



# ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA



**Asignatura:**

*PROGRAMACION ORIENTADA A OBJETOS*

## **ACTIVIDAD EXPERIMENTAL N° 1**

SISTEMA DE GESTION DE PROYECTOS DE TIC

**Realizado por:**

Joan Sebasthian Clavijo Alban

Pablo Guber Camacho Bravo

Estalyn Daniel Licuy Mecías

Nayeli Scarleth Loachamin Tipan

Deisy Abigail Quillupangui Tupe

**DOCENTE:**

Luis Enrique Jaramillo Montaña

# **SISTEMA DE GESTION DE PROYECTOS DE TIC**

## **1. Introducción**

El fútbol es un deporte practicado y admirado por todo el mundo, lo que genera la necesidad de crear herramientas tecnológicas que permitan la organización de torneos. Por ello, este proyecto tiene como propósito diseñar un sistema computacional que facilite la gestión de equipos, partidos y resultados, utilizando conceptos esenciales de la POO como herencia, encapsulación y abstracción.

## **2. Objetivos**

### **Objetivo General**

Desarrollar un sistema de gestión de torneos de fútbol basado en la POO, que permita registrar equipos, programar partidos, y gestionar resultados de manera eficiente.

### **Objetivos Específicos**

1. Crear un sistema que permita registrar equipos con información de los jugadores y el entrenador.
2. Implementar la funcionalidad de creación de partidos con equipos, fechas y resultados.
3. Consultar la información registrada, incluyendo el historial de partidos jugados.
4. Desarrollar una funcionalidad para actualizar resultados de partidos y cambiar su estado de "Pendiente" a "Finalizado".

## **3. Desarrollo del Proyecto**

Metodología

### 1.- Análisis del proceso

Se identificaron las funciones necesarias para realizar el sistema, dividiéndolas en tareas específicas asignadas a todos los integrantes del grupo.

### 2.- Diseño del Sistema

Se diseñaron tres clases principales:

Equipo: Representa a los equipos registrados, incluyendo entrenador y jugadores.

Partido: Representa un encuentro entre dos equipos, con su estado y resultado.

Torneo: Administra los equipos y partidos registrados.

### 3.-Implementación del Código

Se utilizó Java para crear el sistema, aplicando conceptos de POO.

### 4.- Pruebas del Sistema

Se realizó la respectiva revisión para asegurar el correcto funcionamiento de cada funcionalidad de nuestro sistema creado.

## 4. Funcionalidades del Sistema

### 1.- Registrar Equipos

Nuestro sistema permite ingresar un equipo con:

Nombre del equipo 1: Los Tigres

Nombre del entrenador: Pedro López

Lista de jugadores:

Jugador 1: Juan Pérez (Delantero)

Jugador 2: Carlos García (Defensor)

Jugador 3: Luis Martínez (Centrocampista)

Jugador 4: Alberto Gómez (Portero)

Nombre del equipo 2: Las Águilas

Nombre del entrenador: Juan Rodríguez

Lista de jugadores:

Jugador 5: Fernando Ruiz (Portero)

Jugador 6: Miguel Gómez (Delantero)

Jugador 7: Andrés Velázquez (Defensor)

Jugador 8: Samuel Torres (Centrocampista)

## 2. Crear Partidos

Nuestro sistema registra un partido indicando:

Equipos participantes.

Fecha del partido:

Primer partido Equipo 1 VS Equipo 2 esta programado para 2024-12-15

Segundo partido Equipo 2 VS Equipo 1 esta programado para el 2024-12-20

Estado inicial ("Pendiente"). N/A, aun no se conoce el marcador.

## 3. Consultar Información

Lista de equipos registrados con sus jugadores.

