

Instituto Tecnológico CTC

Obligatorio Diseño y Desarrollo de Aplicaciones:

Entregado como requisito para la obtención del título Analista Programador

Alex Besozzi

Octavio Etchevarrén

Tutores: Carlos Rodríguez

Año: 2022

Presentación del problema	3
Análisis y solución	4
Alcances obtenidos	5
Diagrama de Clases	6
Código de las clases	7
Arbitro:	7
Personal:	9
DT:	10
Jugador:	11
Partido:	13
Equipo:	15
Principal:	17
Controladora:	20

Presentación del problema

En el software se deberá poder gestionar un partido de fútbol incluyendo los elementos o figuras que lo componen:

- Equipos (2), donde cada uno deberá poder ingresar:
 - 11 jugadores titulares (nombre, apellido, puesto, número, edad).
 - 5 jugadores suplentes (nombre, apellido, puesto, número, edad).
 - 1 DT (nombre, apellido, edad).
- Terna arbitral, donde deberá integrarse por:
 - 3 árbitros (nombre, apellido, puesto).
- Partido, que deberá identificarse por:
 - Lugar (estadio), día, hora, clima, -minutos jugados- (al final del partido).

Acciones requeridas, se deberá poder:

Realizar altas, bajas y cambios de jugadores durante el encuentro de un titular por un suplente (controlar que solo sea por un suplente).

Adjudicar goles a jugadores (que permita sumar de a un gol por vez).

Los números de dorsales no podrán repetirse dentro del equipo.

Al inicio del partido se deberá poder listar la nómina titular de jugadores por equipo y la de suplentes, junto con la terna arbitral y los datos del partido.

Se tiene que poder finalizar el partido.

Finalizado el mismo (de forma manual) se deberá poder volver a listar la nómina titular de jugadores por equipo y la de suplentes, debiendo verse reflejado los cambios hechos durante el partido (aquellos suplentes que ingresaron a jugar durante el partido ya no estarán en la lista de suplentes sino que estarán en la lista de titulares y aquellos titulares que fueron sustituidos durante el partido, ya no estarán en la lista de titulares).

Finalizado el encuentro se debe poder listar los goleadores.

Será realizado en Consola con posibilidad de persistir o no los datos (a elección del equipo). Si se persisten se deberá hacer en una base de datos a elección.

Análisis y solución

Analizando la letra del problema, llegamos a la conclusión de que la solución permitirá la entrada de datos de 2 equipos de fútbol, 3 árbitros y los datos del partido.

La solución permitirá realizar las acciones necesarias durante un partido:

- Marcar goles.
- Expulsar jugadores.
- Realizar cambios de jugadores
- Listar los datos del partidos y jugadores
- Listar el resultado del partido y los goleadores.

La solución se creó utilizando el lenguaje Java, con el framework IntelliJ e implementado con programación orientada a objetos.

Se crearon las clases necesarias: Jugador y DT, que heredan los datos (nombre, apellido y edad) de una clase Personal.

La clase árbitro contiene los atributos, nombre, apellido y puesto.

Creamos también una clase Equipo donde se almacenan todas los datos del equipo (id, nombre, jugadores titulares, jugadores suplentes y técnico).

La clase partido contiene los atributos, estadio, día, hora, clima y minutos jugados.

En la clase Controladora es donde desarrollamos todas las funciones necesarias para el funcionamiento del juego y por último la clase Principal donde se encuentra el menú de presentación del juego.

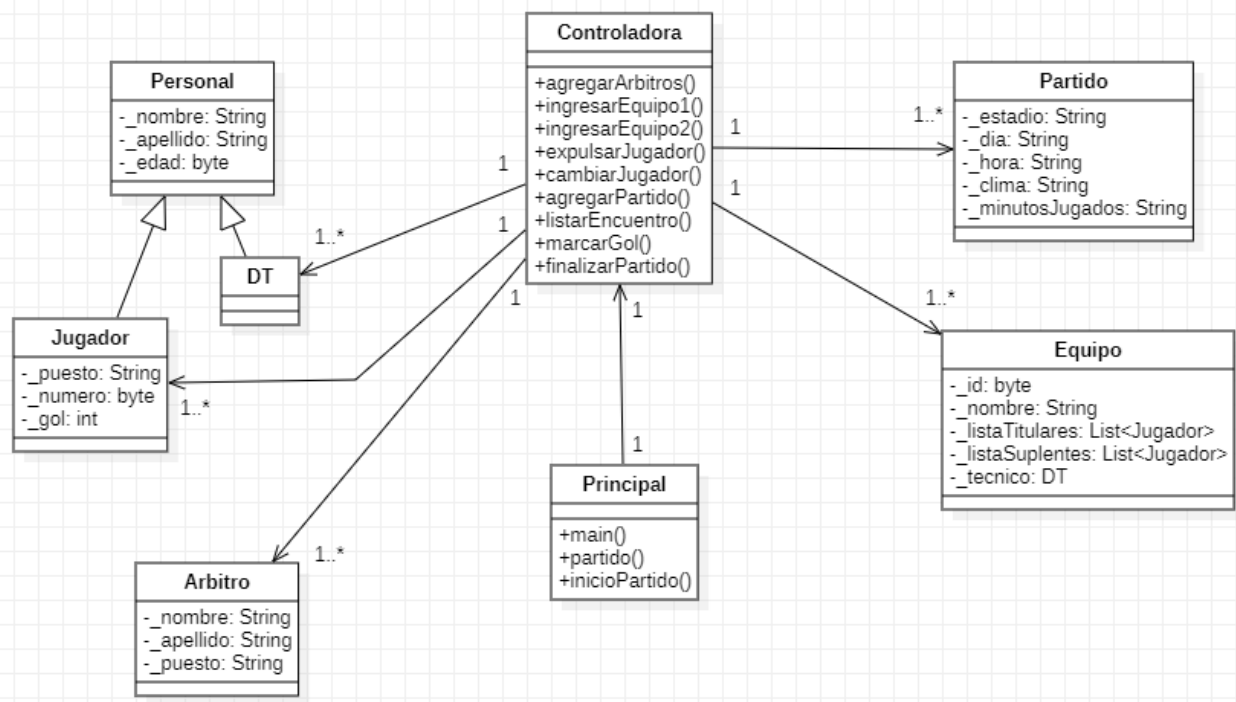
Al iniciar el juego, se solicita ingresar los nombres de los equipos. A continuación se debe ingresar, los datos del partido, la plantilla y la terna arbitral. y una vez iniciado el partido el usuario tiene 5 opciones, Listar el encuentro (1), Cambiar jugador (2), Expulsar jugador (3), Marcar gol (4) y Finalizar Partido (0).

