



Instituto Tecnológico CTC

**Analista Programador
Diseño y desarrollo de aplicaciones**

Obligatorio 1

Partido de Futbol

**Agustin Bartel
Agustina Bagnasco**

2022

INDICE

Letra del problema a resolver	3
Análisis y solución	4
Diagrama de clases	5
Rúbrica con puntos a evaluar	6
Código de clases	7 - 26

PROBLEMA A RESOLVER

- 1) Analizar la realidad abajo planteada y presentar una solución en Java.
- 2) Realizar el Diagrama de clases con sus métodos y atributos.
- 3) Realizar el código que una vez ejecutado permita administrar un partido de fútbol.

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, numero, edad)
 - 5 jugadores suplentes (nombre, apellido, puesto, numero, 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

ANALISIS Y SOLUCION

El programa se creó en lenguaje java, utilizando como framework IntelliJ e implementado con programacion orientada a objetos.

Se crearon las clases concretas necesarias: Jugador, Arbitro y Tecnico, las que heredan los atributos en comun (nombre y apellidos) de la clase Persona.

Creamos tambien una clase equipo que contiene la lista de jugadores titulares y suplentes además del tecnico que lo conforma.

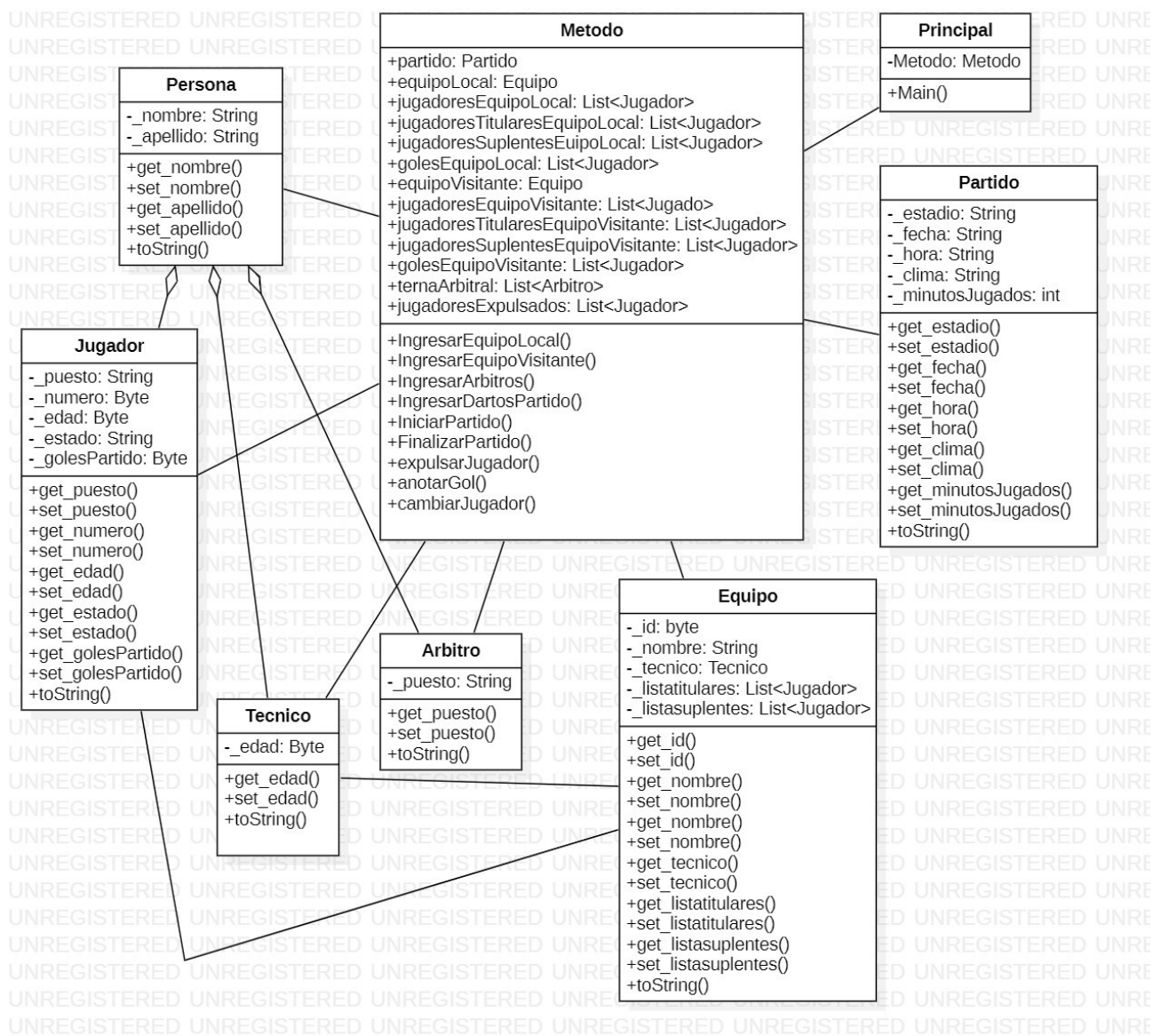
Por otra parte la clase partido tambien con lo atributos necesarios para definir las características del mismo.

