

## **Introducción:**

FIRST (FIFA Ranking System) es un programa de cálculo de puntos para los equipos federados por la FIFA, compilado para la plataforma Windows bajo MS-DOS en modo protegido 32-bit.

Este programa permite el ingreso y actualización de selecciones, confederaciones, torneos como el ingreso de nuevos juegos dentro de éstos. Permite el cálculo mediante juegos históricos cuantos puntos lleva cada equipo en cierto período de tiempo y el premio al mejor desempeño y al mejor equipo del año especificado. Todos los comandos se tendrán que dar en un archivo de entrada TXT especificado por el usuario y el reporte será dado en otro archivo TXT especificado por el usuario.

La interfaz del programa es muy simple, y contiene un menú principal que permite especificar lo que se desea hacer. Sin embargo cualquier duda se puede consultar en este documento, cuya finalidad es explicar paso a paso las características técnicas del programa.

Este programa esta hecho en el lenguaje ANSI C/ISO C++, compilado bajo Borland Turbo C++ 3.1, y es compatible para ser compilado bajo GCC desactivándole la constante BC.

## Tabla de Contenidos:

<b><u>INTRODUCCIÓN:</u></b>	<b><u>1</u></b>
<b><u>TABLA DE CONTENIDOS:</u></b>	<b><u>2</u></b>
<b><u>DOCUMENTO DE ANALISIS, PROYECTO FIRST (FIFA RANKING SYSTEM)</u></b>	<b><u>3</u></b>
PLANTEO INICIAL DE LA SOLUCIÓN:	3
REQUERIMIENTOS:	3
ALCANCES:	3
LIMITACIONES:	3
CALENDARIO:	4
ARQUITECTURA:	5
<b><u>DOCUMENTO DE DISEÑO, PROYECTO FIRST (FIFA RANKING SYSTEM)</u></b>	<b><u>6</u></b>
MODULOS:	6
MODULO UI: INTERFAZ DE USUARIO	6
MÓDULO SCANNER: ANALIZADOR LÉXICO	6
MODULO PARSER: ANALIZADOR SINTÁCTICO	6
MOTOR ENTRADA/MODIFICACIÓN:	6
MOTOR CONSULTA DE DATOS:	6
LÓGICA:	7
A CONTINUACIÓN SE PRESENTA EL MODELO DE ARCHIVOS A UTILIZAR.	7
REGISTROS Y ESTRUCTURAS:	7
<b><u>PRUEBAS:</u></b>	<b><u>8</u></b>
<b><u>MANUAL DE USUARIO</u></b>	<b><u>9</u></b>
REQUISITOS DEL SISTEMA:	9
INSTRUCCIONES DE USO:	9
PRE-PASOS:	9
PASOS:	9
SINTAXIS DE ARCHIVO DE ENTRADA:	10
<b><u>MANUAL TÉCNICO.</u></b>	<b><u>11</u></b>

## **Documento de Analisis, Proyecto FIRST (Fifa Ranking SysTem)**

### **Planteo Inicial de la Solución:**

- Automatizar el sistema de clasificación FIFA/Coca-Cola aprovechando los recursos computacionales que se ofrece en la actualidad, utilizando el lenguaje de programación C.
- Proporcionar una interfaz de usuario intuitiva y fácil de utilizar. Aprovechando la el muestreo de texto ASCII en pantalla (stdout).
- Generar un código estable y maduro para la fecha planificada, implantando toda la funcionalidad requerida con el menor número de errores posibles.
- Hacer que éste código generado corra en máquinas Windows, y si es posible, portarlo a GNU/Linux.
- Establecer como formato de salida un documento HTML, hacia donde se escribirá el reporte.
- Escribir un código fuente claro, con formato legible y bien documentado.

### **Requerimientos:**

- Software: Sistema operativo que soporte operaciones con archivos, compilador de C++ estándar.
- Hardware: Computador con microprocesador compatible con Intel i386, sistema de almacenamiento secundario.

### **Alcances:**

- Se manejará únicamente archivos binarios dentro de las operaciones realizadas por el programa.
- La entrada de datos será únicamente por un archivo de entrada en una sintaxis específica.

### **Limitaciones:**

- La interfaz de usuario será únicamente en modo texto,
- El archivo de datos deberá seguir la sintaxis estricta para que se puedan hacer las operaciones correspondientes, de otro modo se ignorará la operación actual y se escribirá el error en el reporte.
- Posiblemente por razón de tiempo, la salida será únicamente en TXT, por lo que la salida en HTML se considera como una opción.

