

Apellido y nombre: \_\_\_\_\_ Legajo: \_\_\_\_\_ Curso con Prof: \_\_\_\_\_

Cantidad de hojas entregadas: \_\_\_\_\_ Nota: \_\_\_\_\_ Evaluó Prof: \_\_\_\_\_

- Si luego de la lectura del examen, durante la resolución tiene alguna duda, escriba hipótesis de trabajo, las cuales también serán evaluadas.
- Los puntos que solicitan codificación puede ser respondidos en C, ó C++, pero debe indicar el lenguaje utilizado.
- En C y C++ prototipo refiere a la declaración de la función, es decir tipo de dato retornado, nombre de la función, y tipos de los parámetros.

## Sistema para el Seguimiento de Jugadores de Fútbol

*Temas evaluados: Resolución de problemas, estructuras de datos, archivos, listas, y lenguaje de programación.*

### Contexto

Usted es parte de un equipo que desarrolla aplicaciones para la FIFA, y es el responsable de la convocatoria de jugadores que actualmente estén jugando en la liga europea, de fútbol para ser parte de la selección que los convoque.

### Descripción

Para que un jugador pueda ser convocado a la selección se requiere que sea un ciudadano nativo del país convocante, menor de 30 años y no haber participado en el pasado en la selección de ningún país distinto al que lo convoca. La función **edadDelCandidato** es una función que dada una fecha de nacimiento retorna la edad en años cumplidos. La función **jugoEnOtraSelección** retorna el valor verdadero en caso que haya jugado en otra selección y falso en caso contrario.

Los flujos binarios que se disponen responden al siguiente formato:

#### NóminaDeJugadores.dat:

Id_Jugador	Nombres	Fecha_Nacimiento	Pais	Cantidad_goles
Entero	50 caracteres	Entero aaaammdd	50 caracteres	entero

#### SeleccionDondeJugo.dat

Id_Jugador	Pais
Entero	50 Caracteres

### Problema

Se necesita saber si un jugador **puedeSerCitado**, es decir, si cumple con las restricciones de nacionalidad, edad y no desempeño en otra selección nacional.

### Asunciones (Precondiciones)

- En cada archivo hay un solo registro por jugador.
- Para alcanzar eficiencia en la búsqueda deben evitarse datos superfluos en memoria. Por lo que los datos deben asignarse en forma dinámica y solo aquellos que son necesarios.

### Restricciones

Utilice la función de biblioteca InsertarNodo y las estructuras declaradas al final

La función **edadDelCandidato** es dada y responde al siguiente prototipo **int edadDelCandidato(int fechaDeNacimiento)**

La función **jugoEnOtraSeleccion** no debe definirla pero si escribir su prototipo

La función **puedeSerCitado** debe respetar las siguientes restricciones:

- Debe recibir un jugador y la selección a la que se desea citarlo.
- Debe invocar a las funciones **jugoEnOtraSeleccion** y **edadDelCandidato**

### Se pide

1. Codificar o diagramar **cargarDatosEnMemoria** que reciba dos flujos binarios y que retorne o devuelva punteros a estructuras enlazadas con los datos validos de esos flujos a los efectos de facilitar las búsquedas.
2. Codificar el prototipo de **jugoEnOtraSeleccion** definiendo parámetros y justificando la elección.
3. Codificar o diagramar la función **puedeSerCitado**.

<pre> struct jugador {     unsigned id_jugador;     char nombres[50 + 1];     unsigned fecha_nacimiento;     char pais[50 + 1];     unsigned cantidad_goles; };  struct jugo_en {     unsigned id_jugador;     char pais[50 + 1]; }; </pre>	<pre> struct NodoListaJugadores {     jugador info;     NodoListaJugadores *sgte; };  struct NodoListaJugoEn {     jugo_en info;     NodoListaJugoEn *sgte; }; </pre>
---	---

UTN – FRBA – Algoritmos y Estructura de Datos – Examen Final – 13/02/2015

Apellido y nombre: \_\_\_\_\_ Legajo: \_\_\_\_\_ Cursó con Prof: \_\_\_\_\_

Cantidad de hojas entregadas: \_\_\_\_\_ Nota: \_\_\_\_\_ Evaluó Prof: \_\_\_\_\_

