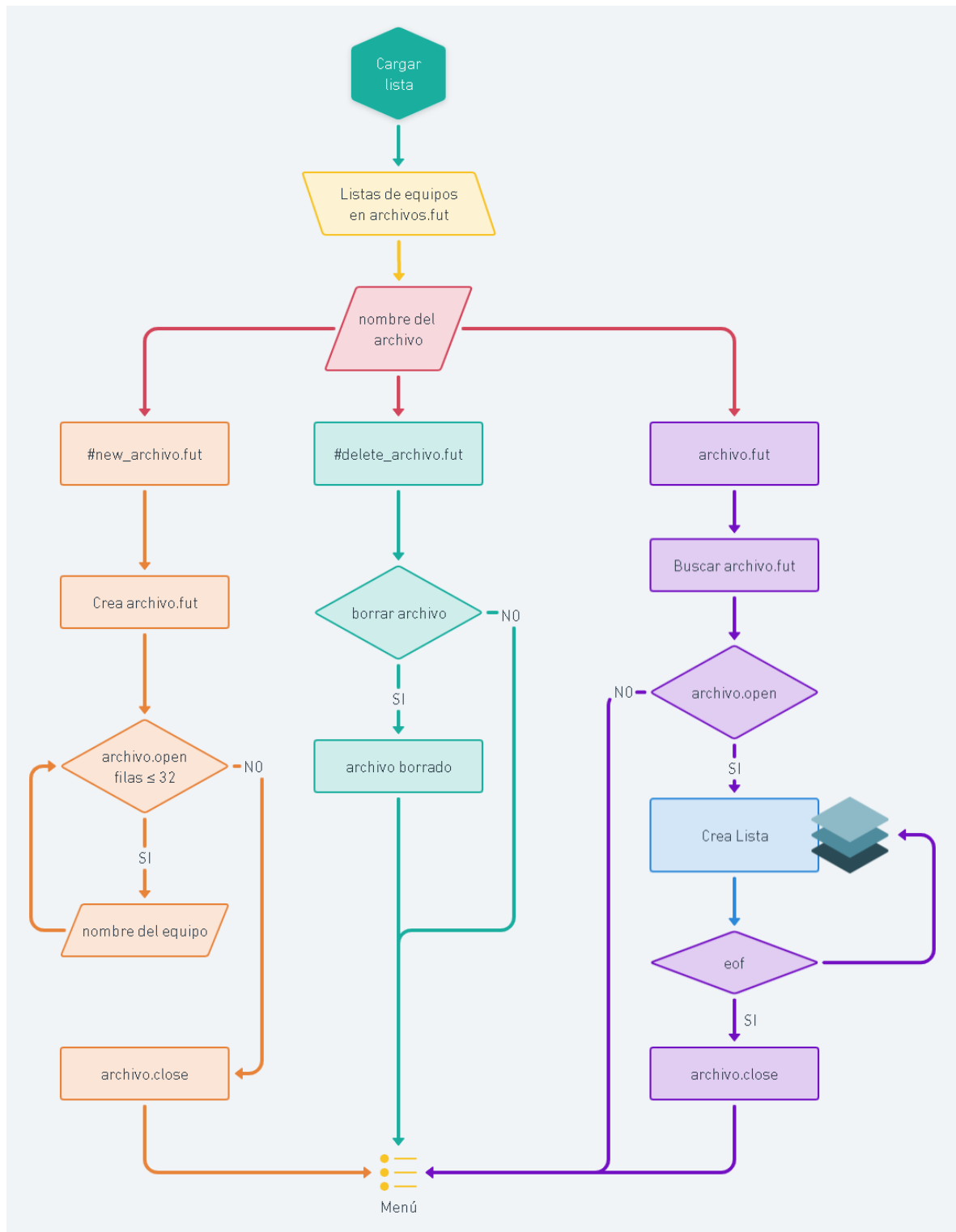
## Diagramas de Flujo



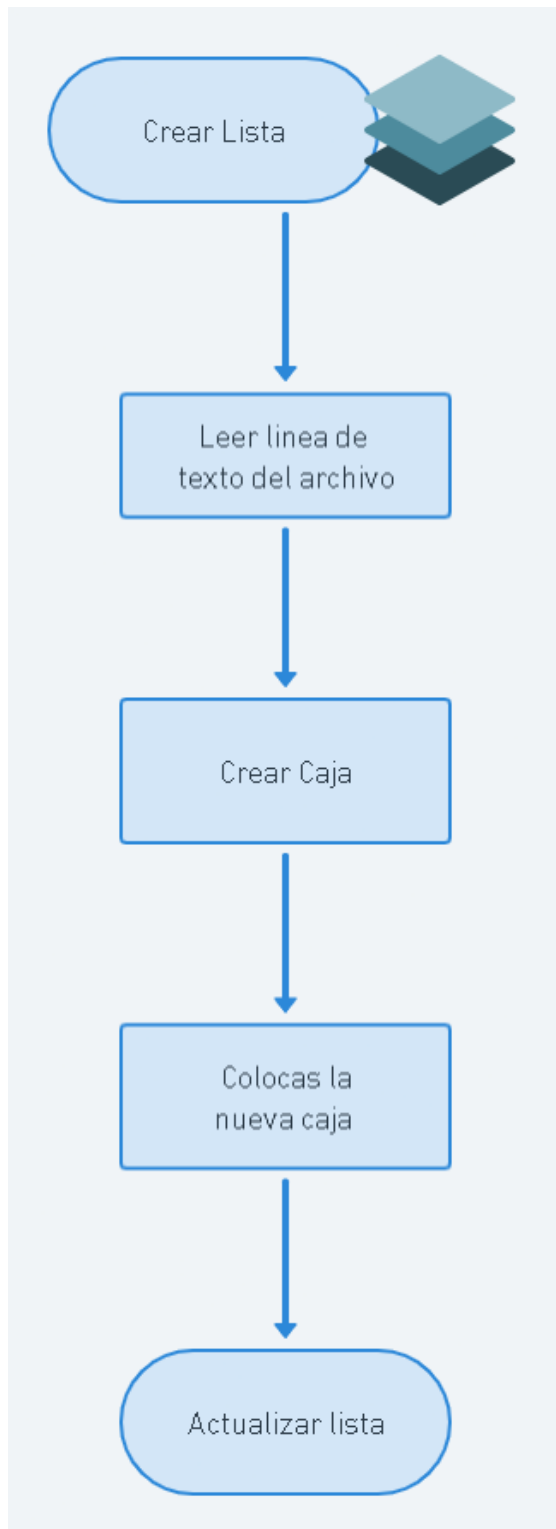
Menú