En la clase Metodo que es la mas compleja desarrollamos las funciones necesarias para cumplir con los objetivos del juego y finalmente una clase Principal conteniendo el metodo Main que va a permitir al usuario acceder por consola al juego.

En principio solicitamos al usuario que ingrese los datos de los jugadores titulares y suplentes, el tecnico y los arbitros asignados asi como los datos del partido. Una vez ingresada la informacion y cumpliendo con las condiciones necesarias como la cantidad de jugadores titulares (11), los suplentes (5) y los arbitros (3). Se incia el partido mostrando los siguientes datos, la terna arbitral, los equipos y sus jugadores y los datos del partido.

Luego de iniciado el partido, el usuario tiene diferentes opciones para jugarlo: con 1- puede expulsar jugadores, con 2- anotar goles, 3 – cambiar jugadores y por ultimo con la opcion 0- finaliza el partido, donde se listaran los resultados (equipos que jugaron, jugadores expulsados, jugadores que hicieron goles, resultado del partido y minutos jugados).

DIAGRAMA DE CLASES



RÚBRICA CON PUNTOS A EVALUAR

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

CODIGO DE CLASES

Principal

```
package com.ctc;
import com.ctc.Metodos.*;
import java.util.Scanner;

public class Principal {

    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);

        Metodo.IniciarPartido();

        byte opcion;

        do {
            System.out.println("");
            System.out.println("0- Salir");
            System.out.println("1- Expulsar jugador");
            System.out.println("2- Anotar gol");
            System.out.println("3- Cambiar jugador");
            opcion = scan.nextByte();
            switch (opcion) {
                case 0:
                    Metodo.finalizarPartido();
                    break;
                case 1:
                    Metodo.expulsarJugador();
                    break;
                case 2:
                    Metodo.anotarGol();
                    break;
                case 3:
                    Metodo.cambiarJugador();
                    break;
                default:
                    System.out.println("Número fuera de rango. Ingrese otro");
            }
        }
        while (opcion > 0);
    }
}
```

