

MEMORIA PRACTICA FINAL

REALIZADA POR JOSUE GUTIERREZ DURAN Y RUBEN SANCHEZ SEVILLA
GRUPO G7

ANALISIS:

Programa Final

Se desea diseñar un programa que gestione los partidos de una liga de “n” equipos.

El programa debe de poder guardar los puntos de un partido individual y los puntos de todos los partidos. Además el programa debe de poder calcular la puntuación actual de un equipo en cualquier momento y calcular también la puntuación actual de todos los equipos.

Además el programa deberá de poder calcular el equipo ganador en ese momento en la liga. Además al programa también se le podrán introducir los nombres de un equipo o de todos para facilitar su identificación a la hora de usar el programa.

Otra opción que se añade al programa es la opción de borrado total, que permite eliminar todos los puntos de todos los equipos para evitar el tener que cerrar y volver a cargar el programa si se quiere gestionar una liga nueva.

Las restricciones que debe de tener el programa de gestión de la liga son las siguientes:

- Nunca se podrá introducir un punto distinto de 0, 1 o 3.
- Siempre que se de la opción sobrescribir un punto se debe de pedir una confirmación del usuario.
- En el programa, nunca se podrá introducir un equipo fuera del numero de equipos “n”.
- Siempre que se quiera ver el equipo ganador, el programa debe de dar una advertencia si no se han introducido todos los puntos.
- Si se pide que el programa calcule los puntos actualizados de uno o todos los equipos, el programa debe de dar una advertencia si no se han introducido todos los puntos.
- El programa debe de dar un aviso de sobreescritura al escribir un nombre a un equipo si dicho equipo ya tiene un nombre.
- El programa debe de pedir una confirmación al intentar borrar los datos de todos los equipos.

Análisis de Métodos:

Método principal (main) : En este método debe de imprimir el menú y leer la opción que introduzca el usuario la cual debe de estar dentro del rango de las opciones del menú (1 al 9). Si el usuario teclea una opción fuera de este rango el programa devolverá un error y volverá a pedir la lectura de una opción del menú.

Método Inicialización Resultados (void) : Este método inicializara la matriz Resultados a "-1".

Método Imprimir Menú (void) : Este método imprimirá todas las opciones del menú.

Método Comprobación Equipo (Int) : Este método comprueba si un equipo tiene un punto introducido. Este método devuelve los valores "1" (Si no hay puntos introducidos) o "0" (Si hay puntos introducidos).

Método Existencia Equipos (int) : Este método comprueba si los equipos "Casa" y "Visitante" existen (Están dentro del rango "n"). Se devuelve el valor "1", (Si ambos equipos existen), el valor "2", (Si el equipo Visitante no existe) o el valor "0" (Si el equipo Casa no existe).

Método Menú Sobreescritura (void) : Este método imprime el menú de sobre escritura, Si se elige la opción "1" los puntos serán sobrescritos, si se elige la opción "2" Los puntos serán ignorados, si se elige la opción "3" Se mostraran los puntos que hay introducidos y se volverá a pedir una nueva opción.

Método Ver Puntos de Todos los Equipos (void) : Este método imprime los puntos de todos los equipos. Si no se han introducido todos los puntos de los equipos se imprime una advertencia. Además se comprueba si los equipos tienen una etiqueta para imprimirla también.

Método Sobreescritura de Puntos (void) : Este método localiza los puntos introducidos en un partido para eliminar los puntos correspondientes a cada equipo.

Método Etiquetador Un Equipo (void) : Este método se guarda el nombre a un equipo.

Método Etiquetador Todos Equipos (void) : Este método se guardan el nombre a todos los equipos.

Método Equipo Ganador (void) : Este método imprime el nombre del equipo con mayor puntuación y los puntos que ha ganado hasta el momento.

Método Calculo de Puntos (void) : Este método calcula los puntos de un equipo.

Método Calculo de Puntos All (void) : Este método calcula los puntos de todos los equipos.

Método Error Etiquetas Equipos (void) : Este método se imprime un mensaje personalizado dependiendo de si se a introducido una etiqueta o no.

Método Escáner Puntos Etiquetas Equipos (void) : Este método imprime los puntos obtenidos por un equipos y su etiqueta si la tiene.