Una vez finalizado el partido se desplegará otro menú, donde el usuario podrá ver la Lista de jugadores, Lista de goleadores o tendrá la opción de finalizar el programa.

Alcances obtenidos

	SI
En el juego	
Se puede realizar altas de jugadores respetando la cantidad.	X
Se puede realizar bajas de jugadores.	X
El partido comienza solo cuando todos los datos estén completos.	X
Se pueden hacer cambios de jugadores durante el encuentro y no antes o después.	X
Se puede adjudicar goles a jugadores de a uno, el sistema no permite adjudicar más de un gol a la vez.	X
El sistema impide que los números de dorsales al asignarlos se repitan.	X
Se pudieron agregar al inicio los elementos o figuras que componen un partido y se listan.	X
El sistema controla que los cambios se hagan solo a un titular que exista y que sean sustituidos solo por un jugador que esté en la nómina de los suplentes.	X
Se puede finalizar el partido de forma manual (ya no se pueden realizar cambios, ni anotar goles).	X
Finalizado el partido se puede volver a listar los titulares de jugadores por equipo y los suplentes, debiendo verse reflejado los cambios hechos durante el partido.	X
Se puede ver la lista de goleadores al final del partido y no antes.	X
En el trabajo global	
El código está bien comentado.	X
Usaron Git como herramienta y se puede evidenciar en GitHub que ambos trabajaron en el código.	X
Se presenta un menú claro y prolijo.	X
Se implementaron buenas prácticas de programación vistas en clase.	X
Se realizó toda la documentación.	X
Se entregó en tiempo y forma el código y toda la documentación.	X

Diagrama de Clases



Código de las clases

Arbitro:

```
package com.paquete.clases;
```

```
public class Arbitro {  
    private String _nombre;  
    private String _apellido;  
    private String _puesto;  
  
    public String get_nombre() {  
        return _nombre;  
    }  
  
    public void set_nombre(String _nombre) {  
        this._nombre = _nombre;  
    }  
  
    public String get_apellido() {  
        return _apellido;  
    }  
  
    public void set_apellido(String _apellido) {  
        this._apellido = _apellido;  
    }  
  
    public String get_puesto() {  
        return _puesto;  
    }  
  
    public void set_puesto(String _puesto) {  
        this._puesto = _puesto;  
    }  
  
    @Override  
    public String toString() {  
        return  
            "Nombre: " + _nombre + "\n" +  
            ", Apellido: " + _apellido + "\n" +  
            ", Puesto: " + _puesto + "\n";  
    }  
  
    public Arbitro(){};
```

```
public Arbitro(String pNombre, String pApellido, String pPuesto) {  
    this._nombre = pNombre;  
    this._apellido = pApellido;  
    this._puesto = pPuesto;  
}  
}
```


Personal:

```
package com.paquete.clases;
public class Personal {
    private String _nombre;
    private String _apellido;
    private byte _edad;

    public String get_nombre() {
        return _nombre;
    }

    public void set_nombre(String _nombre) {
        this._nombre = _nombre;
    }

    public String get_apellido() {
        return _apellido;
    }

    public void set_apellido(String _apellido) {
        this._apellido = _apellido;
    }

    public byte get_edad() {
        return _edad;
    }

    public void set_edad(byte _edad) {
        this._edad = _edad;
    }

    @Override
    public String toString() {
        return
            "Nombre: " + _nombre + "\n" +
            ", Apellido: " + _apellido + "\n" +
            ", Edad: " + _edad;
    }

    public Personal(){};

    public Personal(String pNombre, String pApellido, byte pEdad) {
        this._nombre = pNombre;
        this._apellido = pApellido;
        this._edad = pEdad;
    }
}
```

DT:

```
package com.paquete.clases;
```

```
public class DT extends Personal{
```

```
    @Override
```

```
    public String toString() {  
        return super.toString();  
    }
```

```
    public DT(){}
```

```
    public DT(String pNombre, String pApellido, byte pEdad) {  
        super(pNombre, pApellido, pEdad);  
    }
```

```
}
```

Jugador:

```
package com.paquete.clases;
```

```
public class Jugador extends Personal{
    private String _puesto;
    private byte _numero;

    private int _gol;

    public String get_puesto() {
        return _puesto;
    }

    public void set_puesto(String _puesto) {
        this._puesto = _puesto;
    }

    public byte get_numero() {
        return _numero;
    }

    public int get_gol() {
        return _gol;
    }

    public void set_gol(int _gol) {
        this._gol = _gol;
    }

    public void set_numero(byte _numero) {
        this._numero = _numero;
    }

    @Override
    public String toString() {
        return super.toString() +
            " Puesto: " + _puesto + "\n" +
            ", Numero: " + _numero;
    }

    public String Goleadores() {
        return super.toString() +
            " Goles: " + _gol + "\n" +
            ", Numero: " + _numero;
    }
}
```

```
public Jugador(){};
```

```
    public Jugador(String pNombre, String pApellido, byte pEdad, String pPuesto, byte  
pNumero, int pGol) {  
        super(pNombre, pApellido, pEdad);  
        this._puesto = pPuesto;  
        this._numero = pNumero;  
        this._gol =pGol;  
    }  
}
```