Metodo

```
package com.ctc.Metodos;

import com.ctc.Clases.*;

import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

public class Metodo {
    //region Objetos y listas
    public static Partido partido = new Partido();
    public static Equipo equipoLocal = new Equipo();
    public static List<Jugador> jugadoresEquipoLocal = new ArrayList<>();
    public static List<Jugador> jugadoresTitularesEquipoLocal = new ArrayList<>();
    public static List<Jugador> jugadoresSuplentesEquipoLocal = new ArrayList<>();

    public static List<Jugador> golesEquipoLocal = new ArrayList<>();
    public static Equipo equipoVisitante = new Equipo();
    public static List<Jugador> jugadoresEquipoVisitante = new ArrayList<>();
    public static List<Jugador> jugadoresTitularesEquipoVisitante = new ArrayList<>();
    public static List<Jugador> jugadoresSuplentesEquipoVisitante = new ArrayList<>();

    public static List<Jugador> golesEquipoVisitante = new ArrayList<>();
    public static List<Arbitro> ternaArbitral = new ArrayList<>();
    public static List<Jugador> jugadoresExpulsados = new ArrayList<>();

    //endregion

    //region Métodos auxiliares equipos
    public static Jugador BuscarJugadorEquipoLocal(byte pNumero) { // Buscar jugador para verificar que el
        número ingresado sea diferente.
        for (Jugador jugador : jugadoresEquipoLocal) {
            if (jugador.get_numero() == pNumero) {
                return jugador;
            }
        }
        return null;
    }

    public static Jugador BuscarJugadorEquipoVisitante(byte pNumero) { // Buscar jugador para verificar que
        el número ingresado sea diferente.
        for (Jugador jugador : jugadoresEquipoVisitante) {
            if (jugador.get_numero() == pNumero) {
                return jugador;
            }
        }
        return null;
    }
    //endregion

    //region Ingresar equipos, arbitros y datos del partido
    public static boolean IngresarEquipoLocal() {
        Scanner scan = new Scanner(System.in);
```



```
String nombre, apellido, puesto, estado, nombreEquipo;  
byte numero, edad;
```

```
System.out.println("Ingrese nombre del equipo local"); // Pedimos los datos a ingresar.  
nombreEquipo = scan.next();
```

```
System.out.println("Ingrese nombre del DT de " + nombreEquipo);  
nombre = scan.next();
```

```
System.out.println("Ingrese apellido del DT de " + nombreEquipo);  
apellido = scan.next();
```

```
System.out.println("Ingrese edad del DT de " + nombreEquipo);  
edad = scan.nextByte();  
Tecnico dt = new Tecnico(nombre, apellido, edad);
```

```
while (jugadoresTitularesEquipoLocal.size() < 11) { // Pedimos que ingrese jugadores hasta que la lista  
de titulares tenga 11 objetos.
```

```
    System.out.println("Ingrese nombre del jugador titular de " + nombreEquipo);  
    nombre = scan.next();
```

```
    System.out.println("Ingrese apellido del jugador titular de " + nombreEquipo);  
    apellido = scan.next();
```

```
    System.out.println("Ingrese puesto del jugador titular de " + nombreEquipo);  
    puesto = scan.next();
```

```
    System.out.println("Ingrese numero del jugador titular de " + nombreEquipo);  
    numero = scan.nextByte();
```

```
    System.out.println("Ingrese edad del jugador titular de " + nombreEquipo);  
    edad = scan.nextByte();
```

```
    estado = "Titular";
```

```
    Jugador jugador = new Jugador(nombre, apellido, puesto, numero, edad, estado, (byte) 0, true);
```

```
    if (BuscarJugadorEquipoLocal(jugador.get_numero()) == null) {
```

```
        jugadoresEquipoLocal.add(jugador);
```

```
        jugadoresTitularesEquipoLocal.add(jugador);
```

```
        System.out.println("Ingresado con éxito");
```

```
    } else {
```

```
        System.out.println("No se pudo ingresar porque el número es repetido");
```

```
    }
```

```
}
```

```
while (jugadoresSuplentesEquipoLocal.size() < 5) { // Pedimos que ingrese jugadores hasta que la lista  
de suplentes tenga 5 objetos.
```

```
    System.out.println("Ingrese nombre del jugador suplente de " + nombreEquipo);  
    nombre = scan.next();
```

```
    System.out.println("Ingrese apellido del jugador suplente de " + nombreEquipo);  
    apellido = scan.next();
```

```
System.out.println("Ingrese puesto del jugador suplente de " + nombreEquipo);
puesto = scan.next();

System.out.println("Ingrese numero del jugador suplente de " + nombreEquipo);
numero = scan.nextByte();

System.out.println("Ingrese edad del jugador suplente de " + nombreEquipo);
edad = scan.nextByte();

estado = "Suplente";

Jugador jugador = new Jugador(nombre, apellido, puesto, numero, edad, estado, (byte) 0, true);
if (BuscarJugadorEquipoLocal(jugador.get_numero()) == null) { // Si el número del jugador ingresado
es diferente a cualquiera de los ya ingresados, lo agrega.
    jugadoresEquipoLocal.add(jugador);
    jugadoresSuplentesEquipoLocal.add(jugador);
    System.out.println("Ingresado con éxito");
} else {
    System.out.println("No se pudo ingresar porque el número es repetido");
}
}
equipoLocal = new Equipo((byte) 1, nombreEquipo, dt, jugadoresTitularesEquipoLocal,
jugadoresSuplentesEquipoLocal, jugadoresEquipoLocal);

if (jugadoresTitularesEquipoLocal.size() == 11 && jugadoresSuplentesEquipoLocal.size() == 5) {
    return true;
}
return false;
}
public static boolean IngresarEquipoVisitante() {
    Scanner scan = new Scanner(System.in);

    String nombre, apellido, puesto, estado, nombreEquipo;
    byte numero, edad;

    System.out.println("Ingrese nombre del equipo visitante"); // Pedimos los datos a ingresar.
    nombreEquipo = scan.next();

    System.out.println("Ingrese nombre del DT de " + nombreEquipo);
    nombre = scan.next();

    System.out.println("Ingrese apellido del DT de " + nombreEquipo);
    apellido = scan.next();

    System.out.println("Ingrese edad del DT de " + nombreEquipo);
    edad = scan.nextByte();
    Tecnico dt = new Tecnico(nombre, apellido, edad);

    while (jugadoresTitularesEquipoVisitante.size() < 11) { // Pedimos que ingrese jugadores hasta que la
lista de titulares tenga 11 objetos.
        System.out.println("Ingrese nombre del jugador titular de " + nombreEquipo);
        nombre = scan.next();

        System.out.println("Ingrese apellido del jugador titular de " + nombreEquipo);
```

```
apellido = scan.next();

System.out.println("Ingrese puesto del jugador titular de " + nombreEquipo);
puesto = scan.next();

System.out.println("Ingrese numero del jugador titular de " + nombreEquipo);
numero = scan.nextByte();

System.out.println("Ingrese edad del jugador titular de " + nombreEquipo);
edad = scan.nextByte();

estado = "Titular";

Jugador jugador = new Jugador(nombre, apellido, puesto, numero, edad, estado, (byte) 0, true);
