



DESARROLLO DE ALGORITMOS

Trabajo Práctico Final - 2024

1. Objetivo

Todos los alumnos deben hacer el TP final en forma individual para obtener la cursada de la materia. Aquellos alumnos que están con posibilidad de promocionar deberán además **agregar la parte de promoción y defenderlo oralmente.**

2. **Modalidad Del Trabajo**: El trabajo debe subirse obligatoriamente y debe estar aprobado en un 60%. Se analizará el correcto funcionamiento del mismo y las mejores prácticas vistas en clases; **para el caso de TDA, la inclusión de métodos propios del tipo que mejoren el encapsulamiento de la clase, etc. En el caso de recorridos y búsquedas, la eficiencia de los algoritmos.**

3. **Como presentar el trabajo**: El trabajo se debe subir a la tarea en la PEDCO como último día el Lunes 27/06/2024

4. **Para promoción**, se deberá presentar **un informe** con el detalle de las **estructuras de datos utilizadas (TDAs, Arreglos, Matrices, etc)** , el **método de ordenamiento utilizado** para el inciso "[Ver Jugadores](#)", la función recursiva del inciso "[Ver estadísticas](#)" y el **resultado de las pruebas empíricas realizadas**. Se deberá defender en forma oral entre el lunes y miércoles 29/06/2024

5. **Cómo se arma la nota final en la promoción**: Estará armada por el trabajo presentado, la defensa del mismo, y los parciales, trabajos entregados en la cursada y sus intervenciones

6. Enunciado.

El torneo de fútbol del valle 300Forever nos solicita poder tener un sistema para gestionar sus campeonatos. La idea del mismo es poder registrar los equipos que compiten, con sus jugadores, el fixture del campeonato con sus distintas fechas, el armado de la tabla de posiciones y la tabla de goleadores.

Para los equipos, se posee un archivo con el nombre de los mismos, lo mismo que con los jugadores. Estos archivos deberán importarse al sistema como primera medida, de tal manera de agilizar la carga de datos.

Cada equipo tiene un nombre, una categoría (de la A a la F) y los datos asociados al torneo: puntos acumulados, partidos jugados, ganados, perdidos y empatados, goles a favor, goles en contra, diferencia de goles (goles a favor menos goles en contra). Cada partido ganado suma 3 puntos, cada partido empatado suma 1 punto, partido perdido no suma puntos.

Además, cada equipo tiene un conjunto de 11 jugadores (ver archivo adjunto jugadores.txt). Cada jugador tiene nombre, apellido, edad, dni, y número de camiseta.

Al comenzar el torneo, se debe armar el fixture de todas las fechas posibles del campeonato, para luego, en cada fecha ir ingresando los resultados de cada partido. De esta manera se pueden ir viendo los resultados de las distintas tablas.

El sistema debe tener al menos las siguientes opciones:

- **Armado de Fixture:** Se debe armar el calendario de fechas. La cantidad de fechas a jugarse tiene que ver con la cantidad de equipos que se registraron (es un torneo todos contra todos). Los equipos a enfrentarse los debe decidir el sistema en forma automática, teniendo en cuenta que no pueden dos equipos enfrentarse en dos fechas distintas. Ver [Anexo I](#)
- **Ingreso de resultados de una fecha:** Para cada fecha (se ingresan los resultados de 1 fecha, no de todas juntas), el usuario debe ingresar los resultados de todos los partidos jugados en esa fecha. Para ello, el sistema muestra para cada partido los equipos que se enfrentaron y solicita:
 - el resultado del partido: goles locales/goles visitante
 - si hubo goles de local: el número de camiseta de cada jugador que metió un gol.
 - si hubo goles de visitante: el número de camiseta de cada jugador que metió un gol.

NOTA: Al cargar los datos de resultados de cada partido, se deberá verificar que los números de camiseta ingresados corresponden con los de ese equipo.

NOTA 2: NO se debe poder ingresar los datos de una fecha más de una vez. Es decir, se debe controlar de alguna manera que no se pueda repetir la carga de datos de una fecha.

- **Agregar Jugador:** Se debe permitir al usuario agregar un jugador a un equipo. Para ello, se deben solicitar los datos personales del jugador, verificar que no exista ya en algún equipo y agregarlo al conjunto de jugadores.
- **Ver Tabla de Posiciones:** En cualquier momento se puede consultar la tabla de posiciones del torneo, donde se puede observar la información como algo similar a la siguiente, **donde los datos están ordenados en forma descendente por cantidad de puntos, y a igual cantidad de puntos, por diferencia de goles en forma descendente.**

Equipo	Pts	PJ	PG	PE	PP	GF	GC	DF
Chapecoense	22	9	7	1	1	18	12	6
Pikadura FC	20	10	6	2	2	20	10	10
Elo's	20	10	6	2	2	26	17	9
Juventud THC	19	10	6	1	3	23	14	9
Galatasaray FC	18	10	5	3	2	19	17	2
Ac.	18	10	5	3	2	17	15	2
T.v.e.t. FC	17	10	4	5	1	21	13	8
Huiliches FC	17	10	5	2	3	22	15	7
Canistuta	16	10	5	1	4	18	16	2
Almafuerte FC	15	10	4	3	3	12	7	5
Dynamo FC	14	10	4	2	4	15	16	-1

- **Ver Resultados de una fecha.** En cualquier momento se deben poder ver los resultados de una fecha determinada. Por ejemplo:

Resultado Fecha 2			
Atletico Chala	2	3	Galatasaray FC
Canistuta	0	1	Huiliches FC
Bidon FC	2	2	Pikadura FC
Elo's	2	2	I.p.c.d FC
Almafuerte FC	0	0	Los Pildoritas
T.v.e.t. FC	6	2	Golden Club FC
Nueva Chicago	2	2	Mep FA
Ac.	1	2	Marimba FC
Juventud THC	3	3	Dynamo FC

- **Ver Tabla de Goleadores:** En cualquier momento se puede consultar la tabla de goleadores del torneo, donde se puede observar la información como algo similar a la siguiente, donde los datos están ordenados en forma descendente por cantidad de goles. Utilizar para el ordenamiento el método que crea más conveniente.

