

RECUPERATORIO PRIMER PARCIAL

Alumno: *Augusto Delgado*

División: *1°C*

Dni: *44 964 632*

Materia: *Laboratorio I*

2022

Enlace del video:

<https://drive.google.com/drive/folders/1OMyiPlkMi1-x8uOCQQKTG7QoNauk89Sy?usp=sharing>

Detalles de Las Estructuras

Agregadas

Agregue las Siguietes Estructuras

eClubDeFutbol

Esta nueva estructura contiene los datos de un club de futbol y se relaciona con la estructura **eFiguritas**.

Esta relacion representaría en que club juega el futbolista de la figurita. Esta relación seria de uno a muchos, ya que una figurita puede tener un solo club de futbol, y un club de futbol puede tener varias figuritas.

```
typedef struct
{
    int idDeClub;
    char descripcion[MAX_NOMBRE_CLUB];
    int cantidadDeTrofeos;
    int idDeLiga;
    int estado;
}eClubDeFutbol;
```

Esta nueva estructura está formada por un **id** es un campo único y que nos ayuda a identificar cada club. Una **Descripción** se trata del nombre del club, Una **Cantidad De trofeos** representa la cantidad de trofeos ganados por un club, un **estado** el cual determina si un club está cargado o no.

Listado Agregado: **Informar Porcentaje de Clubs por cada figurita pegada en el Album.** Se deberá mostrar las figuitas con su equipo y liga pegadas en el Álbum de cada club.

eLiga

Esta nueva estructura contiene los datos de una liga de futbol.

La cual está relacionada con la estructura **eClubDeFutbol**.

Esta relación representaría la liga donde juega un club

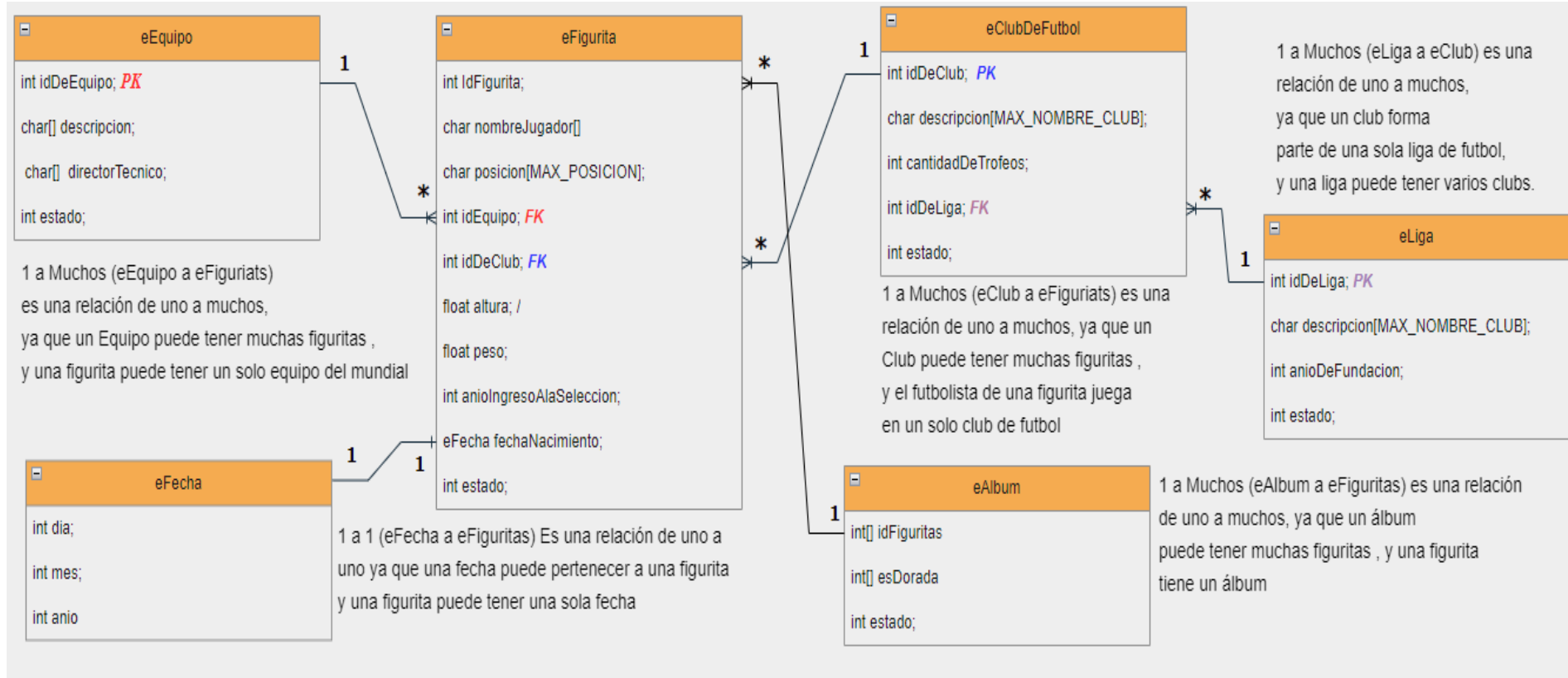
Esta es una relación seria de **uno a muchos**, ya que un club forma parte de una sola liga de futbol, y una liga puede tener varios clubs.

```
typedef struct
{
    int idDeLiga;
    char descripcion[MAX_LIGA_NOMBRE];
    int anioDefundacion;
    int estado;
}eLiga;
```