Partido:

```
package com.paquete.clases;
```

```
public class Partido {  
    private String _estadio;  
    private String _dia;  
    private String _hora;  
    private String _clima;  
    private String _minutosJugados;  
  
    public String get_estadio() {  
        return _estadio;  
    }  
  
    public void set_estadio(String _estadio) {  
        this._estadio = _estadio;  
    }  
  
    public String get_dia() {  
        return _dia;  
    }  
  
    public void set_dia(String _dia) {  
        this._dia = _dia;  
    }  
  
    public String get_hora() {  
        return _hora;  
    }  
  
    public void set_hora(String _hora) {  
        this._hora = _hora;  
    }  
  
    public String get_clima() {  
        return _clima;  
    }  
  
    public void set_clima(String _clima) {  
        this._clima = _clima;  
    }  
  
    public String get_minutosJugados() {  
        return _minutosJugados;  
    }  
}
```

```
public void set_minutosJugados(String _minutosJugados) {  
    this._minutosJugados = _minutosJugados;  
}
```

@Override

```
public String toString() {  
    return "Partido{" +  
        "_estadio=" + _estadio + "\" +  
        ", _dia=" + _dia + "\" +  
        ", _hora=" + _hora + "\" +  
        ", _clima=" + _clima + "\" +  
        ", _minutosJugados=" + _minutosJugados + "\" +  
        '}'";  
}
```

```
public Partido() {}
```

```
public Partido(String Estadio, String Dia, String Hora, String Clima, String  
MinutosJugados) {  
    this._estadio = Estadio;  
    this._dia = Dia;  
    this._hora = Hora;  
    this._clima = Clima;  
    this._minutosJugados = MinutosJugados;  
}  
}
```

Equipo:

```
package com.paquete.clases;  
import java.util.List;
```

```
public class Equipo {  
    private byte _id;  
    private String _nombre;  
    private List<Jugador> _listaTitulares;  
    private List<Jugador> _listaSuplentes;  
    private DT _tecnico;  
  
    public byte get_id() {  
        return _id;  
    }  
  
    public void set_id(byte _id) {  
        this._id = _id;  
    }  
  
    public String get_nombre() {  
        return _nombre;  
    }  
  
    public void set_nombre(String _nombre) {  
        this._nombre = _nombre;  
    }  
  
    public List<Jugador> get_listaTitulares() {  
        return _listaTitulares;  
    }  
  
    public void set_listaTitulares(List<Jugador> _listaTitulares) {  
        this._listaTitulares = _listaTitulares;  
    }  
  
    public List<Jugador> get_listaSuplentes() {  
        return _listaSuplentes;  
    }  
  
    public void set_listaSuplentes(List<Jugador> _listaSuplentes) {  
        this._listaSuplentes = _listaSuplentes;  
    }  
  
    public DT get_tecnico() {  
        return _tecnico;  
    }  
}
```

```
public void set_tecnico(DT _tecnico) {  
    this._tecnico = _tecnico;  
}
```

```
@Override  
public String toString() {  
    return "Equipo{" +  
        "_id=" + _id +  
        ", _nombre=" + _nombre + "\" +  
        ", _listaTitulares=" + _listaTitulares +  
        ", _listaSuplentes=" + _listaSuplentes +  
        ", _tecnico=" + _tecnico +  
        "'";  
}
```

```
public Equipo() { }
```

```
public Equipo(byte Id, String Nombre, List<Jugador> ListaTitulares, List<Jugador>  
ListaSuplentes, DT Tecnico) {  
    this._id = Id;  
    this._nombre = Nombre;  
    this._listaTitulares = ListaTitulares;  
    this._listaSuplentes = ListaSuplentes;  
    this._tecnico = Tecnico;  
}  
}
```


Principal:

```
package com.paquete;
```

```
import java.util.*;
```

```
import com.paquete.clases.*;
```

```
import com.paquete.metodos.Controladora;
```

```
public class Principal {
    static List<Arbitro> listaArbitral = new ArrayList();
    static Equipo Equipo1 = new Equipo();
    static Equipo Equipo2 = new Equipo();
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        System.out.println(" ");
        System.out.println("    Bienvenido!!!    ");
        System.out.println(" ");
        partido();
    }
    public static String equipo1 = "";
    public static String equipo2 = "";
    public static void partido() {
        Scanner scan = new Scanner(System.in);
        boolean plantilla = false;
        boolean partido = false;
        boolean arbitral = false;
        System.out.println(" ");
        System.out.println("    Por favor ingrese    ");
        System.out.println("    Equipo 1    ");
        System.out.println(" ");
        equipo1 = scan.nextLine(); // Solicitamos que ingrese el nombre del equipo 1.
        System.out.println(" ");
        System.out.println("    Por favor ingrese    ");
        System.out.println("    Equipo 2    ");
        System.out.println(" ");
        equipo2 = scan.nextLine(); // Solicitamos que ingrese el nombre del equipo 2.
        byte fin = 1;
        while (fin != 0) {
            System.out.println(" ");
            System.out.println(" 1 Agregar datos del partido ");
            System.out.println(" 2 Agregar plantilla ");
            System.out.println(" 3 Agregar arbitral ");
            System.out.println(" 0 Iniciar partido ");
            System.out.println(" ");
            byte opcion = scan.nextByte();
        }
    }
}
```

```

switch (opcion) {
    case 1:
        if (partido == false) {
            Controladora.agregarPartido(); // Invocamos el método para agregar el
partido.
            partido = true;
        } else {
            System.out.println("Los datos del partido ya fueron ingresados");
        }
        break;
    case 2:
        if (plantilla == false) {
            boolean eq1 = Controladora.ingresarEquipo1(); // Invocamos el método para
ingresar el equipo 1.
            boolean eq2 = Controladora.ingresarEquipo2(); // Invocamos el método para
ingresar el equipo 2.
            Equipo1 = Controladora.equipo1();
            Equipo2 = Controladora.equipo2();
            if (eq1 == true && eq2 == true)
                plantilla = true;
        } else {
            System.out.println("La planilla ya fue ingresada");
        }
        break;
    case 3:
        if (arbitral == false) {
            listaArbitral = Controladora.agregarArbitros(); // Invocamos el método para
agregar la terna arbitral.
            arbitral = true;
        } else {
            System.out.println("El arbitral ya fue ingresado");
        }
        break;
    case 0:
        if (plantilla == true && arbitral == true && partido == true) {
            inicioPartido(); // Invocamos el método para iniciar el partido.
            fin = 0;
        } else {
            System.out.println("Faltan datos");
        }
        break;
    default:
        System.out.println("Opción no valida");
        break;
}
}
}