Método Escáner Etiqueta Equipos (void) : Este método imprime el numero del equipo y su etiqueta si la tuviera.

Método Ver los Puntos de un Equipo (void) : Este método permite ver los puntos de un equipo.

Método Sobreescritura Etiqueta (Void) : Este método imprime un menú y permite sobrecribir la etiqueta de un equipo.

Método Borrar (Void) : Este método imprime un menú y permite borrar los datos del programa.

Método Inicialización Puntos (Void) : Este método elimina todos los puntos.

Método Inicialización Etiquetas (Void) : Este método elimina todas las etiquetas.

Diseño de Métodos:

Método principal (main) : En este método se inicializan las siguientes variables:

- "n" (Int) -> Numero de equipos que juegan en la liga.
- "Resultados" (Int [][]) -> Matriz donde se almacenan los puntos de los equipos.
- "LecturaMenu" (Int) -> Variable de Lectura de la Opción del Menú.
- "Puntos" (Int []) -> Vector de Puntos Totales Actualizados.
- "EtiquetaEquipos" (String) -> String de almacenamiento de los nombres de los equipos.

Se imprimirá una cabecera y después se abrirá un bucle "do, while" que terminara con la opción 9, se leerá la opción del menú y con un "switch" se detectara que opción ha seleccionado el usuario. Si se introduce una opción fuera del menú el programa dará un error y volverá a leer una opción.

Método Inicialización Resultados (void) : En este método se inicializan las siguientes variables:

- "contador1" (Int) -> Contador de recorrido matriz Filas.
- "contador2" (Int) -> Contador de recorrido matriz Columnas.
- * "Resultados" (Int [][]) -> Se requiere la matriz Resultados del *Método principal*.

Este método esta compuesto por dos bucles "do, while" anidados en los que se recorrerá la matriz para inicializarla a "-1"

Método Imprimir Menú (void) : En este método no se usan variables. Para imprimir las opciones del menú se usara "System.out.println".

Método Comprobación Equipo (Int) : En este método se inicializan las siguientes variables:

- "contador" (Int) -> Contador de recorrido.
- * "LecturaEquipo" (Int) -> Equipo que se desea comprobar la sobreescritura, (Se requiere de otro Método).
- * "Resultados" (Int [][]) -> Se requiere la matriz Resultados del *Método principal*.
- * "n" (Int) -> Numero de Equipos, (Se requiere del *Método Principal*).

Este método esta compuesto por dos bucles "do, while" separados, uno para recorrer la matriz resultados por filas y otro para hacerlo por columnas, si en alguno de los recorridos se encuentra un "-1" el método devolverá un "1", si por el contrario el equipo tiene todos los puntos ingresados el método devolverá un "0".

Método Existencia Equipos (int) : En este método se inicializan las siguientes variables:

- * "Visitante" (Int) -> Numero del Equipo Visitante, (Se requiere de otro Método).
- * "Casa" (Int) -> Numero del Equipo Casa, (Se requiere de otro Método).
- * "n" (Int) -> Numero de Equipos, (Se requiere del *Método Principal*).

Este método comprueba si el numero de uno o dos equipos son menores que "n" con dos "if" anidados, si los dos equipos existen retorna un valor, si solo el primer equipo existe retorna un valor diferente y si ninguno de los dos equipos existe retorna un tercer valor.

Método Menú Sobreescritura (void) : En este método se inicializan las siguientes variables:

- "Menú" (Int) -> Variable de Lectura de la Opción del Menú.
- * "Puntos" (Int []) -> Vector de Puntos Totales Actualizados, (Se requiere del *Método Principal*).
- * "Visitante" (Int) -> Numero del Equipo Visitante, (Se requiere de otro Método).
- * "Casa" (Int) -> Numero del Equipo Casa, (Se requiere de otro Método).
- * "Resultados" (Int [][]) -> Se requiere la matriz Resultados del *Método principal*.
- * "EtiquetaEquipos" (String) -> String de almacenamiento de los nombres de los equipos (Se requiere del *Método Principal*).

Este método Imprime un menú de sobreescritura de puntos con tres opciones, lee la opción que elige el usuario y la ejecuta ("switch"). Si la opción que elige no es la correcta devuelve un error y vuelve a imprimir el menú para elegir una opción ("do").