- Si luego de la lectura del examen, durante la resolución tiene alguna duda, escriba hipótesis de trabajo, las cuales también serán evaluadas.
- Los puntos que solicitan codificación puede ser respondidos en C, ó C++, pero debe indicar el lenguaje utilizado.
- En C y C++ prototipo refiere a la declaración de la función, es decir tipo de dato retornado, nombre de la función, y tipos de los parámetros.

```
#include <iostream>
```

```
#include <cstdio>
```

```
#include <string.h>
```

```
#define RUTA_JUGADORES "NominaDeJugadores.dat"
```

```
#define RUTA_DONDEJUGO "SeleccionDondeJugo.dat"
```

```
using namespace std;
```

```
/*Declaración de estructuras necesarias para resolver el programa*/
```

```
struct jugador
```

```
{  
    unsigned id_jugador;  
    char nombres[50 + 1];  
    unsigned fecha_nacimiento;  
    char pais[50 + 1];  
    unsigned cantidad_goles;  
};
```

```
struct jugo_en
```

```
{  
    unsigned id_jugador;  
    char pais[50 + 1];  
};
```

```
struct NodoListaJugadores
```

```
{  
    jugador info;  
    NodoListaJugadores *sgte;  
};
```

```
struct NodoListaJugoEn
```

```
{  
    jugo_en info;  
    NodoListaJugoEn *sgte;  
};
```

## UTN – FRBA – Algoritmos y Estructura de Datos – Examen Final – 13/02/2015

Apellido y nombre: \_\_\_\_\_ Legajo: \_\_\_\_\_ Curso con Prof: \_\_\_\_\_

Cantidad de hojas entregadas: \_\_\_\_\_ Nota: \_\_\_\_\_ Evaluó Prof: \_\_\_\_\_

- Si luego de la lectura del examen, durante la resolución tiene alguna duda, escriba hipótesis de trabajo, las cuales también serán evaluadas.
- Los puntos que solicitan codificación puede ser respondidos en C, ó C++, pero debe indicar el lenguaje utilizado.
- En C y C++ prototipo refiere a la declaración de la función, es decir tipo de dato retornado, nombre de la función, y tipos de los parámetros.

### Punto 1

Codificar o diagramar `cargarDatosEnMemoria` que reciba dos flujos binarios y que retorne o devuelva punteros a estructuras enlazadas con los datos validos de esos flujos a los efectos de facilitar las búsquedas.

```
void cargarDatosEnMemoria(FILE *archJugadores, FILE *archjugoen, NodoListaJugadores *&listajugadores, NodoListaJugoEn *&listajugoen)
{
    listajugadores= crearListaJugadores(archJugadores);
    listajugoen=crearListaDondeJugo(archjugoen);
    return;
}
/*FIN RESPUESTA PUNTO 1*/
```

### Funciones utilizadas

/\*\*\*\*\*

Función `crearListaJugadores`.

Precondiciones: Debe recibir un archivo de jugadores ya abierto.

Postcondiciones: Devuelve una lista SE con los jugadores que existen en el archivo o

NULL en caso que no hubieran jugadores.

\*\*\*\*\*/

`NodoListaJugadores *crearListaJugadores(FILE *data)`

```
{
    NodoListaJugadores *ljugadores=NULL;
    jugador j;
    while (fread(&j, sizeof(jugador),1,data))
    {
        insertarNodoJugadorAlFinal(ljugadores, j);
    }
    fclose(data);
    return ljugadores;
}
```

/\*\*\*\*\*

Función `crearListaDondeJugo`.

Precondiciones: Debe recibir un archivo, ya abierto, de selecciones donde jugaron los participantes.

Postcondiciones: Devuelve una lista SE con las selecciones donde jugaron los participantes o

NULL en caso que no hubieran jugadores.