if (BuscarJugadorEquipoVisitante(jugador.get_numero()) == null) { // Si el número del jugador
ingresado es diferente a cualquiera de los ya ingresados, lo agrega.
    jugadoresEquipoVisitante.add(jugador);
    jugadoresTitularesEquipoVisitante.add(jugador);
    System.out.println("Ingresado con éxito");
} else {
    System.out.println("No se pudo ingresar porque el número es repetido");
}
}

while (jugadoresSuplentesEquipoVisitante.size() < 5) { // Pedimos que ingrese jugadores hasta que la
lista de suplentes tenga 5 objetos.
    System.out.println("Ingrese nombre del jugador suplente de " + nombreEquipo);
    nombre = scan.next();

    System.out.println("Ingrese apellido del jugador suplente de " + nombreEquipo);
    apellido = scan.next();

    System.out.println("Ingrese puesto del jugador suplente de " + nombreEquipo);
    puesto = scan.next();

    System.out.println("Ingrese numero del jugador suplente de " + nombreEquipo);
    numero = scan.nextByte();

    System.out.println("Ingrese edad del jugador suplente de " + nombreEquipo);
    edad = scan.nextByte();

    estado = "Suplente";

    Jugador jugador = new Jugador(nombre, apellido, puesto, numero, edad, estado, (byte) 0, true);
    if (BuscarJugadorEquipoVisitante(jugador.get_numero()) == null) {
        jugadoresEquipoVisitante.add(jugador);
        jugadoresSuplentesEquipoVisitante.add(jugador);
        System.out.println("Ingresado con éxito");
    } else {
        System.out.println("No se pudo ingresar porque el número es repetido");
    }
}

equipoVisitante = new Equipo((byte) 2, nombreEquipo, dt, jugadoresTitularesEquipoVisitante,
jugadoresSuplentesEquipoVisitante, jugadoresEquipoVisitante);
```

```
        if (jugadoresTitularesEquipoVisitante.size() == 11 && jugadoresSuplentesEquipoVisitante.size() == 5) {
            return true;
        }
        return false;
    }
}
public static boolean IngresarArbitros() {
    Scanner scan = new Scanner(System.in);

    String nombre, apellido, puesto;

    for (byte i = 1; i < 4; i++) { // Pedimos que ingrese 3 arbitros
        System.out.println("Ingrese nombre del arbitro " + i);
        nombre = scan.next();

        System.out.println("Ingrese apellido del arbitro " + i);
        apellido = scan.next();

        System.out.println("Ingrese puesto del arbitro " + i);
        puesto = scan.next();

        Arbitro arbitro = new Arbitro(nombre, apellido, puesto);
        ternaArbitral.add(arbitro); // Agregamos el elemento a la lista de arbitros.
    }

    if (ternaArbitral.size() == 3) {
        return true;
    }
    return false;
}
public static Partido IngresarDatosPartido(){
    Scanner scan = new Scanner(System.in);

    String estadio, fecha, hora, clima;
    int minutos = 0;

    System.out.println("Ingrese estadio"); // Pedimos los datos.
    estadio = scan.nextLine();

    System.out.println("Ingrese fecha");
    fecha = scan.next();

    System.out.println("Ingrese hora");
    hora = scan.next();

    System.out.println("Ingrese clima");
    clima = scan.next();

    partido = new Partido(estadio, fecha, hora, clima, minutos); // Instanciamos la clase partido, y se la
    asignamos a la variable global antes declarada.

    return partido;
}
//endregion
```

```
//region Funciones del partido
public static void IniciarPartido() {
    System.out.println("Iniciando partido");
    System.out.println("");
    if (IngresarDatosPartido() != null && IngresarArbitros() && IngresarEquipoLocal() &&
    IngresarEquipoVisitante()) { // Verificamos que los datos del partido, la terna arbitral, y los datos de ambos
    equipos estén ingresados.
        System.out.println("Iniciando partido");
        System.out.println("Arbitros: ");
        for (Arbitro arbitro : ternaArbitral) { // Mostramos la terna arbitral.
            System.out.println(arbitro);
        }

        System.out.println("");

        System.out.println("Datos del partido: "); // Mostramos los datos del partido.
        System.out.println("Estadio: " + partido.get_estadio());
        System.out.println("Fecha: " + partido.get_fecha());
        System.out.println("Hora: " + partido.get_hora());
        System.out.println("Clima: " + partido.get_clima());

        System.out.println("");

        System.out.println("Equipo local: " + equipoLocal.get_nombre()); // Mostramos los datos de los
        equipos.
        System.out.println(equipoLocal.get_tecnico());

        System.out.println("Titulares: ");
        for (byte j = 0; j < equipoLocal.get_listatitulares().size(); j++) {
            System.out.println(equipoLocal.get_listatitulares().get(j).get_numero() + " - " +
            equipoLocal.get_listatitulares().get(j).get_nombre() + " - " +
            equipoLocal.get_listatitulares().get(j).get_apellido());
        }

        System.out.println("Suplentes: ");
        for (byte j = 0; j < equipoLocal.get_listasuplentes().size(); j++) {
            System.out.println(equipoLocal.get_listasuplentes().get(j).get_numero() + " - " +
            equipoLocal.get_listasuplentes().get(j).get_nombre() + " - " +
            equipoLocal.get_listasuplentes().get(j).get_apellido());
        }

        System.out.println("");

        System.out.println("Equipo visitante: " + equipoVisitante.get_nombre());
        System.out.println(equipoVisitante.get_tecnico());

        System.out.println("Titulares: ");
        for (byte j = 0; j < equipoVisitante.get_listatitulares().size(); j++) {
            System.out.println(equipoVisitante.get_listatitulares().get(j).get_numero() + " - " +
            equipoVisitante.get_listatitulares().get(j).get_nombre() + " - " +
            equipoVisitante.get_listatitulares().get(j).get_apellido());
        }
    }
```

```
        System.out.println("Suplentes: ");
        for (byte j = 0; j < equipoVisitante.get_listasuplentes().size(); j++) {
            System.out.println(equipoVisitante.get_listasuplentes().get(j).get_numero() + " - " +
equipoVisitante.get_listasuplentes().get(j).get_nombre() + " - " +
equipoVisitante.get_listasuplentes().get(j).get_apellido());
        }
    } else {
        System.out.println("No se puede iniciar el partido porque falta ingresar datos");
    }
}
public static void finalizarPartido() {
    Scanner scan = new Scanner(System.in);

    int minutos;
    int golesEquipoL = 0;
    int golesEquipoV = 0;

    System.out.println("");
    System.out.println("Ingrese los minutos jugados");
    minutos = scan.nextInt();
    partido.set_minutosJugados(minutos); // Le asignamos los minutos jugados al objeto partido.
    System.out.println("");

    System.out.println("Fin del partido");

    System.out.println("");

    System.out.println("Resultado: "); // Mostramos el resultado del partido.