Método Ver Puntos de Todos los Equipos (void) : En este método se inicializan las siguientes variables:

- "Contador" (Int) -> Recorre el numero de los equipos.
- "Comprobación" (Int) -> Comprueba si los equipos tienen todos los puntos introducidos.
- * "EtiquetaEquipos" (String) -> String de almacenamiento de los nombres de los equipos. (Se requiere del *Método Principal*).
- * "Resultados" (Int [][]) -> Se requiere la matriz Resultados del *Método principal*.
- * "n" (Int) -> Numero de Equipos, (Se requiere del *Método Principal*).
- * "Puntos" (Int []) -> Vector de Puntos Totales Actualizados, (Se requiere del *Método Principal*).

Este método esta compuesto por un bucle que se repite mientras la variable contador sea menor que el número de quipos "n". Dentro de este bucle el programa comprueba si el equipo tiene introducido todos los puntos o no e imprime un mensaje con el numero de puntos (Totales, si ha introducido todos o Actualizados si no ha introducido todos e imprime un mensaje de advertencia).

Método Sobreescritura de Puntos (void) : En este método se inicializan las siguientes variables:

- * "Puntos" (Int []) -> Vector de Puntos Totales Actualizados (Se requiere del *Método Principal*).
- * "Resultados" (Int [][]) -> Se requiere la matriz Resultados del *Método principal*.
- * "Casa" (Int) -> Numero del Equipo Casa, (Se requiere de otro Método).
- * "Visitante" (Int) -> Numero del Equipo Visitante, (Se requiere de otro Método).

Este método resta los puntos existentes en la matriz "Resultados" al equipo correspondiente en el vector "puntos", ("switch").

Método Etiquetador Un Equipo (void) : En este método se inicializan las siguientes variables:

- * "Lectura de equipo" (Int) -> Equipo que se desea etiquetar. (Se requiere de otro Método).
- * "Etiqueta Equipos" (String) -> String de almacenamiento de los nombres de los equipos (Se requiere del *Método Principal*).

Este método comprueba si el equipo introducido tiene una etiqueta, si no la tiene lee la etiqueta introducida y la asigna al equipo, si el equipo ya tiene una etiqueta pone en marcha el método de sobreescritura de etiquetas.

Método Etiquetador Todos Equipos (void) : En este método se inicializan las siguientes variables:

- * "Etiqueta Equipos" (String) -> String de almacenamiento de los nombres de los equipos (Se requiere del *Método Principal*).
- * "n" (Int) -> Numero de Equipos (Se requiere del *Método Principal*).
- "Contador" (Int) -> Recorre el numero de los equipos.

Este método esta compuesto por un bucle que repite el método Etiquetador Un Equipo un numero "n" de veces.

Método Error Etiquetas Equipos (void) : En este método se inicializan las siguientes variables:

- * "Casa" (Int) -> Numero del Equipo Casa.
- * "Etiqueta Equipos" (String) -> String de almacenamiento de los nombres de los equipos (Se requiere del *Método Principal*).

Este método comprueba si el equipo tiene una etiqueta e imprime un mensaje de error de puntos personalizado.

Método Equipo Ganador (void) : En este método se inicializan las siguientes variables:

- "Contador" (Int) -> Recorre el numero de los equipos.
- "Comprobación" (Int) -> Comprueba que los puntos de los equipos cumplen las condiciones.
- "Ganador" (Int) -> Almacena el numero del nuevo equipo ganador.
- "Mayor" (Int) -> Almacena los puntos totales del nuevo equipo ganador.
- * "EtiquetaEquipos" (String) -> String de almacenamiento de los nombres de los equipos (Se requiere del *Método Principal*).
- * "Puntos" (Int []) -> Vector de Puntos Totales Actualizados (Se requiere del *Método Principal*).
- * "n" (Int) -> Numero de Equipos (Se requiere del *Método Principal*).
- * "Resultados" (Int [][]) -> Se requiere la matriz Resultados del *Método principal*.

Este método esta compuesto por un bucle que se repite "n" veces en el que se comprueba si los puntos de un equipo son mayores que una variable "mayor" (Inicializada inicialmente a "0"), si el valor es mayor que esta variable almacena los puntos de ese equipo y la variable "ganador" almacena que equipo es el que va ganando. Después de repetir el bucle "n" veces se imprime cual es el equipo ganador y con cuantos puntos.

