

## Ejercicio 3

Para este ejercicio se crearon las estructuras auxiliares `arbolPaisesGoles` y `nodoArbolEquipos`

Creamos un entero “`int lenPaises = 32`” que utilizaremos para contener el tamaño de un vector.

Creamos un vector del tipo “`arbolPaisesGoles vecPaisesGoles[lenPaises]`” para contener los nombres de los países junto con la cantidad de goles que convirtió cada país.

```
struct arbolPaisesGoles {
    char nombre_pais[20];
    int cantGoles;
};

struct nodoArbolEquipos
{
    nodoArbolEquipos* izq;
    arbolPaisesGoles info;
    nodoArbolEquipos* der;
};
```

## Resolucion del ejercicio 3. Funcionamiento General

1. `cargarDatosParaArbol(matriz, vecPaisesGoles);`

La función carga los datos necesarios en un vector el cual es luego utilizado para cargar los datos dentro de un árbol.

2. `nodoArbolEquipos * arbolEquipos = NULL;`

Creamos el árbol “`arbolEquipos`” que contendrá el listado de todos los equipos junto con la cantidad de goles que convirtió al finalizar el mundial.

3. `arbolEquipos = insertarPaisesYGolesEnArbol(vecPaisesGoles, lenPaises);`

La función inserta los valores que fueron cargados en el vector “`vecPaisesGoles`” en el árbol

4. `mostrarArbolEquipoYGoles(arbolEquipos);`

La función lee y muestra el contenido del árbol utilizando una función del tipo `inOrder`, llamada `inOrderGoles`.

Funcionamiento de los procedimientos

```
***** EQUIPOS ORDENADOS POR CANTIDAD GOLES EN ORDEN (MENOR A MAYOR) *****
AUSTRALIA: 2
COSTA RICA: 2
EGYPT: 2
ICELAND: 2
IRAN: 2
MOROCCO: 2
PANAMA: 2
PERU: 2
POLAND: 2
SAUDI ARABIA: 2
SERBIA: 2
MEXICO: 3
NIGERIA: 3
SOUTH KOREA: 3
SENEGAL: 4
DENMARK: 5
SWITZERLAND: 5
TUNISIA: 5
ARGENTINA: 6
COLOMBIA: 6
GERMANY: 6
```