```

```

public static void inicioPartido() {
    Scanner scan = new Scanner(System.in);
    byte fin = -1;
    while (fin != 0) {
        System.out.println("
        System.out.println(" 1   Listar encuentro   |");
        System.out.println(" 2   Cambio jugador   |");
        System.out.println(" 3   Expulsar jugador  |");
        System.out.println(" 4   Marcar Gol        |");
        System.out.println(" 0   Finalizar partido  |");
        System.out.println("
        byte opcion = scan.nextByte();
        switch (opcion) {
            case 1:
                Controladora.listarEncuentro(); // Invocamos el método para listar el encuentro.
                break;
            case 2:
                Controladora.cambiarJugador(); // Invocamos el método para cambiar
jugadores.
                break;
            case 3:
                Controladora.expulsarJugador(); // Invocamos el método para expulsar
jugadores.
                break;
            case 4:
                Controladora.marcarGol(); // Invocamos el método para asignar los goles.
                break;
            case 0:
                fin = 0;
                break;
            default:
                System.out.println("Opción no valida");
                break;
        }
    }
    Controladora.finalizarPartido(); // Invocamos el método para finalizar el partido.
}
}

```

Controladora:

```
package com.paquete.metodos;

import com.paquete.Principal;

import java.sql.SQLOutput;
import java.util.*;

import com.paquete.clases.*;
import jdk.swing.interop.SwingInterOpUtils;

public class Controladora {

    //region Listas e Instancias
    private static Partido Partido = new Partido(); // Instancia Partido
    private static Equipo Equipo1 = new Equipo(); // Instancia Equipo 1
    private static Equipo Equipo2 = new Equipo(); // Instancia Equipo 2
    static List<Arbitro> listaArbitral = new ArrayList(); // Lista de los árbitros
    static List<Jugador> listaJugadoresTitularesEq1 = new ArrayList(); // Lista de los
jugadores titulares del equipo 1
    static List<Jugador> listaJugadoresSuplentesEq1 = new ArrayList(); // Lista de los
jugadores suplentes del equipo 1
    static List<Jugador> listaJugadoresGoleadoresEq1 = new ArrayList(); // Lista de los
jugadores goleadores del equipo 1
    static List<Jugador> listaJugadoresTitularesEq2 = new ArrayList(); // Lista de los
jugadores titulares del equipo 2
    static List<Jugador> listaJugadoresSuplentesEq2 = new ArrayList(); // Lista de los
jugadores suplentes del equipo 2
    static List<Jugador> listaJugadoresGoleadoresEq2 = new ArrayList(); // Lista de los
jugadores goleadores del equipo 2
    private static List<Jugador> listaJugadoresExpulsadosEquipo1 = new ArrayList(); // Lista
de jugadores expulsados Equipo 1
    private static List<Jugador> listaJugadoresExpulsadosEquipo2 = new ArrayList(); // Lista
de jugadores expulsados Equipo 2
    //endregion
    public static Partido partido() {
        return Partido;
    } // Retorna el partido
    public static Equipo equipo1() {
        return Equipo1;
    } // Retorna el Equipo 1
    public static Equipo equipo2() {
        return Equipo2;
    } // Retorna el Equipo 2
```

```

public static List<Arbitro> agregarArbitros() {
    Scanner scan = new Scanner(System.in);
    System.out.println("_____");
    System.out.println(" | Ingrese los árbitros | "); // Solicita los datos necesarios para
    ingresar los 3 arbitros
    System.out.println("_____");
    for (int i = 0; i < 2; i++) {
        System.out.println("*** Nombre ***");
        String nom = scan.next();
        System.out.println("*** Apellido ***");
        String ape = scan.next();
        System.out.println("*** Puesto ***");
        String puesto = scan.next();
        Arbitro arb = new Arbitro(nom, ape, puesto); // Instancia la clase Arbitro con los
        valores ingresados
        listaArbitral.add(arb); // Añade el arbitro ingresado a la lista de arbitros
    }
    return listaArbitral; // Retorna la lista de Arbitros
}

public static boolean ingresarEquipo1() {
    String nombre = Principal.equipo1;
    byte gol = 0;
    Scanner scan = new Scanner(System.in);
    System.out.println("*** Ingrese los Datos de " + nombre + " ***"); // Solicita todos los datos
    necesarios para ingresar un equipo 1

    System.out.println("*****
    —*");

    System.out.println("*** Ingrese los datos del Director Técnico ***"); // Pide ingresar los
    datos del director técnico del equipo 1

    System.out.println("*** Nombre del director técnico ***");
    String nombreTecnico = scan.next();

    System.out.println("*** Apellido del director técnico ***");
    String apellidoTecnico = scan.next();

    System.out.println("*** Edad del director técnico ** ");
    byte edadTecnico = scan.nextByte();

    DT TecnicoEq1 = new DT(nombreTecnico, apellidoTecnico, edadTecnico); //
    Instanciamos la clase DT(Director Técnico) con los valores ingresados

    System.out.println("*****
    —*");
}