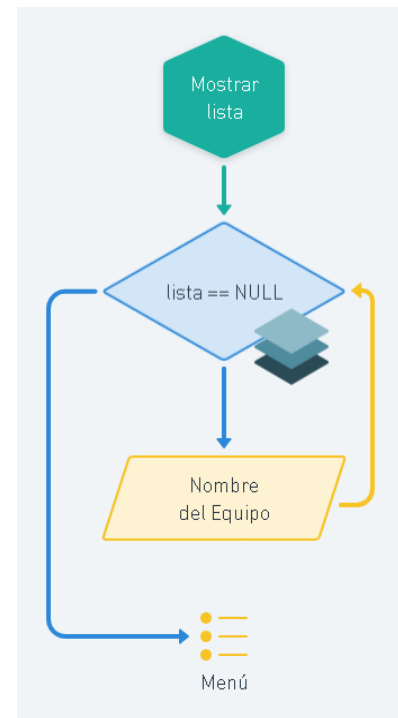
Opción salir



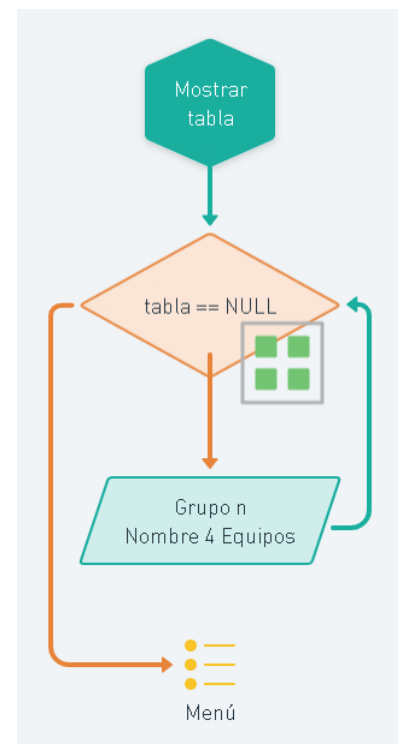
Cargar los archivos con la lista de los equipos