### Calendario:

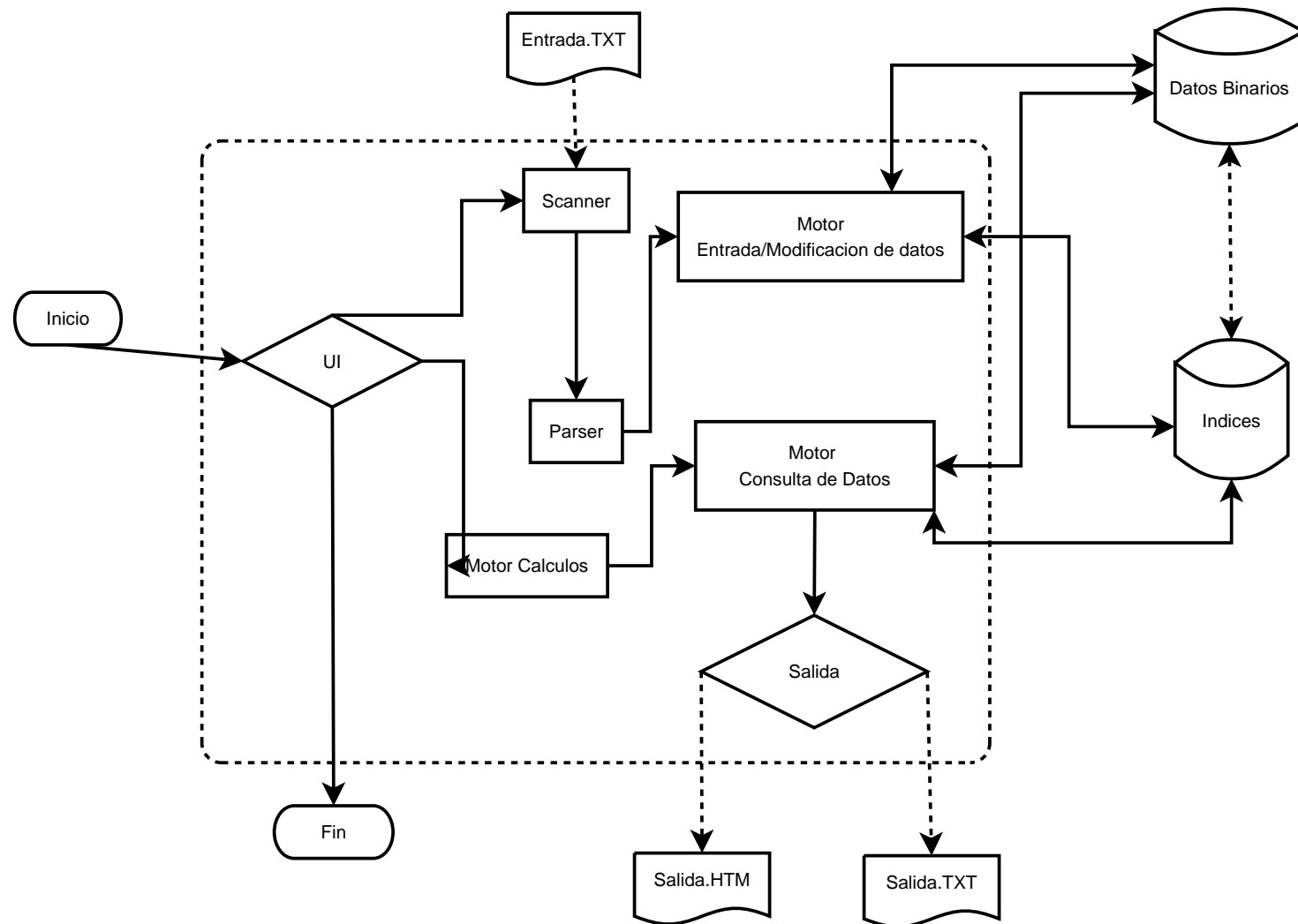
3 – 5 de Septiembre: Análisis y Diseño

6 – 15 de Septiembre: Implantación, Revisiones al diseño.

16 – 19 de Septiembre: Pruebas, Mejoras al Código.

20 – 22 de Septiembre: Documentación y Pruebas finales.