Nombre del equipo 1: Los Tigres

Nombre del entrenador: Pedro López

Lista de jugadores:

Jugador 1: Juan Pérez (Delantero)

Jugador 2: Carlos García (Defensor)

Jugador 3: Luis Martínez (Centrocampista)

Jugador 4: Alberto Gómez (Portero)

Nombre del equipo 2: Las Águilas

Nombre del entrenador: Juan Rodríguez

Lista de jugadores:

Jugador 5: Fernando Ruiz (Portero)

Jugador 6: Miguel Gómez (Delantero)

Jugador 7: Andrés Velázquez (Defensor)

Jugador 8: Samuel Torres (Centrocampista)

Historial de partidos, incluyendo resultados.

Los Tigres VS Las Águilas

Fecha: 2024-12-15

Estado: Pendiente

Resultado: N/A

Las Águilas VS Los Tigres

Fecha: 2024-12-20

Estado: Pendiente

Resultado: N/A

#### **4. Actualizar Resultados**

Permite cambiar el estado de un partido de "Pendiente" a "Finalizado" y registrar el marcador.

Torneo 1 resultado 0, 3-1

Torneo 2 resultado: 0, 2-2

#### **5. Resultados**

1. Se desarrolló un sistema funcional que permite

Registrar equipos con sus datos básicos: Nombres de los equipos 1 y 2

Crear y gestionar partidos: Torneo 1 y Torneo 2

Consultar información sobre equipos y partidos registrados: Marcador del resultado del partido

Actualizar resultados y estados de los partidos: Finalizo en ganar o perder.

2. Prueba del Sistema



Equipo 1: Los Tigres, Entrenador: Pedro López, jugadores: Juan, Carlos, Luis, Alberto.

Equipo 2: Las Águilas, Entrenador: Juan Rodríguez, jugadores: Fernando, Miguel, Andrés, Samuel.

Historial de partidos registrados:

2024-12-15 - Tigres vs Águilas - Estado: Pendiente, Resultado: N/A

2024-12-20 – Águilas vs Tigres - Estado: Pendiente, Resultado: N/A

Actualización de resultados:

2024-12-15 - Tigres vs Águilas - Estado: Finalizado, Resultado: 3-1

2024-12-20 – Águilas vs Tigres - Estado: Finalizado, Resultado: 2-2

## 6. Conclusión

Este proyecto nos permitió comprender cómo aplicar los principios de POO para resolver problemas prácticos en la programación. La implementación del sistema cumplió con los objetivos planteados, ofreciendo una solución funcional y escalable para la gestión de torneos de fútbol.

Además, trabajar en equipo nos ayudó a distribuir responsabilidades, fomentar la colaboración y garantizar la calidad del producto final.

## 7. Anexos

### Diagrama UML: Gestión de Torneos

#### Clase: Equipo

+ nombre : String

+ entrenador : String

+ jugadores : ArrayList<String>

-----

+ Equipo(nombre: String, entrenador: String)

+ agregarJugador(jugador: String) : void

+ mostrarEquipo() : void

### **Clase: Partido**

+ equipo1 : String

+ equipo2 : String

+ fecha : String

+ resultado : String

-----

+ Partido(equipo1: String, equipo2: String, fecha: String, resultado: String)

+ mostrarPartido() : void

Clase: GestionTorneos

+ equipos : ArrayList<Equipo>

+ partidos : ArrayList<Partido>

-----

+ GestionTorneos()

