



# ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS

INNOVACIÓN PARA LA EXCELENCIA



## Actividad Experimental N.º 1 Primer Parcial

**Tema de la Actividad:** Sistema de Gestión de Proyectos de TIC

**Tipo de Actividad:** Desarrollo Práctico

**Asignatura:** Programación Orientada a Objetos

**NRC:** 1322

**Fecha de entrega:** 07/12/2024

### Integrantes:

- Pamela Inés Guapulema Pichamba
- Jesús Ismael Pichucho Montes
- Isaac Sebastian Proaño Panchi
- Karol Elizabeth Pugachi Suarez

**Objetivo de la actividad:**

Realizar un sistema de gestión de torneos de fútbol utilizando Programación Orientada a Objetos, permitiendo las funcionalidades registrar equipos, crear partidos, consultar su información y a su vez actualizar resultados, esto se visualizará por medio de un diagrama UML y de la simulación de código Java.

**Descripción de la Actividad:**

Implemente un sistema de gestión de proyectos utilizando POO. Incluya las funcionalidades de registrar proyectos, asignar tareas y mostrar su estado.

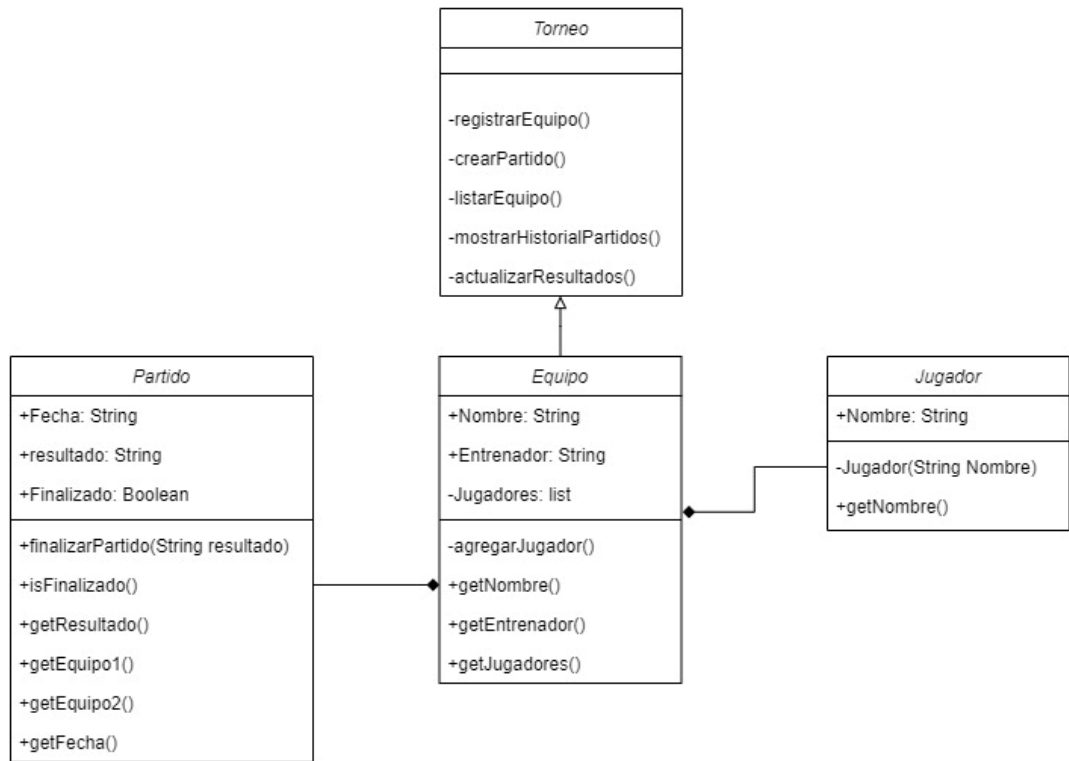
**Ejemplo:**

Implemente un sistema de gestión de torneos de fútbol utilizando Programación Orientada a Objetos (POO). El sistema debe permitir las siguientes funcionalidades:

- Registrar Equipos:
  - o Cada equipo debe tener un nombre, un entrenador, y una lista de jugadores.
- Crear Partidos:
  - o Registrar los equipos que jugarán un partido, la fecha del encuentro y el resultado final.
- Consultar Información:
  - o Listar todos los equipos registrados con sus respectivos jugadores.
  - o Mostrar el historial de partidos jugados, incluyendo los equipos participantes y los resultados.
- Actualizar Resultados:
  - o Cambiar el estado de un partido de "Pendiente" a "Finalizado", añadiendo el marcador.

## Desarrollo de la Actividad:

### 1. Diagrama UML



### 2. Código de las clases

#### 1. Equipo.java

```
public class Equipo {
    private String nombre;

    private String entrenador;

    private List<Jugador> jugadores;

    public Equipo(String nombre, String entrenador) {

        this.nombre = nombre;

        this.entrenador = entrenador;

        this.jugadores = new ArrayList<>();
```

```
}
```

```
public void agregarJugador(Jugador jugador) {
```

```
    jugadores.add(jugador);
```

```
}
```

## 2. Jugador.java

```
public class Jugador {
```

```
    private String nombre;
```

```
    public Jugador(String nombre) {
```

```
        this.nombre = nombre;
```

```
    }
```

```
    public String getNombre() {
```

```
        return nombre;
```

```
    }
```

```
}
```