Método Calculo de Puntos (void) : En este método se inicializan las siguientes variables:

- * "Casa" (Int) -> Numero del Equipo Casa (Se requiere de otro Método).
- * "Visitante" (Int) -> Numero del Equipo Visitante (Se requiere de otro Método).
- * "Resultados" (Int [][]) -> Se requiere la matriz Resultados del *Método principal*.
- * "Puntos" (Int []) -> Vector de Puntos Totales Actualizados. (Se requiere del *Método Principal*).
- * "EtiquetaEquipos" (String) -> String de almacenamiento de los nombres de los equipos (Se requiere del *Método Principal*).

Este método lee los puntos del equipo "Casa" y con un switch calcula los puntos de este equipo y del visitante en el vector "Puntos".

Método Calculo de Puntos All (void) : En este método se inicializan las siguientes variables:

- "Casa" (Int) -> Numero del Equipo Casa.
- "Visitante" (Int) -> Numero del Equipo Visitante.
- * "EtiquetaEquipos" (String) -> String de almacenamiento de los nombres de los equipos (Se requiere del *Método Principal*).
- * "Resultados" (Int [][]) -> Se requiere la matriz Resultados del *Método principal*.
- * "Puntos" (Int []) -> Vector de Puntos Totales Actualizados (Se requiere del *Método Principal*).
- * "n" (Int) -> Numero de Equipos (Se requiere del *Método Principal*).

Este método va pidiendo los puntos de todos los partidos de la liga (dos bucles "do" anidados) y haciendo uso del método Calculo de Puntos se va rellenando la matriz "Resultados" y el vector "Puntos".

Método Escáner Puntos Etiquetas Equipos (void) : En este método se inicializan las siguientes variables:

- "Equipo" (Int) -> Numero del equipo.
- * "EtiquetaEquipos" (String) -> String de almacenamiento de los nombres de los equipos (Se requiere del *Método Principal*).

Este método imprime un mensaje personalizado,
(dependiendo de si el equipo tiene etiqueta o no) para introducir los puntos de un equipo.

Método Escáner Etiqueta Equipos (void) : En este método se inicializan las siguientes variables:

- "Equipo" (Int) -> Numero del equipo.
- * "EtiquetaEquipos" (String) -> String de almacenamiento de los nombres de los equipos (Se requiere del *Método Principal*).

Este método imprime un mensaje personalizado,
(dependiendo de si el equipo tiene etiqueta o no) con el numero del equipo.

Método Ver los Puntos de un Equipo (void) : En este método se inicializan las siguientes variables:

- * "EtiquetaEquipos" (String) -> String de almacenamiento de los nombres de los equipos (Se requiere del *Método Principal*).
- * "Puntos" (Int []) -> Vector de Puntos Totales Actualizados (Se requiere del *Método Principal*).
- * "n" (Int) -> Numero de Equipos (Se requiere del *Método Principal*).
- * "Lecturadeequipo" (Int) -> Equipo que se desea etiquetar.
(Se requiere de otro Método).

Este método imprime un mensaje personalizado,
(dependiendo de si el equipo tiene etiqueta o no) con los puntos del equipo.

Método Sobreescritura Etiqueta (Void) : En este método se inicializan las siguientes variables:

- * "EtiquetaEquipos" (String) -> String de almacenamiento de los nombres de los equipos (Se requiere del *Método Principal*).
- * "Equipo" (Int) -> Equipo que se desea etiquetar.
(Se requiere de otro Método).

Este método Imprime un menú de sobreescritura de etiqueta con tres opciones, lee la opción que elige el usuario y la ejecuta ("switch"). Si la opción que elige no es la correcta devuelve un error y vuelve a imprimir el menú para elegir una opción ("do").

Método Borrar (Void) : En este método se inicializan las siguientes variables:

- * "EtiquetaEquipos" (String) -> String de almacenamiento de los nombres de los equipos (Se requiere del *Método Principal*).
- * "n" (Int) -> Numero de Equipos (Se requiere del *Método Principal*).
- * "Puntos" (Int []) -> Vector de Puntos Totales Actualizados (Se requiere del *Método Principal*).
- * "Resultados" (Int[][]) -> Se requiere la matriz Resultados del *Método principal*.
- "Menú" (Int) -> Variable de Lectura de la Opción del Menú.
- "Comprobación" (Int) -> Comprueba si los equipos tienen todos los puntos introducidos.