\*\*\*\*\*/

`NodoListaJugoEn *crearListaDondeJugo(FILE *data)`

```
{
    NodoListaJugoEn *ljugoen=NULL;
    jugo_en j;
    while (fread(&j, sizeof(jugo_en),1,data))
    {
        insertarNodoJugoEnAlFinal(ljugoen, j);
    }
    fclose(data);
    return ljugoen;
}
```

Apellido y nombre: \_\_\_\_\_ Legajo: \_\_\_\_\_ Curso con Prof: \_\_\_\_\_

Cantidad de hojas entregadas: \_\_\_\_\_ Nota: \_\_\_\_\_ Evaluó Prof: \_\_\_\_\_

- Si luego de la lectura del examen, durante la resolución tiene alguna duda, escriba hipótesis de trabajo, las cuales también serán evaluadas.
- Los puntos que solicitan codificación puede ser respondidos en C, ó C++, pero debe indicar el lenguaje utilizado.
- En C y C++ prototipo refiere a la declaración de la función, es decir tipo de dato retornado, nombre de la función, y tipos de los parámetros.

Punto 2

Codificar el prototipo de `jugoEnOtraSeleccion` definiendo parámetros y justificando la elección.

Precondiciones: Debe recibir la lista de selecciones históricas de los jugadores, el id del participante en cuestión y el país para el que se postula.

Postcondiciones: Devuelve true si jugó en una selección distinta a la que se postula o false en caso contrario.

\*\*\*\*\*/

```
bool jugoEnOtraSeleccion(NodoListaJugoEn *listajugoen, int id_jugador, char pais_postula[])
```

definición de la función no exigida en la evaluación

```
bool jugoEnOtraSeleccion(NodoListaJugoEn *listajugoen, int id_jugador, char pais_postula[])
{
    while (listajugoen!=NULL)
    {
        if ((id_jugador== listajugoen->info.id_jugador) && (strcmp(listajugoen->info.pais, pais_postula)!=0))
            return true; //encuentra al jugador y jugo en otra seleccion
        listajugoen= listajugoen->sgte;
    }
    return false;
}
```

Punto 3

Codificar o diagramar la función **puedeSerCitado**.

Precondiciones: Recibe id de un jugador, el país al que se postula, una lista de jugadores y una lista de selecciones donde participaron los jugadores. El id del jugador DEBE existir en la lista de jugadores.

Postcondiciones: Devuelve true si el jugador es menor de 30 años y no jugó en una selección distinta a la que se postula. False en caso contrario.

Utiliza

- `jugoEnOtraSeleccion`
- `edadDelCandidato`
- `BuscarNodoJugador`

\*\*\*\*\*/

```
bool puedeSerCitado(int id_jugador, char pais[],NodoListaJugadores *listajugadores, NodoListaJugoEn *listajugoen)
{
    if (!jugoEnOtraSeleccion(listajugoen, id_jugador,pais) && edadDelCandidato((BuscarNodoJugador(listajugadores, id_jugador))->info.fecha_nacimiento)<30)
        return true;
    return false;
}
// puede ser citado si NO JUGO EN OTRA SELECCIÓN Y LA EDAD ES MENOR A 30 AÑOS
```

Apellido y nombre: \_\_\_\_\_ Legajo: \_\_\_\_\_ Cursó con Prof: \_\_\_\_\_

Cantidad de hojas entregadas: \_\_\_\_\_ Nota: \_\_\_\_\_ Evaluó Prof: \_\_\_\_\_

- Si luego de la lectura del examen, durante la resolución tiene alguna duda, escriba hipótesis de trabajo, las cuales también serán evaluadas.
- Los puntos que solicitan codificación puede ser respondidos en C, ó C++, pero debe indicar el lenguaje utilizado.
- En C y C++ prototipo refiere a la declaración de la función, es decir tipo de dato retornado, nombre de la función, y tipos de los parámetros.

## FUNCIONES COMPLEMENTARIAS

/\*\*\*\*\*

Procedimiento CargaDatosPruebaJugadores.

Precondiciones: Existencia de espacio y permisos para poder escribir archivos en disco.

Postcondiciones: crea un archivo de jugadores (con ruta y nombre indicados en el define).