## Arquitectura:



## Documento de Diseño, Proyecto FIRST (Fifa Ranking SysTem)

### Modulos:

A continuación se presentan una lista de posibles modulos a implantar en el proyecto. Conforme se avance en la implantación, puede ser que se requiera de más modulos.

#### Modulo UI: Interfaz de Usuario

Int menu(void): Dibuja menú principal y devuelve opcion seleccionada.

char \* setinput(void): Devuelve path a archivo de entrada.

char \* setoutput(void): Devuelve path a archivo de salida.

#### Módulo Scanner: Analizador Léxico

Librería scan.cpp proporcionada.

#### Modulo Parser: Analizador Sintáctico

Parse(char \*): Lee las instrucciones especificas y las analiza.

#### Motor Entrada/Modificación:

Int crearC(&struct): Crea archivo de confederación.

Int crearS(&struct): Crea archivo de selección.

Int crearT(&struct) : Crea archivo de torneos.

Int crearJ(&struct) : Crea archivo de juegos.

#### Motor Consulta de Datos:

Int puntos(int,int,int,int): genera reporte puntos.

Int premioME(int): genera reporte mejor equipo.

Int premioD(int): genera reporte mejor desempeño

Int clasif(int,int): genera reporte clasificacion FIFA/Cocacola

#### Salida:

int Output(char \*, int): genera salida hacia el reporte.

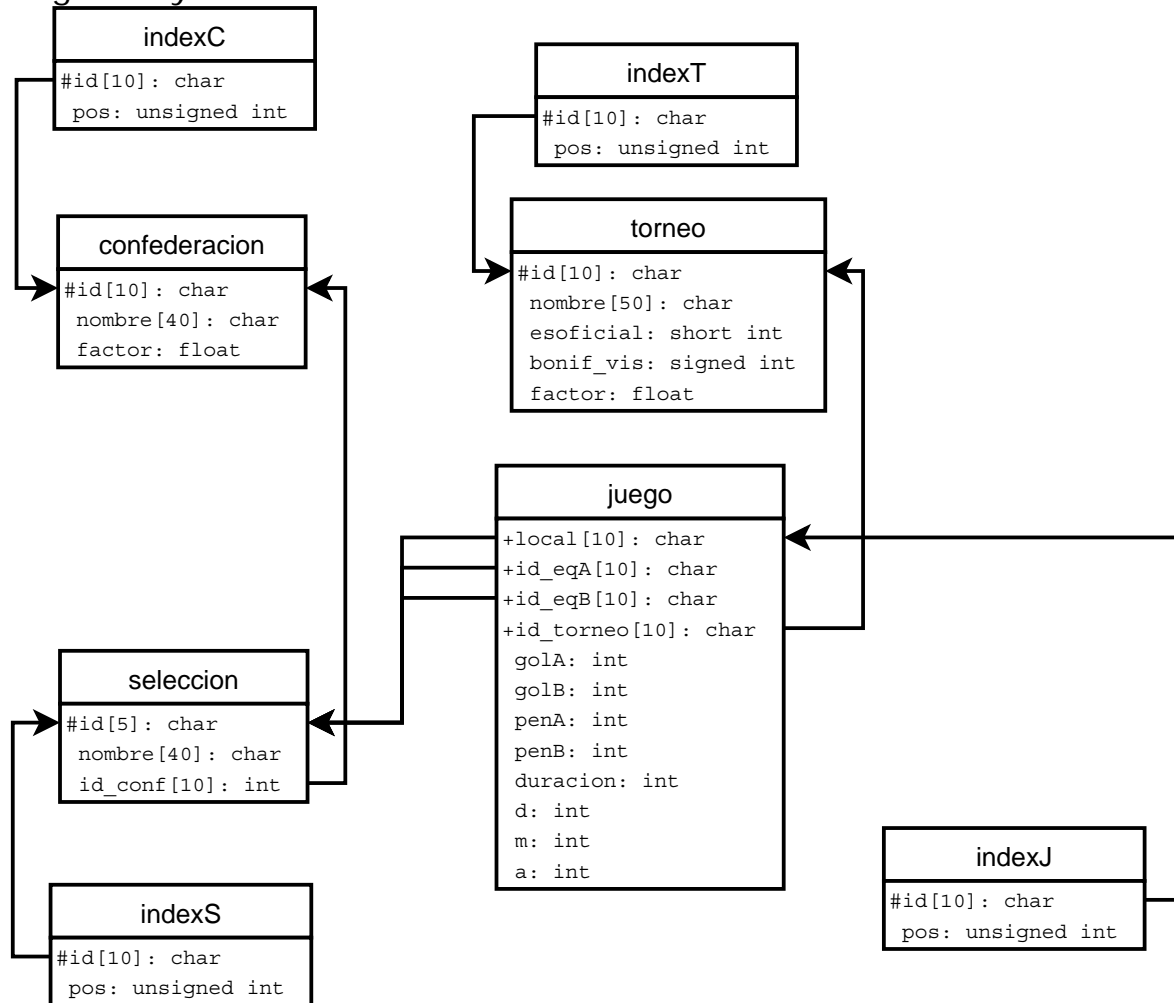
int MsgErr(char\*): genera mensaje de aviso a la pantalla y al reporte.