    for (Jugador jugadorEquipoL : golesEquipoLocal) {
        golesEquipoL += jugadorEquipoL.getGolesPartido();
    }
    for (Jugador jugadorEquipoV : golesEquipoVisitante) {
        golesEquipoV += jugadorEquipoV.getGolesPartido();
    }
    System.out.println(equipoLocal.get_nombre() + " " + golesEquipoL + " - " + golesEquipoV + " " +
equipoVisitante.get_nombre());

    System.out.println("");

    System.out.println("Equipo local: " + equipoLocal.get_nombre()); // Mostramos los datos del equipo
local al final del partido.
    System.out.println(equipoLocal.get_tecnico());

    System.out.println("Titulares: ");
    for (byte j = 0; j < equipoLocal.get_listatitulares().size(); j++) {
        System.out.println(equipoLocal.get_listatitulares().get(j).get_numero() + " - " +
equipoLocal.get_listatitulares().get(j).get_nombre() + " - " +
equipoLocal.get_listatitulares().get(j).get_apellido());
    }

    System.out.println("Suplentes: ");
    for (byte j = 0; j < equipoLocal.get_listasuplentes().size(); j++) {
```

```
        System.out.println(equipoLocal.get_listasuplentes().get(j).get_numero() + " - " +
equipoLocal.get_listasuplentes().get(j).get_nombre() + " - " +
equipoLocal.get_listasuplentes().get(j).get_apellido());
    }

    System.out.println("");

    System.out.println("Equipo visitante: " + equipoVisitante.get_nombre()); // Mostramos los datos del
equipo visitante al final del partido.
    System.out.println(equipoVisitante.get_tecnico());

    System.out.println("Titulares: ");
    for (byte j = 0; j < equipoVisitante.get_listatitulares().size(); j++) {
        System.out.println(equipoVisitante.get_listatitulares().get(j).get_numero() + " - " +
equipoVisitante.get_listatitulares().get(j).get_nombre() + " - " +
equipoVisitante.get_listatitulares().get(j).get_apellido());
    }

    System.out.println("Suplentes: ");
    for (byte j = 0; j < equipoVisitante.get_listasuplentes().size(); j++) {
        System.out.println(equipoVisitante.get_listasuplentes().get(j).get_numero() + " - " +
equipoVisitante.get_listasuplentes().get(j).get_nombre() + " - " +
equipoVisitante.get_listasuplentes().get(j).get_apellido());
    }

    System.out.println("");

    System.out.println("Jugadores expulsados: "); // Mostramos los jugadores que fueron expulsados.
    System.out.println(jugadoresExpulsados);

    System.out.println("");

    System.out.println("Goleadores: "); // Mostramos los jugadores que hicieron uno o más goles.
    System.out.println("Equipo: " + equipoLocal.get_nombre());
    for (byte i = 0; i < golesEquipoLocal.size(); i++) {
        System.out.println("Jugador: " + golesEquipoLocal.get(i).get_numero() + " " +
golesEquipoLocal.get(i).get_nombre() + " " + golesEquipoLocal.get(i).get_apellido() + " - Goles: " +
golesEquipoLocal.get(i).getGolesPartido());
    }

    System.out.println("");

    System.out.println("Equipo: " + equipoVisitante.get_nombre());
    for (byte j = 0; j < golesEquipoVisitante.size(); j++) {
        System.out.println("Jugador: " + golesEquipoVisitante.get(j).get_numero() + " " +
golesEquipoVisitante.get(j).get_nombre() + " " + golesEquipoVisitante.get(j).get_apellido() + " - Goles: " +
golesEquipoVisitante.get(j).getGolesPartido());
    }

    System.out.println("");

    System.out.println("Minutos jugados: " + partido.get_minutosJugados()); // Mostramos los minutos
jugados.
}
```

```

public static void expulsarJugador() {
    Scanner scan = new Scanner(System.in);
    byte idequipo;
    byte numero;

    System.out.println("Ingrese id de equipo: ");
    System.out.println("1- " + equipoLocal.get_nombre());
    System.out.println("2- " + equipoVisitante.get_nombre());
    idequipo = scan.nextByte();

    if (idequipo == 1 || idequipo == 2) { // Verifica si el id del equipo es válido.
        System.out.println("Ingrese número del jugador");
        numero = scan.nextByte();

        for (byte j = 0; j < equipoLocal.get_listatitulares().size(); j++) { // Recorremos la lista de titulares del
            equipo local.
                for (byte k = 0; k < equipoLocal.get_listasuplentes().size(); k++) { // Recorremos la lista de
                    suplentes del equipo local.
                        if (idequipo == 1 && equipoLocal.get_listatitulares().get(j).get_numero() == numero) { //
                            Verificamos si el número de jugador ingresado está en la lista de titulares.
                                jugadoresExpulsados.add(equipoLocal.get_listatitulares().get(j));
                                equipoLocal.get_listatitulares().remove(equipoLocal.get_listatitulares().get(j));
                        } else if (idequipo == 1 && equipoLocal.get_listasuplentes().get(k).get_numero() == numero) { //
                            Verificamos si el número de jugador ingresado está en la lista de suplentes.
                                jugadoresExpulsados.add(equipoLocal.get_listasuplentes().get(k));
                                equipoLocal.get_listasuplentes().remove(equipoLocal.get_listasuplentes().get(k));
                        }
                    }
                }
            }

        for (byte j = 0; j < equipoVisitante.get_listatitulares().size(); j++) { // // Recorremos la lista de titulares
            del equipo visitante.
                for (byte k = 0; k < equipoVisitante.get_listasuplentes().size(); k++) { // Recorremos la lista de
                    suplentes del equipo visitante.
                        if (idequipo == 2 && equipoVisitante.get_listatitulares().get(j).get_numero() == numero) { //
                            Verificamos si el número de jugador ingresado está en la lista de titulares.
                                jugadoresExpulsados.add(equipoVisitante.get_listatitulares().get(j));
                                equipoVisitante.get_listatitulares().remove(equipoVisitante.get_listatitulares().get(j));
                        } else if (idequipo == 2 && equipoVisitante.get_listasuplentes().get(k).get_numero() == numero)
                            { // Verificamos si el número de jugador ingresado está en la lista de suplentes.
                                jugadoresExpulsados.add(equipoVisitante.get_listasuplentes().get(k));
                                equipoVisitante.get_listasuplentes().remove(equipoVisitante.get_listasuplentes().get(k));
                            }
                        }
                    }
                }
            } else {
                System.out.println("No existe el equipo"); // Si el ID del equipo ingresado no es válido, se informa del
                error.
            }
        }
    }

    public static void anotarGol() {
        Scanner scan = new Scanner(System.in);
        byte idequipo;
        byte numero;
    }