Jugador	Equipo	Goles
Dalino, Patricio Julian	Elo's	12
Coceres, Diego	Marimba FC	9
Iriarte, Rodrigo Julian	Pikadura FC	9
Gonzalez, Lucas Agustin	Juventud THC	7
Torres, Matías	Huiliches FC	7
Trucco, Matias Manuel Jesus	Galatasaray FC	7
Vera Velazquez, Juan Gabriel	T.v.e.t. FC	7
Ramirez, Alexis Javier	La Caprichosa	7
Castillo, Facundo Amir	Atletico Chala	7
Cisneros, Mauro Ezequiel	Marimba FC	6

- **Ver Estadísticas:** Se desea poder calcular algunos datos del torneo:

- Edad promedio de los jugadores. Para ello, se necesita contar con una función recursiva que lo calcule.
- Cantidad de jugadores cuya edad supera en 5 la edad promedio de todos los jugadores (función anterior). Debe ser recursiva también.
(SOLO ALUMNOS QUE PROMOCIONAN)
- Dado un equipo y una edad: Apellido y Nombre del primer jugador que tenga menos de esa edad en ese equipo. Para ello, se debe solicitar al usuario el nombre del equipo y la edad. Utilizar una función recursiva para el cálculo.
- **Ver Jugadores**: Se desea poder ver la lista de jugadores ordenada por apellido y nombre. Utilizar para esta opción dos métodos de ordenamiento:
 - Método de ordenamiento más eficaz, por ejemplo MergeSort, HeapSort o QuickSort. **(SOLO ALUMNOS QUE PROMOCIONAN)**
 - Método de ordenamiento por fuerza bruta, por ejemplo Bubble, Selection, Insertion.
 - Permitir al usuario elegir con qué método ordenar.
 - Realizar pruebas empíricas de ambos métodos para demostrar la efectividad de los mismos.

ANEXO I

Se entrega el siguiente código Java con un ejemplo de como crear un Fixture. Se puede re utilizar, realizando las modificaciones correspondientes de acuerdo a las estructuras que se vayan a utilizar. En este ejemplo, los equipos son strings y la cantidad de equipos debe ser par.

```
import java.util.Arrays;
public class FixtureGenerator {
    public static void main(String[] args) {
        String[] teams = {
            "Equipo1", "Equipo2", "Equipo3", "Equipo4",
            "Equipo5", "Equipo6", "Equipo7", "Equipo8"
        };

        generateFixture(teams);
    }

    public static void generateFixture(String[] teams) {
        int cantEquipos= teams.length;

        /*Calculo cantidad de Fechas y cantidad de partidos por fecha*/
        int cantFechas= cantEquipos- 1;
        int partidosPorFecha= cantEquipos/ 2;

        /*Creo una matriz*/
        String[][] rounds = new String[cantFechas][partidosPorFecha];

        for (int fecha = 0; fecha < cantFechas; fecha++) {
            for (int partido= 0; partido< partidosPorFecha; partido++) {
                int local= (fecha + partido) % (cantEquipos - 1);
```

```
int visitante= (cantEquipos - 1 - partido + fecha ) % (cantEquipos - 1);

if (partido == 0) {
    visitante= cantEquipos- 1;
}

rounds[fecha][partido] = teams[local] + " vs " + teams[visitante];
}
}

//imprimo la matriz
printRounds(rounds);
}

public static void printRounds(String[][] rounds) {
    for (int i = 0; i < rounds.length; i++) {
        System.out.println("Round " + (i + 1));
        for (int j = 0; j < rounds[i].length; j++) {
            System.out.println(rounds[i][j]);
        }
        System.out.println();
    }
}
}
```

```
public class GeneradorFixture {

    public static void main(String[] args) {
        String[] equipos = {"1", "2", "3", "4", "5", "6", "7", "8"};
        generateFixture(equipos);
    }

    public static void generateFixture(String[] equipos) {
        int nroEquipos = equipos.length;
        int cantFechas = nroEquipos - 1;
        int partidosFecha = nroEquipos / 2;
        String[][] fechas = new String[cantFechas][partidosFecha * 2];

        for (int i = 0; i < cantFechas; i++) {
            for (int j = 0; j < partidosFecha; j++) {
                int local = (i + j) % (nroEquipos - 1);
                int visita = (nroEquipos - 1 - j + i) % (nroEquipos - 1);
                // el ultimo equipo permanece en el mismo lugar de los demás
                // rotar .
                if (j == 0) {
                    visita = nroEquipos - 1;
                }
                fechas[i][j * 2] = equipos[local];
            }
        }
    }
}
```

```
        fechas[i][j * 2 + 1] = equipos[visita];
    }
}

// intercambio el ultimo equipo para que no quede siempre de visitante
String temp;
for (int i = 0; i < fechas.length; i+=2) {
    temp = fechas[i][1];
    fechas[i][1] = fechas[i][0];
    fechas[i][0] = temp;
}

// Muestra las fechas
for (int i = 0; i < fechas.length; i++) {
    System.out.println("Fecha " + (i + 1) + ":");
    for (int j = 0; j < fechas[i].length / 2; j++) {
        System.out.println("Equipo " + fechas[i][j * 2] + " vs Equipo "
+ fechas[i][j * 2 + 1]);
    }
    System.out.println();
}
```