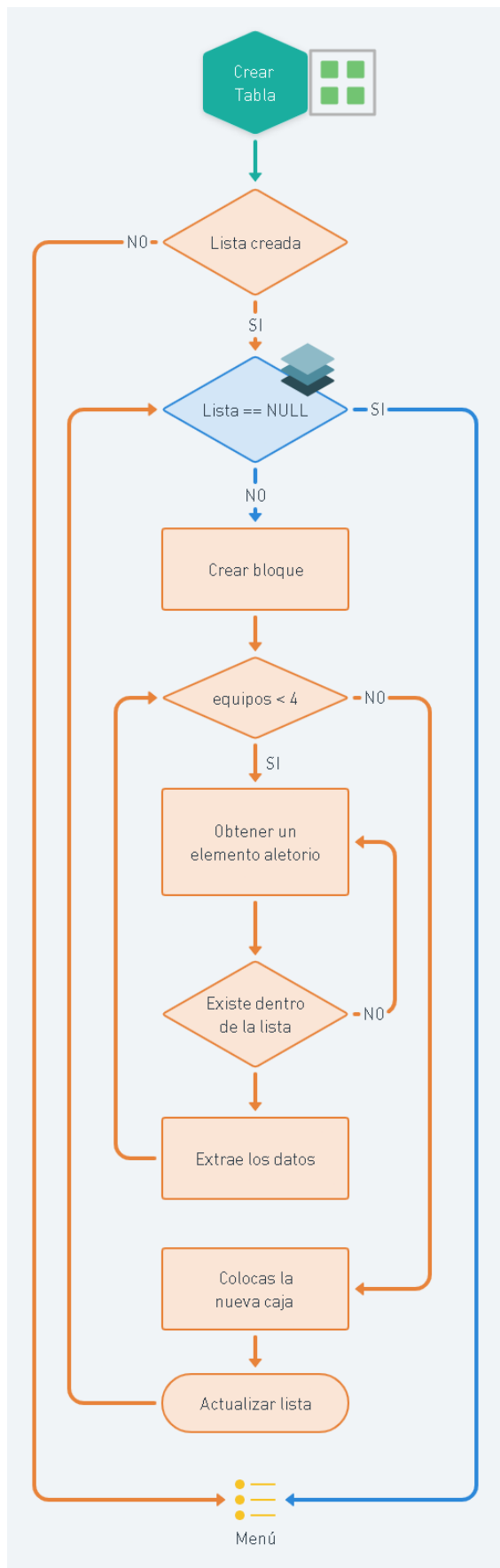
Crear la lista



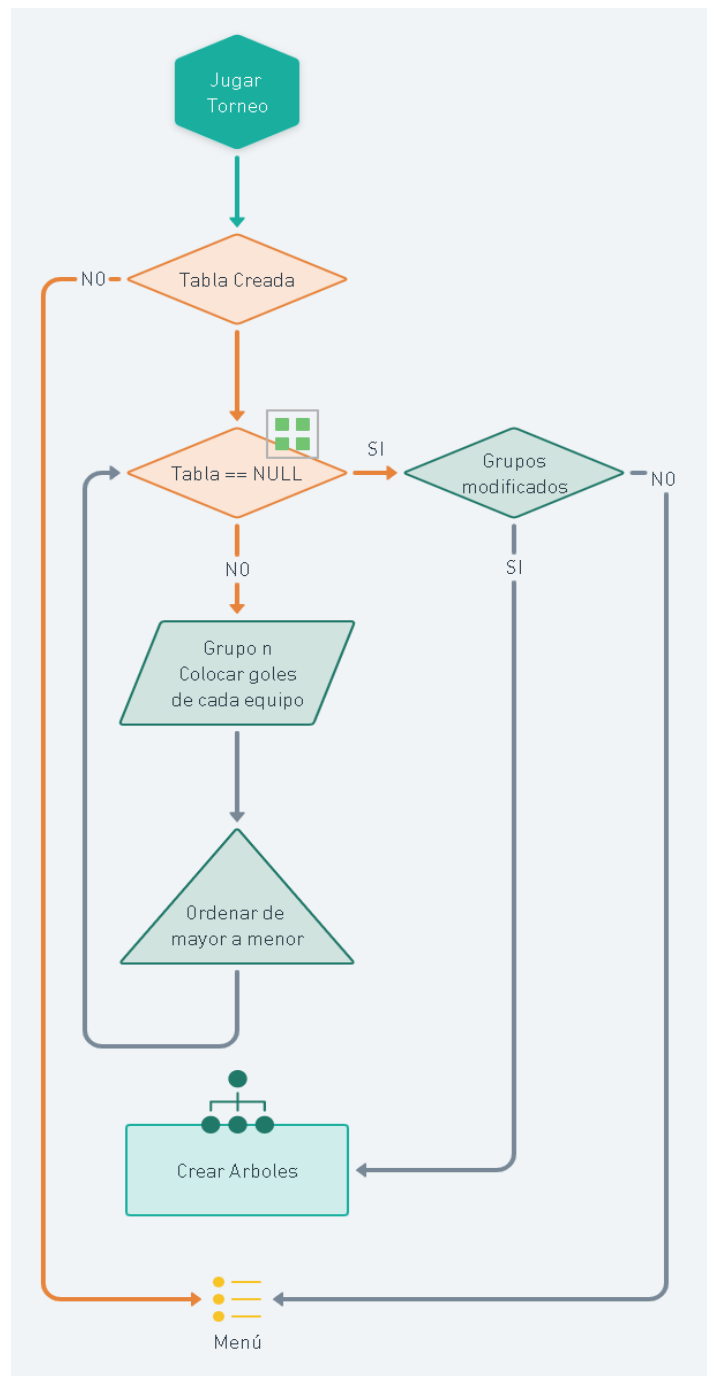