int buscapos(int): busca posiciones para modificar registros.

Lógica:

A continuación se presenta el modelo de archivos a utilizar.

Registros y Estructuras:



# Pruebas:

```

main.cpp - emacs@localhost.localdomain
File Edit Options Buffers Tools C++ Help

return -1;
}

//*****CalculaPuntosJuego()*****
// calcula puntos acumulados para cada equipo depenediendo del
// juego
int CalculaPuntosJuego(juego &RegData){
    float PCEA = 0, PCEB = 0, /*fit = 0, /*farea = 0, fnreb = 0, bonif = 0
    FCEA = 0, FCEB = 0, pra = 0, prb = 0, totalA = 0, totalB = 0,
    bonifa = 0, bonifB = 0;
    // float PAA_0 = 0, PAA_1 = 0, PAB_0 = 0, PAB_1 = 0;
    float factor;
    // bool salir = false;
    int ResPartido = 0, puntosprevA = 0, puntosprevB = 0;
    // int size = 0, i = 0, j = 0;
    FILE * in;
    // juego tmp;
    torneo torn;

    // -->Puntos Iniciales
    if (RegData.gola > RegData.golB){
main.cpp (C++ RCS:1.26 Abbrev)--L1204--48%
cd /mnt/diskd/programa/IPC2/proy1/
make -k
g++ -g -gdb -Wall -o main main.cpp
In file included from main.cpp:47:
scanner.cpp: In function 'int File_Out(char*)':
scanner.cpp:155: aviso: variable 'nom' sin uso
scanner.cpp: In function 'int identifica_reservada()':
scanner.cpp:271: aviso: comparaci3n entre expresiones enteras sign
main.cpp: In function 'void QuickSort(idxpuntosac*, int, int)':
main.cpp:940: aviso: converting a 'int' desde 'float'
main.cpp:947: aviso: converting a 'int' desde 'float'
main.cpp: In function 'int GuardarPuntos(char*, float)':
main.cpp:1076: aviso: el control alcanza el final de una funci3n q
main.cpp: In function 'int CalculaPuntosJuego(juego&)':
main.cpp:1280: aviso: converting a 'int' desde 'double'
main.cpp:1281: aviso: converting a 'int' desde 'double'
main.cpp:1283: aviso: converting a 'int' desde 'double'
main.cpp:1284: aviso: converting a 'int' desde 'double'

Compilation finished at Sun Sep 18 10:03:33

-1: ** *compilation* (Compilation: exit [0])--L11--All-----1: ** *gud* (Debugger: run)--L18383--Bot-----

main.cpp - emacs@localhost.localdomain
File Edit Options Buffers Tools C++ Help

// [HECHO] - crear codigo para menu;
// [HECHO] - implantar codigo parse;
// [HECHO] - Codigo Inicializar
// [HECHO] - codigo para instruccion CreaConfed
// [HECHO] - Crear seleccion
// [HECHO] - Crear Torneos (probar modificar torneosXo)
// [HECHO] - Crear Juegos
// [HECHO] - Implementar Funcion de Calculo puntos acumulados par
// [*] - Mostrar Puntos
// [HECHO] - Parsear y Validar Fechas
// [HECHO] - Implementar Puntos Ac y Index Puntos AC
// [HECHO] - Hacer funcion de Calculo PuntosAc
// [HECHO] - hacer algoritmo de Guardar y Actualizar PuntosAc
// [*HACER FUNCIONAR*] - Hacer Indices e implementar QuickSort[fi
// [HECHO] - Volcar Indices Ordenados a disco para uso posterior
// [HECHO*] - Imprimir datos ordenados basados en indice
//
// [HECHO] Arreglar la salida de meses
// -[HECHO] Mejor Equipo
// -[HECHO] Mejor Desempeno
// -[HECHO*] Clasificacion
// --[FIX] Arreglar orden de muestreo en GenerarReporteClasif
main.cpp (C++ RCS:1.26 Abbrev)--L22--1%
Nigeria -15 0
Espana 165 0
Bienvenido a FIRST
*****
(PiFA Ranking System)
[]
Proyecto 1: IPC 2
(c) 2005 Erik Vladimir Giron Marquez

MENU PRINCIPAL
-----
Seleccione lo que desea hacer
1. Seleccionar Archivo de Entrada...
2. Seleccionar Archivo de Salida...
3. Procesar datos y generar reporte...
[qualquier otra tecla] SALIR...

Ingrese su opcion-> 4
-1: ** *gud* (Debugger: run)--L692--98%