Esta nueva estructura está formada por un **Id** es un campo único y que nos ayuda a identificar cada liga. Una **Descripción** se trata del nombre de la liga, Un **anioDeFundacion** que representa el anio en el que se fundo la liga, un **estado** el cual determina si una liga está cargado o no.

Listado Agregado: **Determinar cuál/cuales son las ligas que mayor cantidad de figuritas pegadas en el Álbum.** Se deberá mostrar las figuritas con su equipo y liga pegadas en el Álbum de la/s liga mayor.

Diagrama Entidad Relación



Documentación de todas las funciones Nexo del programa

Nexo.h

```
/// @param int[] un array de figuritas
/// @param int el tamaño del array de Figuritas
/// @param eFiguritas[] el array de eFiguritas
/// @param int el tamaño del array de eFiguritas
/// @param eEquipo[] el array de eEquipo
/// @param int el tamaño del array de eEquipo
/// @param eAlbum* el puntero al album
/// @param eColeccionista* el puntero al coleccionista
/// @param eLiga[] el array de ligas
/// @param int el tamaño del array de ligas
/// @param eClubDeFutbol[] el array de Club de futbol
/// @param int el tamaño del array de Club de futbol
/// @param eAuxiliar[] un array auxiliar
/// @param sizeAuxiliar el tamaño del array auxiliar

/// @fn int SubMenuAdministrador(eFiguritas[], int, eEquipo[], int)
/// @brief Muestra el submenú De coleccionista y le permite al usuario ingresar una opción
///
```

```

/// @param eFiguritas[] el array de eFiguritas
/// @param int el tamaño del array de eFiguritas
/// @param eEquipo[] el array de eEquipo
/// @param int el tamaño del array de eEquipo
int SubmenuColeccionista(eFiguritas listaDeFiguritas[],int sizeDeFiguritas,eEquipo
listaDeEquipos[],int sizeEquipos,eAlbum* unAlbumDeFiguritas,eColeccionista*
unColeccionista,eClubDeFutbol[],int,eLiga[],int);

/// @fn int SubMenuAdministrador(eFiguritas[], int, eEquipo[], int)
/// @brief Muestra el submenu De Listados y le permite al usuario ingresar una opcion
///
/// @param eFiguritas[] el array de eFiguritas
/// @param int el tamaño del array de eFiguritas
/// @param eEquipo[] el array de eEquipo
/// @param int el tamaño del array de eEquipo
int SubMenuDeListados(eFiguritas listaDeFiguritas[],int sizeDeFiguritas,eEquipo
listaDeEquipos[],int sizeEquipos,
eAlbum* unAlbumDeFiguritas,eColeccionista* unColeccionista,int
cantidadDeSobres,float dineroGastadoParaCompletarElAlbum,eClubDeFutbol
listaDeClubs[],int sizeDeClubs,eLiga listaDeligas[],int sizeDeLigas);

/// @fn int AbrirSobre(eFiguritas[], int, int[], int, int)
/// @brief Guarda una cantidad determina de figuritas random dentro del array pasado por
parametro