Visualizar la lista de los equipos



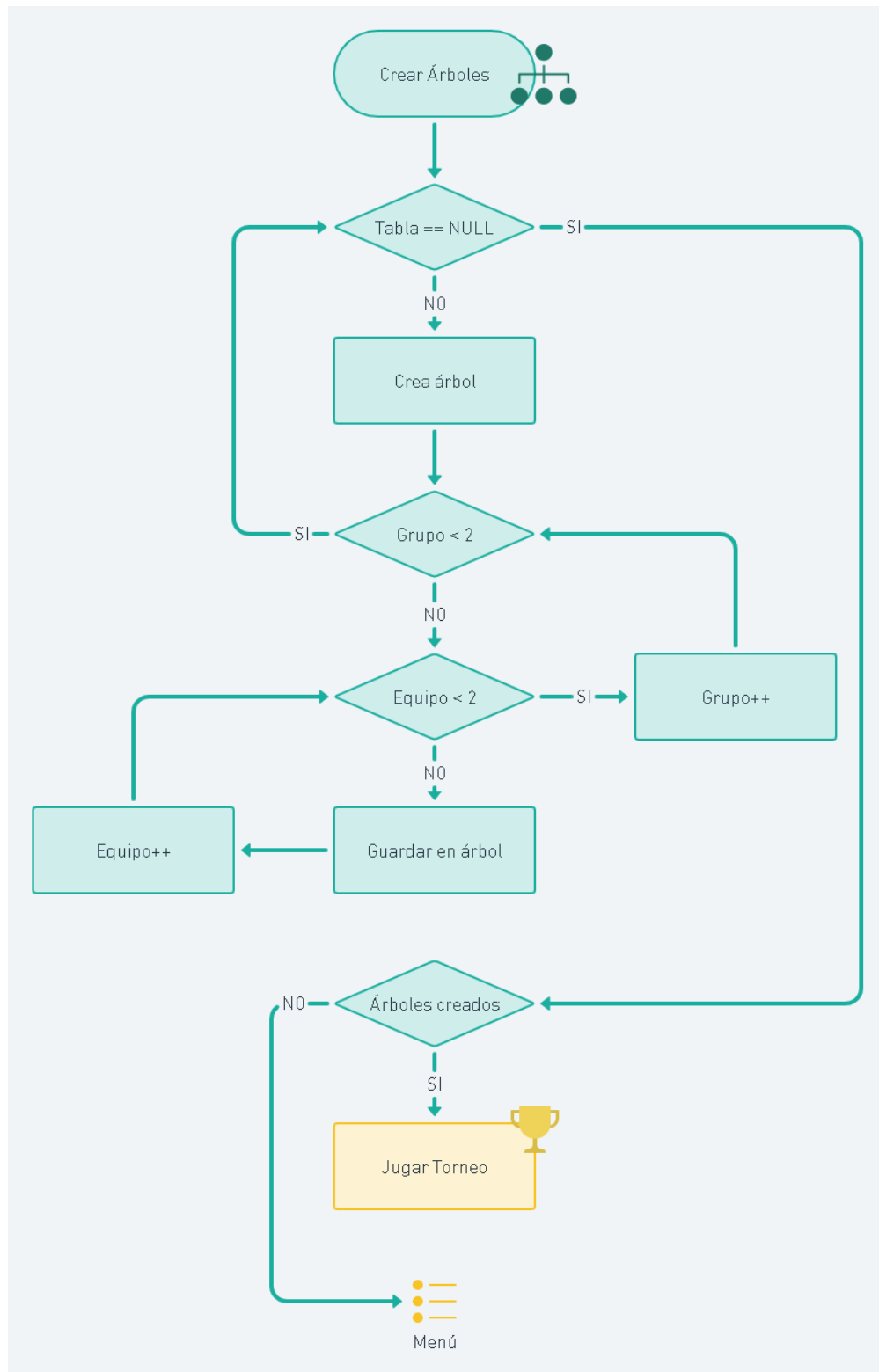
Mostrar los grupos creados



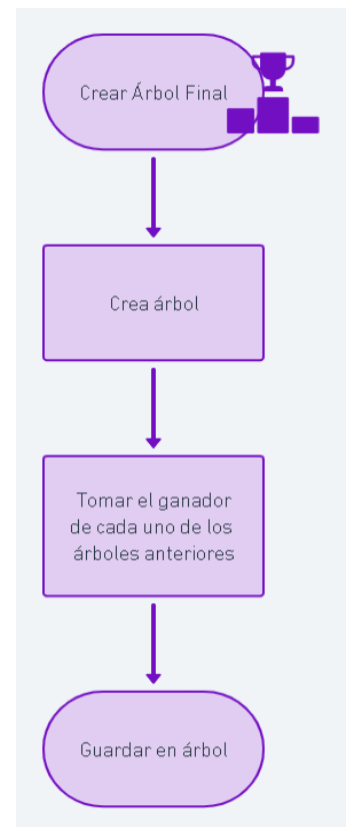