+ registrarEquipo(equipo: Equipo) : void

+ registrarPartido(partido: Partido) : void

+ listarEquipos() : void

+ listarPartidos() : void

### **Relaciones**

- GestionTorneos "contiene" 0..\* Equipo

- GestionTorneos "contiene" 0..\* Partido

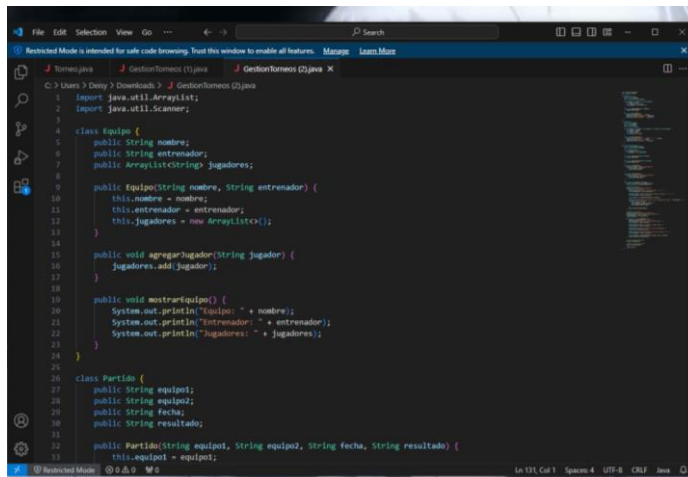
### **Codigo implementado**

<https://uceedu->

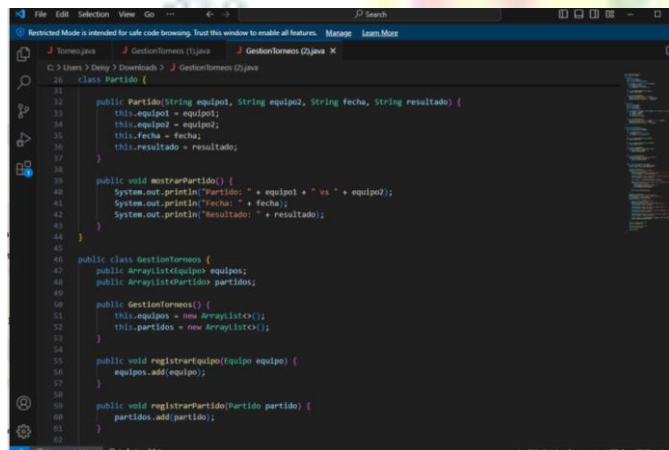
[my.sharepoint.com/:u:/g/personal/lfquillupangui\\_uce\\_edu\\_ec/EToaNI9ZY6xGjo](https://my.sharepoint.com/:u:/g/personal/lfquillupangui_uce_edu_ec/EToaNI9ZY6xGjo)

[Y2vUK4OFMBZByv9xusNgH2bVN94WosnQ?e=tndcJ4](https://my.sharepoint.com/:u:/g/personal/lfquillupangui_uce_edu_ec/EToaNI9ZY6xGjoY2vUK4OFMBZByv9xusNgH2bVN94WosnQ?e=tndcJ4)