```

```
///  
/// @param eFiguritas[] el array de eFiguritas  
/// @param int el tamaño del array de eFiguritas  
/// @param inr[] el array donde se va a guardar las figuritas  
/// @param int el tamaño del array de figuritas  
/// @param int la cantidad de figuritas que se va a guardar  
/// @return [-1] si hubo error en los parametros , [1] si esta ok  
int AbrirSobre(eFiguritas listaDeFiguritas[],int size,int arrayDeFiguritas[],int sizeAuxiliar,int  
cantidadDeFiguritas);
```

```
/// @fn int IntercambiarFiguritas(eFiguritas[], int, eEquipo[], int, int[], int, eClubDeFutbol[],  
int, eLiga[], int)  
/// @brief Permite al usuario intercambiar una figurita repetida ,por una figurita la lista de  
figuritas  
///  
/// @return [-1] si hubo error en los parametros ,[-2] si se cancelo el intercambio , [indice]  
de la figurita que se eligio,  
int IntercambiarFiguritas(eFiguritas listaDeFiguritas[],int sizeDeFiguritas,eEquipo  
listaDeEquipos[],int sizeEquipos,int arrayDeRepetidas[],int  
sizeDeRepetidas,eClubDeFutbol listaDeClubs[],int sizeDeClubs,eLiga listaDeLigas[],int  
sizeDeLiga);
```

```
/// @fn int PedirIdDeFiguritaRepetida(eFiguritas[], int, eEquipo[], int, eClubDeFutbol[], int,  
eLiga[], int, int[], int, char[], char[], int)
```

```

/// @brief Permite Ingresar le id una figurita de la lista de repetidas y devuelve el indice
donde esta la figurita que se eligio
///
/// @param int[] el array de repetidas
/// @param int el tamaño del array de eFiguritas
/// @param char[] el mensaje que indica que debe ingresar
/// @param char[] el mensaje que indica que debe ingresar un numero dentro del rango
pedido
/// @param int la cantidad de reintentos
/// @return [-1] si hubo error en los parametros ,[1] si esta ok
int PedirIdDeFiguritaRepetida(eFiguritas listaDeFiguritas[],int sizeDeFiguritas,eEquipo
listaDeEquipos[],int sizeEquipos,eClubDeFutbol listaDeClubs[],int sizeDeClubs,eLiga
listaDeLigas[],int sizeDeLigas,int arrayDeRepetida[],int sizeDeRepetidas,char
mensaje[],char mensajeError[],int reintentos);

/// @fn int MostrarListaDeRepetidas(int[], int, eFiguritas[], int, eEquipo[], int,
eClubDeFutbol[], int, eLiga[], int)
/// @brief Muestra la lista de figuritas repetidas con la cantidad de repeticiones en la pila
///
/// @return [-1] si la lista no tiene figuritas , [1] si esta ok
int MostrarListaDeRepetidas(int listaDeRepetidas[],int sizeDeRepetidas,eFiguritas
listaDeFiguritas[],int sizeDeFiguritas,eEquipo listaDeEquipos[],int
sizeEquipos,eClubDeFutbol listaDeClubs[],int sizeDeClubs,eLiga listaDeLigas[],int
sizeDeLigas);