```

[illegible]

```

        System.out.println("*** Ingrese los datos de los jugadores suplentes ***"); // Solicitamos
los datos de los jugadores suplentes.
        System.out.println("");
        for (int i = 0; i < 1; i++) {
            System.out.println("*** Ingrese el nombre del jugador ***");
            String nombreSuplente = scan.next();

            System.out.println("*** Ingrese el apellido del jugador ***");
            String apellidoSuplente = scan.next();

            System.out.println("*** Ingrese la edad del jugador ***");
            byte edadSuplente = scan.nextByte();

            System.out.println("*** Ingrese el puesto del jugador ***");
            String puestoSuplente = scan.next();

            System.out.println("*** Ingrese el dorsal del jugador ***");
            byte dorsalSuplente = scan.nextByte();
            System.out.println("_____");
            boolean repetidoT = false;
            boolean repetidoS = false;
            for (Jugador unJugador : listaJugadoresTitularesEq1) { // Chequeamos que el dorsal
del jugador suplente no este repetido
                if (unJugador.get_numero() == dorsalSuplente) {
                    repetidoT = true;
                }
            }
            for (Jugador unJugador : listaJugadoresSuplentesEq1) { // Chequeamos que el
dorsal del jugador suplente no este repetido
                if (unJugador.get_numero() == dorsalSuplente) {
                    repetidoS = true;
                }
            }
            if (repetidoT == true || repetidoS == true) { // Si el dorsal está repetido, retornamos
que está en uso.
                System.out.println("Este dorsal ya esta en uso!");
                i--;
            } else { // De lo Contrario lo agregamos a la lista de jugadores suplentes del equipo 1
                Jugador jugador = new Jugador(nombreSuplente, apellidoSuplente,
edadSuplente, puestoSuplente, dorsalSuplente, gol);
                listaJugadoresSuplentesEq1.add(jugador);
            }
        }
        // Instanciamos la clase Equipo con los datos ingresados.
        Equipo team = new Equipo((byte) 1, nombre, listaJugadoresTitularesEq1,
listaJugadoresSuplentesEq1, TecnicoEq1);
        Equipo1 = team; // Le asignamos el valor del Equipo 1 instanciado a una instancia
global de la clase Equipo 1

```

```

if (TecnicoEq1 != null && listaJugadoresTitularesEq1.size() == 2 &&
listaJugadoresSuplentesEq1.size() == 1) {
    // Si las listas de jugadores titulares y la de jugadores suplentes esta completa
    retornamos, true
    return true;
}
return false; // De lo contrario retornamos, false
}

public static boolean ingresarEquipo2() {
    String nombre = Principal.equipo2;
    Scanner scan = new Scanner(System.in);
    byte gol = 0;
    System.out.println("*** Ingrese los Datos de " + nombre + " ***"); // Solicita todos los
datos necesarios para ingresar un equipo 2

System.out.println("*****
*****");

    System.out.println("*** Ingrese los datos del Director Técnico ***"); // Pide ingresar los
datos del director técnico del equipo 2

    System.out.println("*** Nombre del director técnico ***");
    String nombreTecnico = scan.next();

    System.out.println("*** Apellido del director técnico ***");
    String apellidoTecnico = scan.next();

    System.out.println("*** Edad del director técnico ***");
    byte edadTecnico = scan.nextByte();

    // Instanciamos la clase DT(Director Técnico) con los valores ingresados
    DT TecnicoEq2 = new DT(nombreTecnico, apellidoTecnico, edadTecnico);

System.out.println("*****
*****");

    System.out.println("*** Ingrese los Jugadores Titulares ***"); // Solicita ingresar los datos
de los Jugadores Titulares del equipo 2
    System.out.println("");
    for (int i = 0; i < 2; i++) {
        System.out.println("*** Ingrese el nombre del jugador ***");
        String nombreTitular = scan.next();

        System.out.println("*** Ingrese el apellido del jugador ***");
        String apellidoTitular = scan.next();

        System.out.println("*** Ingrese la edad del jugador ***");
        byte edadTitular = scan.nextByte();
    }
}

```



```

System.out.println("*** Ingrese el puesto del jugador ***");
String puestoTitular = scan.next();

System.out.println("*** Ingrese el dorsal del jugador ***");
byte dorsalTitular = scan.nextByte();
System.out.println("_____");

if (listaJugadoresTitularesEq2.size() == 0) {
    // Instanciamos los Jugadores con los datos ingresados
    Jugador jugador = new Jugador(nombreTitular, apellidoTitular, edadTitular,
    puestoTitular, dorsalTitular, gol);
    listaJugadoresTitularesEq2.add(jugador); // Añade los jugadores a la lista de
    jugadores titulares del equipo 2
} else {
    boolean repetido = false;
    for (Jugador unJugador : listaJugadoresTitularesEq2) { // Chequeamos que el
    dorsal de los jugadores no se repita
        if (unJugador.get_numero() == dorsalTitular) {
            System.out.println("Este dorsal ya esta en uso!");
            i--;
        } else {
            repetido = true;
        }
    }
    if (repetido == true) { // En caso de que no se repita el dorsal del jugador, los
    agregamos a la lista de jugadores titulares.
        Jugador jugador = new Jugador(nombreTitular, apellidoTitular, edadTitular,
        puestoTitular, dorsalTitular, gol);
        listaJugadoresTitularesEq2.add(jugador);
    }
}
}

System.out.println("*****");
System.out.println("Ingrese los datos de los jugadores suplentes"); // Solicitamos los
datos de los jugadores suplentes.
System.out.println("");
for (int i = 0; i < 1; i++) {
    System.out.println("Ingrese el nombre del jugador");
    String nombreSuplente = scan.next();

    System.out.println("Ingrese el apellido del jugador");
    String apellidoSuplente = scan.next();

    System.out.println("Ingrese la edad del jugador");
    byte edadSuplente = scan.nextByte();

```

```

System.out.println("Ingrese el puesto del jugador");
String puestoSuplente = scan.next();

System.out.println("Ingrese el dorsal del jugador");
byte dorsalSuplente = scan.nextByte();
System.out.println("_____");
boolean repetidoT = false;
boolean repetidoS = false;
for (Jugador unJugador : listaJugadoresTitularesEq2) { // Chequeamos que el dorsal
del jugador suplente no este repetido
    if (unJugador.get_numero() == dorsalSuplente) {
        repetidoT = true;
    }
}
for (Jugador unJugador : listaJugadoresSuplentesEq2) { // Chequeamos que el
dorsal del jugador suplente no este repetido
    if (unJugador.get_numero() == dorsalSuplente) {
        repetidoS = true;
    }
}
if (repetidoT == true || repetidoS == true) { // Si el dorsal está repetido, retornamos
que está en uso.
    System.out.println("Este dorsal ya esta en uso!");
    i--;
} else {
    Jugador jugador = new Jugador(nombreSuplente, apellidoSuplente,
edadSuplente, puestoSuplente, dorsalSuplente, gol);
    listaJugadoresSuplentesEq2.add(jugador); // De lo Contrario lo agregamos a la
lista de jugadores suplentes del equipo 2
}
}
// Instanciamos la clase Equipo con los datos ingresados.
Equipo team = new Equipo((byte) 1, nombre, listaJugadoresTitularesEq2,
listaJugadoresSuplentesEq2, TecnicoEq2);
Equipo2 = team; // Le asignamos el valor del Equipo 2 instanciado a una instancia
global de la clase Equipo 2

if (TecnicoEq2 != null && listaJugadoresTitularesEq2.size() == 2 &&
listaJugadoresSuplentesEq2.size() == 1) {
    return true; // Si las listas de jugadores titulares y la de jugadores suplentes esta
completa retornamos, true
}
return false; // De lo contrario retornamos, false
}
public static boolean expulsarJugador() {
    Scanner scan = new Scanner(System.in);