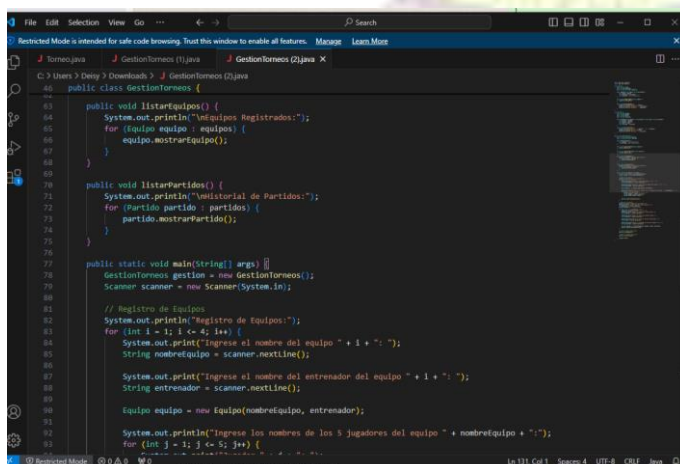
## 8. Evidencias



```
1 import java.util.ArrayList;
2 import java.util.Scanner;
3
4 class Equipos {
5     public String nombre;
6     public String entrenador;
7     public ArrayList<String> jugadores;
8
9     public Equipos(String nombre, String entrenador) {
10         this.nombre = nombre;
11         this.entrenador = entrenador;
12         this.jugadores = new ArrayList<>();
13     }
14
15     public void agregarJugador(String jugador) {
16         jugadores.add(jugador);
17     }
18
19     public void mostrarEquipos() {
20         System.out.println("Equipos: " + nombre);
21         System.out.println("Entrenador: " + entrenador);
22         System.out.println("Jugadores: " + jugadores);
23     }
24 }
25
26 class Partido {
27     public String equipo1;
28     public String equipo2;
29     public String fecha;
30     public String resultado;
31
32     public Partido(String equipo1, String equipo2, String fecha, String resultado) {
33         this.equipo1 = equipo1;
```



```
34         this.equipo2 = equipo2;
35         this.fecha = fecha;
36         this.resultado = resultado;
37     }
38
39     public void mostrarPartido() {
40         System.out.println("Partido: " + equipo1 + " vs " + equipo2);
41         System.out.println("Fecha: " + fecha);
42         System.out.println("Resultado: " + resultado);
43     }
44 }
45
46 public class GestionTorneos {
47     public ArrayList<Equipos> equipos;
48     public ArrayList<Partido> partidos;
49
50     public GestionTorneos() {
51         this.equipos = new ArrayList<>();
52         this.partidos = new ArrayList<>();
53     }
54
55     public void registrarEquipos(Equipos equipo) {
56         equipos.add(equipo);
57     }
58
59     public void registrarPartido(Partido partido) {
60         partidos.add(partido);
61     }
62 }
```



```
63
64     public void listarEquipos() {
65         System.out.println("Equipos Registrados:");
66         for (Equipos equipo : equipos) {
67             equipo.mostrarEquipos();
68         }
69     }
70
71     public void listarPartidos() {
72         System.out.println("Historial de Partidos:");
73         for (Partido partido : partidos) {
74             partido.mostrarPartido();
75         }
76     }
77
78     public static void main(String[] args) {
79         GestionTorneos gestion = new GestionTorneos();
80         Scanner scanner = new Scanner(System.in);
81
82         // Registro de Equipos
83         System.out.println("Registro de Equipos:");
84         for (int i = 1; i <= 4; i++) {
85             System.out.print("Ingrese el nombre del equipo " + i + ": ");
86             String nombreEquipo = scanner.nextLine();
87
88             System.out.print("Ingrese el nombre del entrenador del equipo " + i + ": ");
89             String entrenador = scanner.nextLine();
90
91             Equipos equipo = new Equipos(nombreEquipo, entrenador);
92             gestion.registrarEquipos(equipo);
93         }
94         System.out.println("Ingrese los nombres de los 5 jugadores del equipo " + nombreEquipo + ":");
95         for (int j = 1; j <= 5; j++) {
96             System.out.print("Ingrese el nombre del jugador " + j + ": ");
97             String nombreJugador = scanner.nextLine();
98             gestion.agregarJugador(nombreEquipo, nombreJugador);
99         }
100     }
101 }
```



```
File Edit Selection View Go ... Search
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

Torneos.java GestionTorneos (1).java GestionTorneos (2).java X
C:\Users\> D:\Users\> Downloads\> GestionTorneos (2).java
66 public class GestionTorneos {
67     public static void main(String[] args) {
68         Scanner scanner = new Scanner(System.in);
69         // Limpiar Buffer
70
71         for (int i = 1; i <= numPartidos; i++) {
72             System.out.println("Partido " + i + ":");
73             System.out.print("Ingresar el nombre del primer equipo: ");
74             String equipo1 = scanner.nextLine();
75
76             System.out.print("Ingresar el nombre del segundo equipo: ");
77             String equipo2 = scanner.nextLine();
78
79             System.out.print("Ingresar la fecha del partido (dd/mm/aaaa): ");
80             String fecha = scanner.nextLine();
81
82             System.out.print("Ingresar el resultado del partido (Ejemplo: 2-1): ");
83             String resultado = scanner.nextLine();
84
85             Partido partido = new Partido(equipo1, equipo2, fecha, resultado);
86             gestion.registrarPartido(partido);
87         }
88
89         // Mostrar Equipos Registrados
90         gestion.listarEquipos();
91
92         // Mostrar Partidos Registrados
93         gestion.listarPartidos();
94
95         scanner.close();
96     }
97 }
```