```



```
/// @fn int CantidadDeFiguritasPegadasPorCadaClubs(eClubDeFutbol listaDeClubs[],int
sizeDeClubs,eFiguritas listaDeFiguritas[],int sizeDeLista,eAlbum*,eAuxiliar
arrayAuxiliar[],int sizeAuxiliar);
/// @brief Cuenta la cantidad de figuritas pegadas en el album por cada clubs de futbol y la
guarda en un array auxiliar
///
```

```
/// @param eAuxiliar[] un array auxiliar
/// @param sizeAuxiliar el tamaño del array auxiliar
/// @return [-1] si hubo error en los parametros , [1] si esta ok
int CantidadDeFiguritasPegadasPorCadaClubs(eClubDeFutbol listaDeClubs[],int
sizeDeClubs,eFiguritas listaDeFiguritas[],int sizeDeLista,eAlbum*,eAuxiliar
arrayAuxiliar[],int sizeAuxiliar);
```

```
/// @fn int ContarFiguritasPegadasDeUnClub(eFiguritas[], int, int[], int, int)
/// @brief Cuenta la cantidad de figuritas pegadas en el album de un clubs de futbol
///
/// @return la cantidad de figuritas que conto
int ContarFiguritasPegadasDeUnClub(eFiguritas listaDeFiguritas[],int sizeDeLista,int
arrayDeFiguritas[],int sizeDeFiguritas,int idDeClub);
```

```
/// @fn int MostrarPorcentajeDeFigutiasPegadasPorCadaClubs(eFiguritas[], int, eEquipo[],
int, eClubDeFutbol[], int, eLiga[], int, eAlbum*, eAuxiliar[], int)
```

```
/// @brief Muestra el porcentaje de figuritas que hay en el album por cada clubs
///
/// @return [-1] si hubo error en los parametros o si la lista esta vacia,[1] si esta ok
int MostrarPorcentajeDeFigutiasPegadasPorCadaClubs(eFiguritas listaDeFiguritas[],int
sizeDeLista,eEquipo listaDeEquipos[],int sizeEquipos,eClubDeFutbol listaDeClubs[],int
sizeDeClubs,eLiga listaDeligas[],int sizeDeLigas,eAlbum* unAlbumDeFiguritas ,eAuxiliar
arrayAuxiliar[],int sizeAuxiliar);
```

```
/// @fn int MostrarUnFiguritaPegadaDeUnClub(eFiguritas[], int, eEquipo[], int, eLiga[], int,
eAlbum*, eClubDeFutbol)
/// @brief Muestra las figuritas pegadas en el album de un club determinado
///
/// @return [-1] si hubo error en los parametros o si la lista esta vacia,[1] si esta ok
int MostrarUnFiguritaPegadaDeUnClub(eFiguritas listaDeFiguritas[],int
sizeDeLista,eEquipo listaDeEquipos[],int sizeEquipos,eLiga listaDeligas[],int
sizeDeLigas,eAlbum* unAlbumDeFiguritas,eClubDeFutbol unClubDeFutbol);
```

```
/// @fn int InitList([], int,[], int, int)
/// @brief Realiza una carga forzada en el array de figuritas y de Equipos
///
/// @return [-1] si no pudo realizar la carga forzada, [1] si pudo realizar la carga forzada
```

```
int CargaForzada(eFiguritas listaDeFiguritas[],int sizeDeFiguritas,eEquipo  
listaDeEquipos[],int sizeEquipos,eClubDeFutbol listaDeClubs[],int sizeDeClubs,eLiga  
listaDeLigas[],int sizeDeLiga,int maximo);
```

```
/// @fn int ContarFiguritasRepetidasDeUnArray(int[], int, int[], int, eFiguritas[], int)  
/// @brief Cuenta cuntas figuritas repetidas hay en el array  
///
```

```
/// @return [-1] si hubo error en los parametros o si la lista esta vacia,[la cantidad que  
conto] si esta ok
```

```
int ContarFiguritasRepetidasDeUnArray(int figuritas[],int size,int figuritasRepetidas[],int  
sizeDeRepetidas,eFiguritas listaDeFiguritas[],int sizeDeFiguritas);
```

```
/// @fn int CantidadDeFiguritasPegadasPorCadaLiga(eLiga[], int, eClubDeFutbol[], int,  
eFiguritas[], int, eAlbum*, eAuxiliar[], int)
```

```
/// @brief Cuenta la cantidad de figuritas pegadas en el album por cada liga y las guarda en  
un array auxlira
```

```
///
```

```
/// @return [-1] si hubo error en los parametros o si la lista esta vacia,[la cantidad de que  
conto] si esta ok
```

```
int CantidadDeFiguritasPegadasPorCadaLiga(eLiga listaDeLigas[],int  
sizeDeLigas,eClubDeFutbol listaDeClubs[],int sizeDeClubs,eFiguritas listaDeFiguritas[],int  
sizeDeLista,eAlbum*,eAuxiliar arrayAuxiliar[],int sizeAuxiliar);
```

```

/// @fn int ContarFiguritasPegadasDeUnaLiga(eFiguritas[], int, eClubDeFutbol[], int, int[],
int, int)
/// @brief Cueta la cantidad de figuritas pegadas en el album de una liga
///
/// @param int la liga que se va a contar
/// @return [-1] si hubo error en los parametros o si la lista esta vacia,[la cantidad de que
conto] si esta ok
int ContarFiguritasPegadasDeUnaLiga(eFiguritas listaDeFiguritas[],int
sizeDeLista,eClubDeFutbol listaDeClubs[],int sizeDeClubs,int arrayDeFiguritas[],int
sizeDeFiguritas,int idDeLiga);

/// @fn int MostrarFiguritasPegadaDeCadaLiga(eFiguritas[], int, eEquipo[], int,
eClubDeFutbol[], int, eAlbum*, int)
/// @brief Muestra las figuritas pegadas en el album de una liga determinada
/// @param int la liga que se va mostrar
/// @return [-1] si hubo error en los parametros o si la lista esta vacia,[1] si esta ok
int MostrarFiguritasPegadaDeCadaLiga(eFiguritas listaDeFiguritas[],int
sizeDeFiguritas,eEquipo listaDeEquipos[], int sizeEquipos,eClubDeFutbol
listaDeClubs[],int sizeDeClubs,eAlbum* unAlbumDeFiguritas,int idDeLiga);

/// @fn int MostrarUnaFiguritaPegadaDeUnaLiga(eFiguritas[], int, eEquipo[], int, eAlbum*,
eClubDeFutbol)
/// @brief Muestra las figuritas pegadas en el Album de un club determinado
///