```

```

System.out.println("Ingrese el número del equipo, en el que se encuentra el jugador");
// Pedimos ingresar el número del equipo al que el jugador pertenece
System.out.println("1. " + equipo1().get_nombre());
System.out.println("2. " + equipo2().get_nombre());
byte opcion = scan.nextByte();

System.out.println("Ingrese el dorsal del jugador"); // Solicitamos el dorsal del jugador
que quiere expulsar
byte dorsal = scan.nextByte();

if (opcion == 1) {
    for (Jugador jugador : listaJugadoresTitularesEq1) {
        // Buscamos en la lista de jugadores titulares, el jugador que quiere expulsar
        if (jugador.get_numero() == dorsal) {
            Equipo1.get_listaTitulares().remove(jugador); // Si el jugador se encuentra en la
lista, lo removemos de la lista.
            listaJugadoresExpulsadosEquipo1.add(jugador); // Y lo añadimos en la lista de
jugadores expulsados del equipo 1.
            return true; // Por último retornamos verdadero
        } else { // En caso de que el jugador, no se encuentre en la lista de jugadores
titulares
            for (Jugador unJugador : listaJugadoresSuplentesEq1) {
                // Buscamos en la lista de jugadores suplentes
                if (unJugador.get_numero() == dorsal) {
                    Equipo1.get_listaSuplentes().remove(jugador); // Si el jugador se
encuentra en la lista, lo removemos de la lista
                    listaJugadoresExpulsadosEquipo1.add(jugador); // Y lo añadimos en la
lista de jugadores expulsados del equipo 1.
                    return true; // Por último retornamos verdadero
                }
            }
        }
    }
    return false; // Si no se cumplen ninguna de las 2 condiciones retornaremos, falso
} else if (opcion == 2) {
    for (Jugador jugador : listaJugadoresTitularesEq2) {
        // Buscamos en la lista de jugadores titulares, el jugador que quiere expulsar
        if (jugador.get_numero() == dorsal) {
            Equipo2.get_listaTitulares().remove(jugador); // Si el jugador se encuentra en la
lista, lo removemos de la lista.
            listaJugadoresExpulsadosEquipo2.add(jugador); // Y lo añadimos en la lista de
jugadores expulsados del equipo 2.
            return true; // Por último retornamos verdadero
        } else { // En caso de que el jugador, no se encuentre en la lista de jugadores
titulares
            for (Jugador unJugador : listaJugadoresSuplentesEq2) {
                // Buscamos en la lista de jugadores suplentes

```

```

if (unJugador.get_numero() == dorsal) {
    Equipo2.get_listaSuplentes().remove(jugador); // Si el jugador se
    encuentra en la lista, lo removemos de la lista
    listaJugadoresExpulsadosEquipo2.add(jugador); // Y lo añadimos en la
    lista de jugadores expulsados del equipo 2.
    return true; // Por último retornamos verdadero
}
}
}
}
return false; // Si no, se cumplen ninguna de las 2 condiciones retornaremos, falso
} else { // En caso de que el equipo ingresado no exista retornaremos, falso
    System.out.println("Ese equipo no existe");
    return false;
}
}
}

public static boolean cambiarJugador() {
    Scanner scan = new Scanner(System.in);
    System.out.println("Ingrese el número del equipo, en el que se encuentra el jugador
que desea sustituir ");
    // Pedimos ingresar el número del equipo al que el jugador pertenece
    System.out.println("1. " + equipo1().get_nombre());
    System.out.println("2. " + equipo2().get_nombre());
    byte opcion = scan.nextByte();

    // Solicitamos el número del dorsal del jugador titular
    System.out.println("Ingrese el dorsal del jugador titular");
    byte dorsalTitular = scan.nextByte();

    // Solicitamos el número del dorsal del jugador suplente
    System.out.println("Ingrese el dorsal del jugador suplente");
    byte dorsalSuplente = scan.nextByte();

    Jugador jugador = new Jugador();
    Jugador jugador = new Jugador();

    boolean jgT = false;
    boolean jgS = false;

    if (opcion == 1)
    {
        for(Jugador unJugador : listaJugadoresTitularesEq1) // Buscamos si el jugador titular
        existe.
        {

```

```

if(unJugador.get_numero() == dorsalTitular)
{
    // Si el jugador es encontrado, le asignamos los valores del jugador encontrado
a una instancia de jugador.
    jugador = unJugador;
    jgT = true;
}
}
for(Jugador unJugador : listaJugadoresSuplentesEq1) // Buscamos si el jugador
suplente existe.
{
    if (unJugador.get_numero() == dorsalSuplente)
    {
        // Si el jugador es encontrado, le asignamos los valores del jugador encontrado
a una instancia de jugador.
        jugador = unJugador;
        jgS = true;
    }
}
if (jgT == true && jgS == true) // Si el jugador titular y suplente fueron encontrados.
{
    Equipo1.get_listaTitulares().remove(jugador); // Removemos el jugador titular de la
lista de titulares
    Equipo1.get_listaTitulares().add(jugador); // Y añadimos el jugador titular a la
lista de suplentes
    Equipo1.get_listaSuplentes().remove(jugador); // Removemos el jugador suplente
de la lista de suplentes
    Equipo1.get_listaSuplentes().add(jugador); // Y añadimos el jugador suplente a
la lista de titulares
    return true; // Retornamos verdadero para finalizar
} else { // Si no encontró uno de los 2 dorsales devolverá un mensaje de error y
retornara falso
    System.out.println("Uno de los dorsal no es correcto!");
    return false;
}
}
else if(opcion == 2)
{
    for(Jugador unJugador : listaJugadoresTitularesEq2) // Buscamos si el jugador titular
existe.
    {
        if(unJugador.get_numero() == dorsalTitular)
        {
            // Si el jugador es encontrado, le asignamos los valores del jugador encontrado
a una instancia de jugador.
            jugador = unJugador;
            jgT = true;
        }
    }
}