```



# Manual de Usuario

## Requisitos del Sistema:

- IBM-PC o compatible con Procesador de clase i386 o superior.
- Para la versión de Windows: Windows 9x/2k/XP con 2 MB libres de memoria RAM (aparte de la requerida por el sistema operativo).

(NOTA: Asegúrese de cumplir los requisitos del sistema, o el programa no correrá como debería.)

## Instrucciones de Uso:

### Pre-Pasos:

- Encienda su computador y verifique que cumple con los requisitos del sistema.
- Ingrese a su sistema operativo.

### Pasos:

1. Inserte en la unidad de discos, el disco del programa.
- Escriba desde la línea de comandos lo siguiente:
  - **d:** y luego escriba
  - **cd win32** luego
  - **first.exe**
  - (donde **d:** es la unidad donde se encuentra su unidad de discos removibles).
- O si lo prefiere, abra Mi PC, y luego abra la unidad de discos donde introdujo la unidad removible. Ya dentro, vera en su pantalla un directorio que dice "win32", haga doble clic sobre éste, luego aparecerá el icono que dice "first.exe"; Haga doble clic sobre éste para abrir el programa.
2. Aparecerá una ventana negra, en donde UD. Podrá ver el menú principal (fig 1). Podrá elegir entre las opciones presionando el número correspondiente a cada una de éstas.
3. Ya en el menú primero deberá especificar la ruta hacia el archivo de entrada (opción no 1).

```

Bienvenido a FIRST
=====
(FIFA Ranking System)

Proyecto 1: IPC 2
(c) 2005 Erik Vladimir Giron Marquez

MENU PRINCIPAL
=====
Seleccione lo que desea hacer
  1. Seleccionar Archivo de Entrada...
  2. Seleccionar Archivo de Salida...
  3. Procesar datos y generar reporte...
  [cualquier otra tecla] SALIR...

Ingrese su opcion-> █
```

Fig 1

4. Luego de haber escrito correctamente la ruta regresará a la pantalla de menú, donde también deberá especificar la ruta para el archivo de salida.
5. Ya definido la ruta hacia el archivo de entrada y el de salida UD. ya podrá entonces presionar la opción 3 del menú para iniciar el procesamiento de datos.
6. Luego de que los datos sean procesados. puede presionar cualquier otra tecla dentro del menú para salir de la aplicación. Entonces podrá dirigirse a buscar el archivo de salida, donde podrá ver los resultados de su búsqueda.

### ***Sintaxis de Archivo de Entrada:***

Los siguientes comandos son aceptados por el programa, Sin importar mayúsculas y minúsculas. Los valores que no se encuentren entre comillas son numéricos, los que sí se encuentren entre comillas son caracteres.

```
Inicializar();  
Crear_Confederacion('id','nombre',FACTOR);  
Crear_seleccion('id','nombre','id_confederacion');  
Crear_Torneos('id','nombre','es_oficial(si/no)',BONIFICACION,  
FACTOR);  
Juegos('Equipo A','Equipo B','Equipo Local',GOLESEQUIPOA,  
GOLESEQUIPOB',PENALESEQUIPOA, PENALESEQUIPOB,  
DURACION,DD,MM,AA,'id torneo');  
Mostrar_Puntos_Acumulados(AA0,MM0,AA1,MM1);  
Premio_Mejor_Equipo(AA);  
Premio_desempenio(AA);  
Clasificacion(AA,MM);
```

## Manual Técnico.

- Definición de Funciones(Mas información ver documento de Diseño y Código fuente).