```

```
/// @return [-1] si hubo error en los parametros o si la lista esta vacia,[1] si esta ok
int MostrarUnaFiguritaPegadaDeUnaLiga(eFiguritas listaDeFiguritas[],int
sizeDeLista,eEquipo listaDeEquipos[],int sizeEquipos,eAlbum*
unAlbumDeFiguritas,eClubDeFutbol unClubDeFutbol);
```

```
/// @fn int MostrarLigasConMayorCantidad(eFiguritas[], int, eEquipo[], int, eAlbum*,
eClubDeFutbol[], int, eLiga[], int, eAuxiliar[], int, int)
/// @brief Muestra la liga/s con mayor cantidad de figuritas pegadas en el album , con sus
figuritas
///
/// @param eAuxiliar[] el array auxiliar
/// @param int el tamaño del array auxiliar
/// @param int la mayor cantidad de figuritas contadas
/// @return [-1] si hubo error en los parametros o si la lista esta vacia,[1] si esta ok
int MostrarLigasConMayorCantidad(eFiguritas listaDeFiguritas[],int
sizeDeFiguritas,eEquipo listaDeEquipos[],int sizeEquipos,eAlbum*
unAlbumDeFiguritas,eClubDeFutbol listaDeClubs[],int sizeDeClubs,eLiga listaDeLigas[],int
sizeDeLigas,eAuxiliar arrayAuxiliar[],int sizeAuxiliar,int maximo);
```

```
/// @fn int OrdenarPorEquipo(eFiguritas[], int, eEquipo[], int)
/// @brief Ordena la lista de figurita por equipos alfabeticamente
///
/// @param listaDeFiguritas
/// @param size
```

```

/// @param listaDeEquipo
/// @param sizeDeEquipo
/// @return [-1] si hubo error en los parametros o si la lista esta vacia,[1] si esta ok
int OrdenarListaDeFigurasPorEquipo(eFiguritas listaDeFiguritas[],int
sizeDeFiguritas,eEquipo listaDeEquipo[],int sizeDeEquipo,eClubDeFutbol
listaDeClubs[],int sizeDeClubs,eLiga listaDeLigas[],int sizeDeLigas);

/// @fn int MostrarListaDeFiguritasDeUnEquipo(eFiguritas[], int, eEquipo, eClubDeFutbol[],
int, eLiga[], int)
/// @brief Muestra la lista de figuritas de un equipo determinado
///
/// @return [-1] si hubo error en los parametros o si la lista esta vacia,[1] si esta ok
int MostrarListaDeFiguritasDeUnEquipo(eFiguritas listaDeFiguritas[],int
sizeDeFiguritas,eEquipo unEquipo,eClubDeFutbol listaDeClubs[],int sizeDeClubs,eLiga
listaDeLigas[],int sizeDeLigas);

/// @fn int PedirIdDeFigurita(eFiguritas[], int, eEquipo[], int, eClubDeFutbol[], int, eLiga[], int,
char[], char[], int)
/// @brief Permite pedir un id una figurita y devuelve el indice donde se encuentra la figurita
elegida
///
/// @param char[] el mensaje que indica que debe ingresar
/// @param char[] el mensaje que indica que debe ingresar un numero dentro del rango
pedido