\*\*\*\*\*/

```
void CargaDatosDePruebaJugadores()
{
    jugador j;
    FILE *f;
    f=fopen(RUTA_JUGADORES,"w");
    j.id_jugador = 1;
    strcpy(j.nombres,"Messi Leo");
    j.fecha_nacimiento=19870627;
    strcpy(j.pais,"Argentina");
    j.cantidad_goles=386;
    fwrite(&j,sizeof(j),1,f);
    j.id_jugador = 2;
    strcpy(j.nombres,"Mascherano Javier");
    j.fecha_nacimiento=19850608;
    strcpy(j.pais,"Argentina");
    j.cantidad_goles=3;
    fwrite(&j,sizeof(j),1,f);
    j.id_jugador = 3;
    strcpy(j.nombres,"Francisco dos Santos");
    j.fecha_nacimiento=19331028;
    strcpy(j.pais,"Brasil");
    j.cantidad_goles=244;
    fwrite(&j,sizeof(j),1,f);
    j.id_jugador = 4;
    strcpy(j.nombres,"Lucas Barrios");
    j.fecha_nacimiento=19901028;
    strcpy(j.pais,"Argentina");
    j.cantidad_goles=15;
    fwrite(&j,sizeof(j),1,f);
    fclose(f);
}
```

/\*\*\*\*\*

Procedimiento CargaDatosPruebaJugoEn.

Precondiciones: Existencia de espacio y permisos para poder escribir archivos en disco.

Postcondiciones: crea un archivo (con ruta y nombre indicados en el define)

con las seleccion(es) donde jugaron los jugadores.

\*\*\*\*\*/

```
void CargaDatosDePruebaJugoEn()
{
    jugo_en j;
    FILE *f;
    f=fopen(RUTA_DONDEJUGO,"w");
```

## UTN – FRBA – Algoritmos y Estructura de Datos – Examen Final – 13/02/2015

Apellido y nombre: \_\_\_\_\_ Legajo: \_\_\_\_\_ Curso con Prof: \_\_\_\_\_

Cantidad de hojas entregadas: \_\_\_\_\_ Nota: \_\_\_\_\_ Evaluó Prof: \_\_\_\_\_

- Si luego de la lectura del examen, durante la resolución tiene alguna duda, escriba hipótesis de trabajo, las cuales también serán evaluadas.
- Los puntos que solicitan codificación puede ser respondidos en C, ó C++, pero debe indicar el lenguaje utilizado.
- En C y C++ prototipo refiere a la declaración de la función, es decir tipo de dato retornado, nombre de la función, y tipos de los parámetros.

```
j.id_jugador = 3;
strcpy(j.pais,"Brasil");
fwrite(&j,sizeof(j),1,f);
j.id_jugador = 1;
strcpy(j.pais,"Argentina");
fwrite(&j,sizeof(j),1,f);
j.id_jugador = 2;
strcpy(j.pais,"Argentina");
fwrite(&j,sizeof(j),1,f);
j.id_jugador = 4;
strcpy(j.pais,"Paraguay");
fwrite(&j,sizeof(j),1,f);
fclose(f);
}
```

/\*\*\*\*\*

Procedimiento CargaDatosPrueba.

Precondiciones: Existencia de espacio y permisos para poder escribir archivos en disco.

Postcondiciones: crea dos archivos (con ruta y nombre indicados en el define)

con jugadores y lista donde jugaron. Para ello utiliza:

- CargaDatosPruebaJugadores

- CargaDatosPruebaJugoEn

\*\*\*\*\*/

```
void CargaDatosDePrueba()
{
    CargaDatosDePruebaJugadores();
    CargaDatosDePruebaJugoEn();
}
```

/\*\*\*\*\*

Procedimiento insertarNodoJugadorAlFinal.

Precondiciones: Se debe pasar una estructura jugador con los datos ya inicializados.

Postcondiciones: Agrega al final de la lista parámetro un nuevo nodo con los datos de la estructura indicada.