```



```

System.out.println("Ingrese id de equipo: ");
System.out.println("1- " + equipoLocal.get_nombre());
System.out.println("2- " + equipoVisitante.get_nombre());
idequipo = scan.nextByte();

if (idequipo == 1 || idequipo == 2) { // Verificamos si el ID del equipo es válido.
    System.out.println("Ingrese número del jugador");
    numero = scan.nextByte();

    for (byte i = 0; i < equipoLocal.get_listatitulares().size(); i++) { // Recorremos la lista de titulares del
equipo local.
        if (idequipo == 1 && equipoLocal.get_listatitulares().get(i).get_numero() == numero) { //
Verificamos si el número de jugador ingresado está en la lista de titulares.
            golesEquipoLocal.add(equipoLocal.get_listatitulares().get(i));
            equipoLocal.get_listatitulares().get(i).setGolesPartido((byte)
(equipoLocal.get_listatitulares().get(i).getGolesPartido() + 1));
            System.out.println("Gol de jugador " + equipoLocal.get_listatitulares().get(i).get_nombre() + " " +
equipoLocal.get_listatitulares().get(i).get_apellido() + " ingresado con éxito.");
        }
    }

    for (byte j = 0; j < equipoVisitante.get_listatitulares().size(); j++) { // Recorremos la lista de titulares del
equipo visitante.
        if (idequipo == 2 && equipoVisitante.get_listatitulares().get(j).get_numero() == numero) { //
Verificamos si el número de jugador ingresado está en la lista de titulares.
            golesEquipoVisitante.add(equipoVisitante.get_listatitulares().get(j));
            equipoVisitante.get_listatitulares().get(j).setGolesPartido((byte)
(equipoVisitante.get_listatitulares().get(j).getGolesPartido() + 1));
            System.out.println("Gol de jugador " + equipoVisitante.get_listatitulares().get(j).get_nombre() + "
" + equipoVisitante.get_listatitulares().get(j).get_apellido() + " ingresado con éxito.");
        }
    }
} else {
    System.out.println("No existe el equipo"); // Si el ID de equipo ingresado no es válido, se informa del
error.
}
}

public static void cambiarJugador(){
    Scanner scan = new Scanner(System.in);
    byte idequipo;
    byte numeroTitular;
    byte numeroSuplente;

    System.out.println("Ingrese id de equipo: ");
    System.out.println("1- " + equipoLocal.get_nombre());
    System.out.println("2- " + equipoVisitante.get_nombre());
    idequipo = scan.nextByte();

    if (idequipo == 1) { // Verificamos si el id del equipo ingresado es el del equipo local.
        System.out.println("Ingrese número del jugador titular del equipo local");
        numeroTitular = scan.nextByte();

        System.out.println("Ingrese número del jugador suplente del equipo local");
    }
}