```

```

    }
    for(Jugador unJugador : listaJugadoresSuplentesEq2) // Buscamos si el jugador
suplente existe.
    {
        if (unJugador.get_numero() == dorsalSuplente)
        {
            // Si el jugador es encontrado, le asignamos los valores del jugador encontrado
a una instancia de jugador.
            jugador = unJugador;
            jgS = true;
        }
    }
    if (jgT == true && jgS == true) // Si el jugador titular y suplente fueron encontrados.
    {
        Equipo2.get_listaTitulares().remove(jugador); // Removemos el jugador titular de la
lista de titulares
        Equipo2.get_listaTitulares().add(jugador); // Y añadimos el jugador titular a la
lista de suplentes
        Equipo2.get_listaSuplentes().remove(jugador); // Removemos el jugador suplente
de la lista de suplentes
        Equipo2.get_listaSuplentes().add(jugador); // Y añadimos el jugador suplente a
la lista de titulares
        return true; // Retornamos verdadero para finalizar
    } else { // Si no encontró uno de los 2 dorsales devolverá un mensaje de error y
retornara falso
        System.out.println("Uno de los dorsales no es correcto!");
        return false;
    }
}
}
else // Si el número que ingreso del equipo no existe, mostrara un mensaje y retornara
falso.
{
    System.out.println("Ese equipo no existe, intente nuevamente");
    return false;
}
}

```

```

public static void agregarPartido() {
    Scanner scan = new Scanner(System.in);
    System.out.println("_____");
    System.out.println("| Ingrese los datos del partido | "); // Solicita los datos necesarios
para ingresar un partido.
    System.out.println("_____");
    System.out.println("*** Ingrese el estadio donde se disputa el encuentro ***");
    String estadio = scan.nextLine();
    System.out.println("*** Ingrese el dia del encuentro ***");
    String dia = scan.next();
    System.out.println("*** Ingrese la hora del encuentro ***");
}

```

```

String hora = scan.next();
System.out.println("*** Ingrese el clima ***");
String clima = scan.next();
String minutos = "";
Partido partido = new Partido(estadio, dia, hora, clima, minutos); // Instaciamos la
clase "Partido" con los datos ingresados.

Partido = partido; // Le asignamos el partido ingresado a la instancia de partido global.
}
public static void listarEncuentro() {
    System.out.println("*** Partido: " + equipo1().get_nombre() + " vs " +
equipo2().get_nombre() + " ***");
    System.out.println("*** Estadio ***");
    System.out.println(partido().get_estadio()); // Mostramos el nombre del estadio.
    System.out.println("_____");
    System.out.println(equipo1().get_nombre());
    System.out.println("*** Titulares ***"); // Mostramos todos los jugadores titulares del
equipo 1.
    for(Jugador unJugador : listaJugadoresTitularesEq1){
        System.out.println(unJugador.toString());
    }
    System.out.println("*** Suplentes ***"); // Mostramos todos los jugadores suplentes del
equipo 1.
    for(Jugador unJugador : listaJugadoresSuplentesEq1){
        System.out.println(unJugador.toString());
    }
    System.out.println("*** Técnico ***"); // Mostramos el director técnico del equipo 1.
    System.out.println(equipo1().get_tecnico());
    System.out.println("_____");
    System.out.println(equipo2().get_nombre());
    System.out.println("*** Titulares ***"); // Mostramos todos los jugadores titulares del
equipo 2.
    for(Jugador unJugador : listaJugadoresTitularesEq2){
        System.out.println(unJugador.toString());
    }
    System.out.println("*** Suplentes ***"); // Mostramos todos los jugadores suplentes del
equipo 2.
    for(Jugador unJugador : listaJugadoresSuplentesEq2){
        System.out.println(unJugador.toString());
    }
    System.out.println("*** Técnico ***"); // Mostramos el director técnico del equipo 2.
    System.out.println(equipo2().get_tecnico());
}
public static void marcarGol() {
    Scanner scan = new Scanner(System.in);
    Jugador jugadoraso = new Jugador();
    boolean yaGol = false;
    boolean jugadorExistente = false;

```

```

        System.out.println("*** Que EQUIPO metió gol **?"); // Solicitamos que ingrese el
número del equipo, que metió gol
        System.out.println("1- " + equipo1().get_nombre());
        System.out.println("2- " + equipo2().get_nombre());
        byte equip = scan.nextByte();
        System.out.println("*** Que JUGADOR metió gol? (Número de la camiseta) ***"); //
Pedimos el dorsal del jugador que metió el gol
        byte jugadorNum = scan.nextByte();
        switch (equip){
            case 1:
                for (Jugador unJugador: listaJugadoresTitularesEq1) { // Buscamos si el jugador
ingresado existe en la lista de jugadores.
                    if(unJugador.get_numero() == jugadorNum){
                        jugadorExistente = true;
                        jugadoraso = unJugador; // Le asignamos los valores del jugador encontrado
a una instancia de jugador.
                        jugadoraso.set_gol(jugadoraso.get_gol() + 1); // Le sumamos el gol al jugador.
                    }
                }
                if(jugadorExistente == true){
                    if(listaJugadoresGoleadoresEq1.size() == 0){ // Chequeamos que la lista de
goleadores este vacía.
                        listaJugadoresGoleadoresEq1.add(jugadoraso); // Añadimos el jugador a la
lista de goleadores.
                    }
                    else{ // Si la lista no esta vacía.
                        for (Jugador Goleador : listaJugadoresGoleadoresEq1) {
                            if(Goleador.get_numero() == jugadoraso.get_numero()){ // Buscamos si el
jugador ingresado ya había metido un gol
                                yaGol = true;
                            }
                        }
                        if(yaGol == true){ // Si el jugador ya tenía goles a su favor, remplazamos los
goles.
                            for (Jugador Goleador : listaJugadoresGoleadoresEq1){
                                if(Goleador.get_numero() == jugadoraso.get_numero()){
                                    Goleador.set_gol(jugadoraso.get_gol());
                                }
                            }
                        }
                        else{ // De lo contrario, lo agregamos a la lista de goleadores.
                            listaJugadoresGoleadoresEq1.add(jugadoraso);
                        }
                    }
                }
            }
        }

```

```

else{ // Si el jugador no es encontrado, mostraremos un mensaje de error.