\*\*\*\*\*/

```
void insertarNodoJugadorAlFinal(NodoListaJugadores *&lista, jugador j)
{
    if (lista==NULL)
    {
        lista= new NodoListaJugadores();
        lista->info = j;
        lista->sgte = NULL;
    }
    else
    {
        NodoListaJugadores *aux=lista;
        while (aux->sgte!=NULL)
        {
            aux= aux->sgte;
        }
        aux->sgte= new NodoListaJugadores();
    }
}
```

## UTN – FRBA – Algoritmos y Estructura de Datos – Examen Final – 13/02/2015

Apellido y nombre: \_\_\_\_\_ Legajo: \_\_\_\_\_ Curso con Prof: \_\_\_\_\_

Cantidad de hojas entregadas: \_\_\_\_\_ Nota: \_\_\_\_\_ Evaluó Prof: \_\_\_\_\_

- Si luego de la lectura del examen, durante la resolución tiene alguna duda, escriba hipótesis de trabajo, las cuales también serán evaluadas.
- Los puntos que solicitan codificación puede ser respondidos en C, ó C++, pero debe indicar el lenguaje utilizado.
- En C y C++ prototipo refiere a la declaración de la función, es decir tipo de dato retornado, nombre de la función, y tipos de los parámetros.

```
        aux = aux->sgte;
        aux->info= j;
        aux->sgte = NULL;
    }
    return;
}
```

/\*\*\*\*\*

Procedimiento insertarNodoJugoEnAlFinal.

Precondiciones: Se debe pasar una estructura jugado\_en con los datos ya inicializados.

Postcondiciones: Agrega al final de la lista parámetro un nuevo nodo con los datos de la estructura indicada.

\*\*\*\*\*/

```
void insertarNodoJugoEnAlFinal(NodoListaJugoEn *&lista, jugo_en j)
```

```
{
    if (lista==NULL)
    {
        lista= new NodoListaJugoEn();
        lista->info = j;
        lista->sgte = NULL;
    }
    else
    {
        NodoListaJugoEn *aux=lista;
        while (aux->sgte!=NULL)
        {
            aux= aux->sgte;
        }
        aux->sgte= new NodoListaJugoEn();
        aux = aux->sgte;
        aux->info= j;
        aux->sgte = NULL;
    }
    return;
}
```

/\*\*\*\*\*

Función BuscarNodoJugador.

Precondiciones: Se debe pasar la lista de jugadores y el id del jugador buscado.

Postcondiciones: Devuelve un puntero al jugador encontrado o NULL si no se encontró.

\*\*\*\*\*/

```
NodoListaJugadores* BuscarNodoJugador(NodoListaJugadores *lista, int id_jugador)
```

```
{
    while (lista)
    {
        if (lista->info.id_jugador==id_jugador)
            return lista;
        lista= lista->sgte;
    }
    return lista;
}
```

# UTN – FRBA – Algoritmos y Estructura de Datos – Examen Final – 13/02/2015

Apellido y nombre: \_\_\_\_\_ Legajo: \_\_\_\_\_ Cursó con Prof: \_\_\_\_\_

Cantidad de hojas entregadas: \_\_\_\_\_ Nota: \_\_\_\_\_ Evaluó Prof: \_\_\_\_\_

- Si luego de la lectura del examen, durante la resolución tiene alguna duda, escriba hipótesis de trabajo, las cuales también serán evaluadas.
  - Los puntos que solicitan codificación puede ser respondidos en C, ó C++, pero debe indicar el lenguaje utilizado.
  - En C y C++ prototipo refiere a la declaración de la función, es decir tipo de dato retornado, nombre de la función, y tipos de los parámetros.
- }

/\*\*\*\*\*

Función BuscarNodoJugoEn.

Precondiciones: Se debe pasar la lista selecciones donde jugaron los jugadores y el id del jugador buscado.

Postcondiciones: Devuelve un puntero al jugador encontrado o NULL si no se encontró.