```

```

numeroSuplente = scan.nextByte();

for (byte i = 0; i < equipoLocal.get_listatitulares().size(); i++) { // Recorremos la lista de titulares.
    for (byte j = 0; j < equipoLocal.get_listasuplentes().size(); j++) { // Recorremos la lista de suplentes.
        if (equipoLocal.get_listatitulares().get(i).get_numero() == numeroTitular &&
equipoLocal.get_listatitulares().get(i).get_cambiable()) { // Verificamos si uno de los números ingresados es el
de un jugador titular y el estado 'cambiable' es true.
            if (equipoLocal.get_listasuplentes().get(j).get_numero() == numeroSuplente &&
equipoLocal.get_listasuplentes().get(j).get_cambiable()) { // Verificamos si el otro número ingresado es el de
un jugador suplente y el estado 'cambiable' es true.
                equipoLocal.get_listatitulares().get(i).setCambiable(false); // Cambiamos el estado de
cambiable a false, esto para que no pueda volver a ingresar.
                equipoLocal.get_listasuplentes().add(equipoLocal.get_listatitulares().get(i)); // Agregamos
el jugador que ingresa a la de titulares.
                equipoLocal.get_listatitulares().remove(equipoLocal.get_listatitulares().get(i)); //
Eliminamos al jugador que sale de la lista de titulares.

                equipoLocal.get_listatitulares().add(equipoLocal.get_listasuplentes().get(j)); // Agregamos
al jugador que sale a la lista de suplentes.
                equipoLocal.get_listasuplentes().remove(equipoLocal.get_listasuplentes().get(j)); //
Eliminamos al jugador que ingresa de la lista de suplentes.
            }
        }
    }
}
} else if (idequipo == 2) { // Verificamos si el id del equipo ingresado es el del equipo visitante.
    System.out.println("Ingrese número del jugador titular del equipo visitante");
    numeroTitular = scan.nextByte();

    System.out.println("Ingrese número del jugador suplente del equipo visitante");
    numeroSuplente = scan.nextByte();

    for (byte i = 0; i < equipoVisitante.get_listatitulares().size(); i++) { // Recorremos la lista de titulares.
        for (byte j = 0; j < equipoVisitante.get_listasuplentes().size(); j++) { // Recorremos la lista de
suplentes.
            if (equipoVisitante.get_listatitulares().get(i).get_numero() == numeroTitular &&
equipoVisitante.get_listatitulares().get(i).get_cambiable()) { // Verificamos si uno de los números ingresados
es el de un jugador titular y el estado 'cambiable' es true.
                if (equipoVisitante.get_listasuplentes().get(j).get_numero() == numeroSuplente &&
equipoVisitante.get_listasuplentes().get(j).get_cambiable()) { // Verificamos si el otro número ingresado es el
de un jugador suplente y el estado 'cambiable' es true.
                    equipoVisitante.get_listatitulares().get(i).setCambiable(false); // Cambiamos el estado de
cambiable a false, esto para que no se pueda volver a cambiar por ese jugador.
                    equipoVisitante.get_listasuplentes().add(equipoVisitante.get_listatitulares().get(i)); //
Agregamos el jugador que ingresa a la de titulares.
                    equipoVisitante.get_listatitulares().remove(equipoVisitante.get_listatitulares().get(i)); //
Eliminamos al jugador que sale de la lista de titulares.

                    equipoVisitante.get_listatitulares().add(equipoVisitante.get_listasuplentes().get(j)); //
Agregamos al jugador que sale a la lista de suplentes.
                    equipoVisitante.get_listasuplentes().remove(equipoVisitante.get_listasuplentes().get(j)); //
Eliminamos al jugador que ingresa de la lista de suplentes.
                }
            }
        }
    }
}