```

```

/// @param int la cantidad de reintentos
/// @return[-1] si hubo error en los parametros o no se ingreso correctamente,[INDICE] si
esta ok
int PedirIdDeFigurita(eFiguritas listaDeFiguritas[],int sizeDeFiguritas,eEquipo
listaDeEquipos[],int sizeEquipos,eClubDeFutbol listaDeClubs[],int sizeDeClubs,eLiga
listaDeLigas[],int sizeDeLigas,char mensaje[],char mensajeError[],int reintentos);

/// @fn int MostrarFiguritasDeUnArray(int[], int, eFiguritas[], int, eEquipo[], int,
eClubDeFutbol[], int, eLiga[], int)
/// @brief Muestra las figuritas que guarda el array pasado por parametro
///
/// @return [-1] si hubo error en los parametros o si la lista esta vacia,[1] si esta ok
int MostrarFiguritasDeUnArray(int figuritas[],int size,eFiguritas listaDeFiguritas[],int
sizeDeFiguritas,eEquipo listaDeEquipos[],int sizeDeEquipos,eClubDeFutbol
listaDeClubs[],int sizeDeClubs,eLiga listaDeLigas[],int sizeDeLigas);

/// @fn int MostrarUnaFiguritaCompleta(eFiguritas, eEquipo[], int, eClubDeFutbol[], int,
eLiga[], int)
/// @brief Busca los datos relacionados de una figurita y la muestra
///
/// @return [-1] si no encontro algun dato ,[1] si esta ok
int MostrarUnaFiguritaCompleta(eFiguritas listaDeFiguritas,eEquipo listaDeEquipos[],int
sizeDeEquipos,eClubDeFutbol listaDeClubs[],int sizeDeClub,eLiga listaDeLigas[],int
sizeDeLiga);

```

```

/// @fn int ListarFiguritasGuardadasDeUnEquipo(eEquipo, eFiguritas[], int, int[], int,
eClubDeFutbol[], int, eLiga[], int)
/// @brief Muestra las figuritas pegadas en el album de un equipo determinado
///
/// @param eEquipo el equipo que se va a buscar la figuritas
/// @return [-1] si el array esta vacio , [1] si esta ok
int ListarFiguritasGuardadasDeUnEquipo(eEquipo unEquipo,eFiguritas
listaDeFiguritas[],int sizeDeFiguritas,int arrayDeFiguritas[],int
sizeDelAlbum,eClubDeFutbol listaDeClubs[],int sizeDeClubs,eLiga listaDeligas[],int
sizeDeLigas);

```

```

/// @fn int CantidadDeFiguritasGuardadasDeUnEquipo(eEquipo, eFiguritas[], int, int[], int)
/// @brief Cuenta la cantidad de figuritas pegadas en el album de un equipo determinado
///
/// @param eEquipo el equipo que se va a buscar la figuritas
/// @param eFiguritas[] el array de eFiguritas
/// @param int el tamaño del array de eFiguritas
/// @param int[] el array donde se va a buscar las figuritas
/// @param int el tamaño del array

```



```
/// @return [-1] si el array esta vacio , [la cantidad que conto] si esta ok  
int CantidadDeFiguritasGuardadasDeUnEquipo(eEquipo unEquipo,eFiguritas  
listaDeFiguritas[],int sizeDeFiguritas,int arrayDeFiguritas[],int sizeDeArray);
```