\*\*\*\*\*/

NodoListaJugoEn\* BuscarNodoJugoEn(NodoListaJugoEn\*lista, int id\_jugador)

```
{
    while (lista)
    {
        if (lista->info.id_jugador==id_jugador)
            return lista;
        lista= lista->sgte;
    }
    return lista;
}
```

/\*\*\*\*\*

Procedimiento cargarDatosEnMemoria

Precondiciones: Debe recibir un archivo de jugadores ya abierto, un archivo de listado de selecciones donde participaron los jugadores, también ya abierto. Además debe recibir la lista de jugadores y lista de selecciones históricas de los jugadores AMBAS EN NULL, de lo contrario se perderán los datos preexistentes en las listas.

Postcondiciones: Devuelve las listas pasadas por parámetros con los datos cargados.

Utiliza

- crearListaDondeJugo.
- crearListaDondejugadores.

\*\*\*\*\*/

/\*\*\*\*\*

Función edadDelCandidato.

Precondiciones: recibe una fecha en formato AAAAMMDD en un integer

Postcondiciones: Devuelve la cantidad de años desde el 2015

\*\*\*\*\*/

int edadDelCandidato(int fechanacimiento)

```
{
    int agnos = 2015 - fechanacimiento / 10000;
    return agnos;
}
```

/\*\*\*\* COMPLEMENTO: FUNCION QUE MUESTRA EL CONTENIDO DE LOS ARCHIVOS \*\*\*\*\*/

```
void muestraArchivos()
{
    jugador j;
    jugo_en jug;
    FILE *fnomina;
    FILE *fdondejugo;
    fnomina= fopen(RUTA_JUGADORES,"r");
}
```



Apellido y nombre: \_\_\_\_\_ Legajo: \_\_\_\_\_ Curso con Prof: \_\_\_\_\_

Cantidad de hojas entregadas: \_\_\_\_\_ Nota: \_\_\_\_\_ Evaluó Prof: \_\_\_\_\_

- Si luego de la lectura del examen, durante la resolución tiene alguna duda, escriba hipótesis de trabajo, las cuales también serán evaluadas.
- Los puntos que solicitan codificación puede ser respondidos en C, ó C++, pero debe indicar el lenguaje utilizado.
- En C y C++ prototipo refiere a la declaración de la función, es decir tipo de dato retornado, nombre de la función, y tipos de los parámetros.

```
while (fread(&j, sizeof(jugador),1,fnomina))
{
    cout<< j.id_jugador << endl;
    cout<< j.nombres<< endl;
    cout<< j.fecha_nacimiento<< endl;
    cout<< j.cantidad_goles << endl;
    cout<< j.pais << endl;
    cout << "*****" << endl;
}
fclose(fnomina);
fdondejugo= fopen(RUTA_DONDEJUGO,"r");
while (fread(&jug, sizeof(jug),1,fnomina))
{
    cout<< jug.id_jugador << endl;
    cout<< jug.pais << endl;
    cout << "*****" << endl;
}
fclose(fdondejugo);
return;
}
/*****/
int main()
{
    CargaDatosDePrueba();
    FILE *fnomina;
    FILE *fdondejugo;
    NodoListaJugadores *ljugadores=NULL;
    NodoListaJugoEn *ljugoen=NULL;
    NodoListaJugadores *aux;
    bool puede;
    //muestraArchivos();
    if ((fnomina =fopen(RUTA_JUGADORES,"r")) && (fdondejugo =fopen(RUTA_DONDEJUGO,"r")))
    {
        cargarDatosEnMemoria(fnomina, fdondejugo, ljugadores, ljugoen);
        /*Cerramos los archivos, total, como ya tenemos los datos en memoria para procesar no los vamos a usar */
        fclose(fnomina);
        fclose(fdondejugo);
        /* Ahora vamos ir recorriendo la lista de jugadores e indicando si puede jugar o no cada uno de ellos */
        aux = ljugadores;
        while (aux)
        {
            puede =puedeSerCitado(aux->info.id_jugador,aux->info.pais,ljugadores,ljugoen);
            if (puede)
                cout << "El jugador " << aux->info.nombres << " puede jugar para " << aux->info.pais << endl;
            else
                cout << "El jugador " << aux->info.nombres << " NO puede jugar para " << aux->info.pais << endl;
            aux= aux->sgte;
        }
    }
    else
        cout << "Error al intentar abrir un archivo" << endl;
    return 0;
}
```