Crear los grupos con los equipos



Jugar cada uno de los grupos

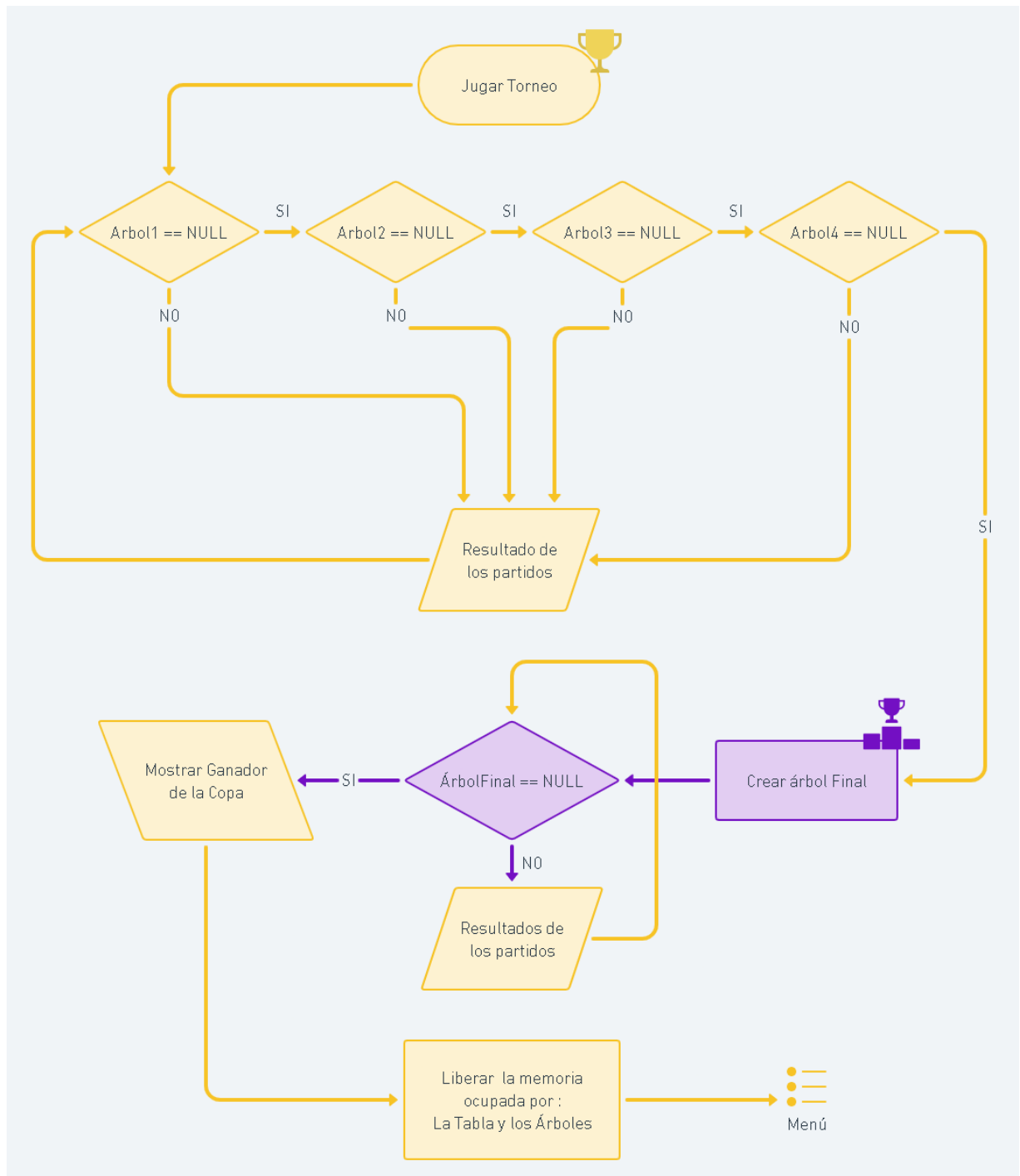


Crear cada uno de los árboles



Crear el ultimo árbol





Jugar todos los partidos hasta finalizar el juego