**void setInput(char \*):** Establece archivo de entrada.

**void setOutput(char \*):** Establece archivo de salida.

**void msgerr(char\*,int):** Escribe mensaje o error mensaje[], a archivo de salida si tipo salida >1 o solo a stdout si tiposalida = 0.

**int ActualizaConfed(confederacion):** Actualiza archivo de Confederaciones, devuelve 0 si se creo registro, o 1 si se actualiza registro a escribir.

**int ActualizaTorneo(torneo):** Actualiza archivo de Torneos, devuelve 0 si se creo registro, o 1 si se actualiza registro a escribir o -1 si hay error.

**int ActualizaSeleccion(seleccion):** Crea o actualiza seleccion en archivo de selecciones retorna 0 si crea, 1 se actualiza.

**int GeneraReporteClasif(idxclasif \*,int):** Genera reporte de la clasificacion FIFA/CocaCola.

**int GuardarClasif(clasif):** Almacena registro clasificacion FIFA Cocacola en archivo en disco.

**int AcumularPuntosMes(clasif &,idxclasif &):** Calcula puntos acumulados del mes que se desee y del mes anterior a este del equipo especificado en la estructura.

**void InicializarIndice(idxclasif \*,int):** Inicializa Indice de Clasificacion.

**int Clasificacion(clasif &c):** Funcion principal para calcular la clasificacion fifa/cocacola.

**int buscaDesemp(desemp&):** Busca en el archivo ArchDesemp por el equipo con mayor punteo.

**int calcularPuntosGanados(desemp &):** Funcion que calcula puntos ganados para cada seleccion contenida en el registro desemp.

**int prem\_desemp(desemp &d):** Funcion principal de calculo de premio de mejor desempenio.

**int premioME(eq\_year &):** Genera reporte del premio al mejor equipo de año..

**int guardarEqA(eq\_year &):** Guarda los registros de los punteos en el a~no de cada selección en archivo de salida.

**int leerPunteoPME(eq\_year &):** Lee punteo actual del registro de Juegos para cada equipo comando premio mejor equipo.

**int eqA(eq\_year &):** Funcion principal de calculo para premio mejor equipo.

**int ValidarJuego(juego):** valida datos para juego.

**int ValidaFecha(intervfecha):** Valida fecha para Puntos Acumulados.

**int sizeoffile(FILE \*):** Calcula el tamaño del archivo dado.

**char \* mes(int):** Se ingresan numero de mes y regresa el nombre del mes.

**int printPuntosAc(intervfecha):** Imprime los puntos acumulados en pantalla y archivo.

**int CrearIndicePuntos(void):** Crea indice para registros de puntos acumulados.

**int GuardarPuntos(char\*, float):** Almacena puntos acumulados en archivo de tipo puntosac.

**int PuntosAC(intervfecha):** Calcula y escribe estructura puntos en archivo.

**int PClasif(char\*, unsigned int, int &):** Calcula puntos de clasif.

**float FNR(char\* idsel):** retorna el Factor Nivel Region respecto el id.

**CalculaPuntosJuego(juego &):** calcula puntos acumulados para cada equipo depenediendo del juego.

**int CreaJuego(juego &):** Crea en archivo binario de juego un nuevo registro.

**int menu(void):** Escribe en pantalla menu principal.

**void inicializar(void):** Inicializa todos los archivos y variables a 0.

**int cmd\_creaconfed(confederacion &):** Analiza la instruccion crear\_confederacion.

**int cmd\_creaselec(seleccion &):** parser para comando crea\_seleccion.

**int cmd\_creatorneo(torneo &):** parser para comando crear\_torneo

**int cmd\_juego(juego &):** parser para comando juego;

**int cmd\_puntosac(intervfecha &):** parser para comando puntos\_acumulados;.

**int cmd\_premio\_me(eq\_year &):** parser para comando premio\_mejor\_equipo;

**int cmd\_mejor\_desemp(desemp &):** parser para comando mejor\_desempenio.

**int cmd\_clasif(clasif &):** parser para comando clasificacion;.

**int parse(char \*):** Analiza sintacticamente el archivo de entrada y llama a funciones encargadas de cada instruccion.

**void QuickSort(idxpuntosac \*, int, int), void QuickSort(int \*,int,int), void QuickSort(idxclasif \*, int, int):** Ordena usando el algoritmo recursivo QuickSort de Hoare para ordenar punteos en memoria.

**char \* mes(int):** Se ingresan numero de mes y regresa el nombre del mes.