```

```
    }  
  }  
}  
else{  
    System.out.println("No existe el equipo");  
}  
}  
//endregion  
}
```

Persona

```
package com.ctc.Clases;  
  
public class Persona {  
    private String _nombre;  
    private String _apellido;  
  
    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 Persona(String pNombre, String pApellido) {  
        _nombre = pNombre;  
        _apellido = pApellido;  
    }  
  
    public Persona(){}  
  
    @Override  
    public String toString() {  
        return "Nombre:" + _nombre + "\n" + ", Apellido: " + _apellido + "\n";  
    }  
}
```

Jugador

```
package com.ctc.Clases;

public class Jugador extends Persona{
    private String _puesto;
    private Byte _numero;
    private Byte _edad;
    private String _estado;
    private Byte _golesPartido = 0;
    private Boolean _cambiable;

    public String get_puesto() {
        return _puesto;
    }

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

    public Byte get_numero() {
        return _numero;
    }

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

    public Byte get_edad() { return _edad; }

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

    public String get_estado() { return _estado;}

    public void set_estado(String _estado) { this._estado = _estado;}

    public Byte get_golesPartido() {
        return _golesPartido;
    }

    public void set_golesPartido(Byte golesPartido) {
        this._golesPartido = golesPartido;
    }

    public Boolean get_cambiable() { return _cambiable; }

    public void setCambiable(Boolean _cambiable) { this._cambiable = _cambiable; }

    public Jugador(String pNombre, String pApellido, String pPuesto, Byte pNumero, Byte pEdad, String
pEstado, Byte pGoles, Boolean pCambiable){
        super(pNombre, pApellido);
        this._puesto = pPuesto;
    }
}
```

```
        this._numero = pNumero;
        this._edad = pEdad;
        this._estado = pEstado;
        this._golesPartido = pGoles;
        this._cambiable = pCambiable;
    }

    public Jugador(){}

    @Override
    public String toString() {
        return super.toString() +
            " Puesto=" + _puesto + "\" +
            ", Numero=" + _numero +
            ", Edad=" + _edad + "\" +
            ", Estado=" + _estado;
    }
}
```

Arbitro

```
package com.ctc.Clases;

public class Arbitro extends Persona{
    private String _puesto;

    public String get_puesto() {return _puesto;}

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

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

    public Arbitro() {}

    @Override
    public String toString() {
        return "-" + super.toString() +
            " Puesto= " + _puesto + "\"";
    }
}
```

Tecnico

```
package com.ctc.Clases;

public class Tecnico extends Persona{
    private Byte _edad;

    public Byte get_edad() {return _edad;}

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

    public Tecnico(String pNombre, String pApellido, Byte pEdad) {
        super(pNombre, pApellido);
        this._edad = pEdad;
    }

    public Tecnico(){}

    @Override
    public String toString() {
        return "DT: " +
            super.toString() +
            ", Edad=" + _edad;
    }
}
```

Equipo

```
package com.ctc.Clases;
import java.util.ArrayList;
import java.util.List;
import java.util.stream.Stream;

public class Equipo {
    private byte _id;
    private String _nombre;
    private Tecnico _tecnico;
    private List<Jugador> _listaJugadores;
    private List<Jugador> _listatitulares;
    private List<Jugador> _listasuplentes;

    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 Tecnico get_tecnico() {
    return _tecnico;
}

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

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 List<Jugador> get_listaJugadores() {
    return _listaJugadores;
}

public void set_listaJugadores(List<Jugador> _listaJugadores) {
    this._listaJugadores = _listaJugadores;
}

public Equipo(Byte pId, String pNombre, Tecnico pTecnico, List<Jugador> pListaTitulares, List<Jugador>
pListaSuplentes, List<Jugador> pListaJugadores ) {
    this._id = pId;
    this._nombre = pNombre;
    this._tecnico = pTecnico;
    this._listatitulares = pListaTitulares;
    this._listasuplentes = pListaSuplentes;
    this._listaJugadores = pListaJugadores;
}

public Equipo(){ }

@Override
public String toString() {
    return "Equipo{" +
        "_nombre=" + _nombre + "\" +
        ", _tecnico=" + _tecnico +
        ", _listatitulares=" + _listatitulares + ", _listasuplentes=" + _listasuplentes +
        "}";
}
}

```

Partido

```
package com.ctc.Clases;

public class Partido {
    private String _estadio;
    private String _fecha;
    private String _hora;
    private String _clima;
    private int _minutosJugados;
    public String get_estadio() {
        return _estadio;
    }

    public void set_estadio(String _estadio) {
        this._estadio = _estadio;
    }

    public String get_fecha() {
        return _fecha;
    }

    public void set_fecha(String _fecha) {
        this._fecha = _fecha;
    }

    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 int get_minutosJugados() {
        return _minutosJugados;
    }

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

    public Partido(String pEstadio, String pFecha, String pHora, String pClima, int pMinutosJugados) {
        this._estadio = pEstadio;
        this._fecha = pFecha;
        this._hora = pHora;
        this._clima = pClima;
    }
}
```



```
        this._minutosJugados = pMinutosJugados;
    }

    public Partido(){}

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