Este método Imprime un menú de borrado con tres opciones, lee la opción que elige el usuario y la ejecuta ("switch"). Si la opción que elige no es la correcta devuelve un error y vuelve a imprimir el menú para elegir una opción ("do").

Método Inicialización Puntos (void) : En este método se inicializan las siguientes variables:

- "contador" (Int) -> Contador de recorrido el Vector.
- * "Puntos" (Int []) -> Se requiere el vector Puntos del *Método principal*.

Este método esta compuesto por un bucle "do, while" en el que se recorrerá el vector para inicializarlo a "0".

Método Inicialización Etiquetas (void) : En este método se inicializan las siguientes variables:

- "contador" (Int) -> Contador de recorrido el Vector.
- * "etiquetaequipos" (String []) -> Se requiere el String Etiqueta Equipos del *Método principal*.

Este método esta compuesto por un bucle "do, while" en el que se recorrerá el String para inicializarlo a "null".

Manual de Usuario:

Ejecución:

Para ejecutar la aplicación se deberá teclear el comando "java Liga X" siendo "X" el numero de equipos que deberá de gestionar el programa.

El programa mostrara un menú con 9 opciones que deberán de seleccionarse pulsando el numero de opción que se desea ejecutar y pulsando "Intro".

La primera opción permite introducir los puntos de un equipo, el programa pedirá que se le introduzca cual es el equipo que juega en casa y después el equipo que juega como visitante, después de esto el programa pedirá los puntos del equipo que juega en casa los cuales serán "1" en caso de empate, "0" en caso de derrota o "3" en caso de victoria.

La segunda opción pedirá que se le introduzcan todos los puntos de todos los partidos que habrá en la liga. El programa ira indicando cuales de los equipos esta jugando como local y como visitante y los puntos del equipo que hay que introducir.

¡Advertencia! Siempre que haya datos introducidos y se vayan a sobrescribir el programa dará un aviso y pedirá que el usuario sobrescriba los datos, ignore y pase a introducir otro dato o poder ver los datos existentes.

La tercera opción del programa permite ver los puntos actualizados de un equipo que se desee. Si el equipo no ha jugado todos los partidos el programa dará una advertencia y mostrara los puntos que lleva hasta el momento.

La cuarta opción mostrara los puntos de todos los equipos. Si alguno de los equipos no ha jugado todos los partidos el programa mostrara una advertencia en dicho equipo y mostrara sus puntos actualizados hasta el momento.

La quinta opción mostrara el equipo ganador de la liga, si aun no se han jugado todos los partidos el programa dará una advertencia y mostrara el equipo ganador hasta en ese momento.

La sexta opción de este programa permite asignar un nombre a un equipo concreto. Si el equipo al que se le quiere asignar el nombre ya tiene un nombre el programa te permite ver el nombre existente, ignorar el nuevo nombre o sobrescribir el nombre antiguo por uno nuevo que el usuario desee introducir.

La séptima opción permite asignar un nombre a cada equipo. Si uno de los equipos ya tiene un nombre asignado el programa te permite ver el nombre existente, ignorar el nuevo nombre y pasar a asignar un nombre al siguiente equipo de la lista o sobrescribir el nombre antiguo por uno nuevo que el usuario desee introducir.

La octava opción permite eliminar todos los datos introducidos en la liga, dentro de esta opción, un submenú permite eliminar solo los puntos de los equipos, solo los nombres asignados a cada equipo o ambas opciones a la vez. Antes de que se produzca la eliminación de los datos, el usuario deberá de confirmar que desea eliminar los datos para evitar borrados accidentales.

¡**Advertencia!** Al borrar los datos del programa solo se eliminaran los nombres y/o los puntos, pero el numero total de equipos seguirá siendo el mismo. Para cambiar el numero de equipos que gestionara la liga se deberá cerrar el programa y volver a ejecutarlo con un nuevo numero de equipos.

La novena opción permite cerrar el programa de manera segura.