```



```

        System.out.println("No se a encontrado el jugador");
    }
    break;
case 2:
    for (Jugador unJugador: listaJugadoresTitularesEq2) { // Buscamos si el jugador
ingresado existe en la lista de jugadores.
        if(unJugador.get_numero() == jugadorNum){
            jugadorExistente = true;
            jugadoraso = unJugador; // Le asignamos los valores del jugador encontrado
a una instancia de jugador.
            jugadoraso.set_gol(jugadoraso.get_gol() + 1); // Le sumamos el gol al jugador.
        }
    }
    if(jugadorExistente == true){
        if(listaJugadoresGoleadoresEq2.size() == 0){ // Chequeamos que la lista de
goleadores este vacía.
            listaJugadoresGoleadoresEq2.add(jugadoraso); // Añadimos el jugador a la
lista de goleadores.
        }
        else{ // Si la lista no esta vacía.
            for (Jugador Goleador : listaJugadoresGoleadoresEq2) {
                if(Goleador.get_numero() == jugadoraso.get_numero()){ // Buscamos si el
jugador ingresado ya había metido un gol
                    yaGol = true;
                }
            }
            if(yaGol == true){ // Si el jugador ya tenía goles a su favor, remplazamos los
goles.
                for (Jugador Goleador : listaJugadoresGoleadoresEq2){
                    if(Goleador.get_numero() == jugadoraso.get_numero()){
                        Goleador.set_gol(jugadoraso.get_gol());
                    }
                }
            }
            else{ // De lo contrario, lo agregamos a la lista de goleadores.
                listaJugadoresGoleadoresEq2.add(jugadoraso);
            }
        }
    }
    else{ // Si el jugador no es encontrado, mostraremos un mensaje de error.
        System.out.println("No se a encontrado el jugador");
    }
    break;
default:
    System.out.println("Equipo invalido");
    break;
}
}
}

```

```

public static void finalizarPartido(){
    Scanner scan = new Scanner(System.in);
    // Solicitamos que ingrese los minutos, jugados del partido.
    System.out.println("Ingrese los minutos que duro el partido (Formato del ingreso de
minutos 00:00)");
    String minutos = scan.next();
    Partido.set_minutosJugados(minutos); // Cargamos los minutos jugados del partido.
    int goleseq1 = 0;
    int goleseq2 = 0;
    for(Jugador Goleador : listaJugadoresGoleadoresEq1){
        goleseq1 = goleseq1 + Goleador.get_gol();
    }
    for(Jugador Goleador : listaJugadoresGoleadoresEq2){
        goleseq2 = goleseq2 + Goleador.get_gol();
    }
    byte fin = - 1;
    System.out.println(" ");
    System.out.println(" ** Partido Finalizado ** | ");
    System.out.println(" ");
    System.out.println(equipo1().get_nombre() + " " + goleseq1 + " - " + goleseq2 + " " +
equipo2().get_nombre());
    System.out.println("Duración " + partido().get_minutosJugados());
    while(fin != 0) {
        System.out.println(" ");
        System.out.println(" 1- Listar Jugadores | ");
        System.out.println(" 2- Listar Goleadores | ");
        System.out.println(" 3- Salir | ");
        System.out.println(" ");

        byte opcion = scan.nextByte();
        switch (opcion) {
            case 1:
                System.out.println(equipo1().get_nombre());
                System.out.println("*** Titulares ***");
                for(Jugador unJugador : listaJugadoresTitularesEq1){ // Mostramos todos los
jugadores titulares del equipo 1.
                    System.out.println(unJugador.toString());
                }
                System.out.println("*** Suplentes ***");
                for(Jugador unJugador : listaJugadoresSuplentesEq1){ // Mostramos todos los
jugadores suplentes del equipo 1.
                    System.out.println(unJugador.toString());
                }
                System.out.println("*** Jugadores Expulsados ***");

                for (Jugador unJugador : listaJugadoresExpulsadosEquipo1) // Mostramos todos los
jugadores expulsados del equipo 1.
                {

```

```

        System.out.println(unJugador.toString());
    }
    System.out.println("-----");
    System.out.println(equipo2().get_nombre());
    System.out.println("*** Titulares ***");
    for(Jugador unJugador : listaJugadoresTitularesEq2){ // Mostramos todos los
jugadores titulares del equipo 2.
        System.out.println(unJugador.toString());
    }
    System.out.println("*** Suplentes ***");
    for(Jugador unJugador : listaJugadoresSuplentesEq2){ // Mostramos todos los
jugadores suplentes del equipo 2.
        System.out.println(unJugador.toString());
    }
    System.out.println("*** Jugadores Expulsados ***");
    for (Jugador unJugador : listaJugadoresExpulsadosEquipo2) // Mostramos
todos los jugadores expulsados del equipo 2.
    {
        System.out.println(unJugador.toString());
    }
    break;
case 2:
    System.out.println("*** Goleadores: " + equipo1().get_nombre() + " ***");
    for(Jugador unJugador : listaJugadoresGoleadoresEq1){ // Mostramos todos los
jugadores que anotaron gol del equipo 1.
        System.out.println(unJugador.Goleadores());
    }
    System.out.println("*** Goleadores: " + equipo2().get_nombre() + " ***");
    for(Jugador unJugador : listaJugadoresGoleadoresEq2){ // Mostramos todos los
jugadores que anotaron gol del equipo 2.
        System.out.println(unJugador.Goleadores());
    }
    break;
case 3:
    fin = 0;
    break;
default:
    System.out.println("Opción no valida");
    break;
    }
}
}
}

```