## 3. Partido.java

```
public class Partido {
```

```
    private Equipo equipo1;
```

```
    private Equipo equipo2;
```

```
    private String fecha;
```

```
    private String resultado;
```

```
    private boolean finalizado;
```

```
    public Partido(Equipo equipo1, Equipo equipo2, String fecha) {
```

```
        this.equipo1 = equipo1;

        this.equipo2 = equipo2;

        this.fecha = fecha;

        this.resultado = "Pendiente";

        this.finalizado = false;

    }

    public void finalizarPartido(String resultado) {

        this.resultado = resultado;

        this.finalizado = true;

    }

    public boolean isFinalizado() {

        return finalizado;

    }

    public String getResultado() {

        return resultado;

    }

    public Equipo getEquipo1() {

        return equipo1;

    }
```

```
public Equipo getEquipo2() {  
  
    return equipo2;  
  
}
```

```
public String getFecha() {  
  
    return fecha;  
  
}  
}
```

#### 4. Torneo.java

```
public class Torneo {  
    private List<Equipo> equipos;
```

```
    List<Partido> partidos;
```

```
    public Torneo() {  
  
        equipos = new ArrayList<>();  
  
        partidos = new ArrayList<>();  
  
    }
```

```
    public void registrarEquipo(Equipo equipo) {  
  
        equipos.add(equipo);  
  
    }
```

```
    public void crearPartido(Equipo equipo1, Equipo equipo2, String fecha) {  
  
        partidos.add(new Partido(equipo1, equipo2, fecha));  
  
    }
```

```

public void listarEquipos() {

    for (Equipo equipo : equipos) {

        System.out.println("Equipo: " + equipo.getNombre() + ", Entrenador: " + equipo.getEntrenador());

        System.out.print("Jugadores: ");

        for (Jugador jugador : equipo.getJugadores()) {

            System.out.print(jugador.getNombre() + " ");

        }

        System.out.println();

    }

}

public void mostrarHistorialPartidos() {

    for (Partido partido : partidos) {

        String estado = partido.isFinalizado() ? "Finalizado" : "Pendiente";

        System.out.println("Partido: " + partido.getEquipo1().getNombre() + " vs " + partido.getEquipo2().getNombre() + " | Fecha: " + partido.getFecha() + " | Resultado: " + partido.getResultado() + " | Estado: " + estado);

    }

}

public void actualizarResultado(Partido partido, String resultado) {

```

```

        if (!partido.isFinalizado()) {

            partido.finalizarPartido(resultado);

        } else {

            System.out.println("El partido ya está finalizado.");

        }

    }

}

```

##### 5. Ejecutable.java

```

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

        Torneo torneo = new Torneo();

        // Registrar equipos

        Equipo equipo1 = new Equipo("Equipo A", "Entrenador A");

        equipo1.agregarJugador(new Jugador("Jugador 1"));

        equipo1.agregarJugador(new Jugador("Jugador 2"));

        torneo.registrarEquipo(equipo1);

        Equipo equipo2 = new Equipo("Equipo B", "Entrenador B");

        equipo2.agregarJugador(new Jugador("Jugador 3"));

        equipo2.agregarJugador(new Jugador("Jugador 4"));

        torneo.registrarEquipo(equipo2);

        // Crear partido
    }
}

```



```

torneo.crearPartido(equipo1, equipo2, "2023-10-01");

// Listar equipos

System.out.println("Equipos registrados:");

torneo.listarEquipos();

// Mostrar historial de partidos

System.out.println("\nHistorial de partidos:");

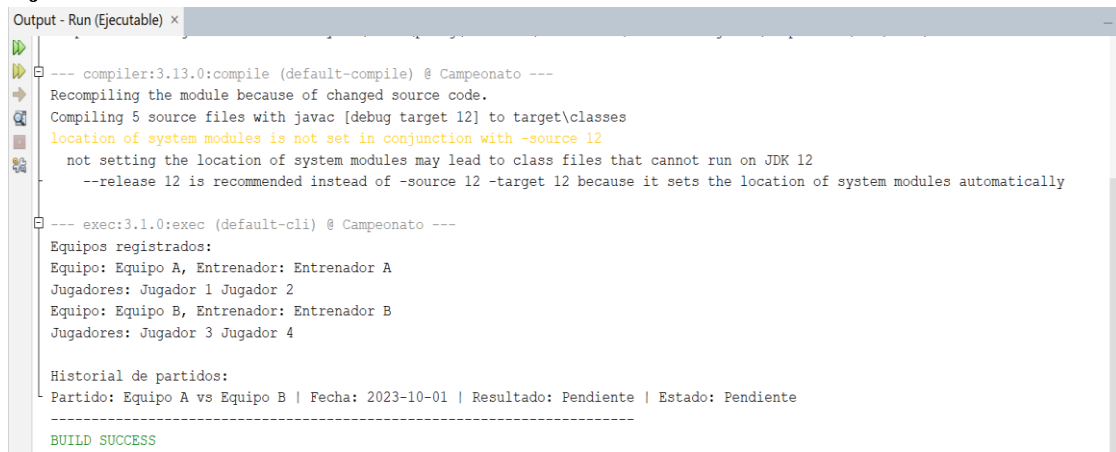
torneo.mostrarHistorialPartidos();

// Actualizar resultado

Partido partido = torneo.partidos.get(0);
    }
}

```

### 3. Ejecución



```

Output - Run (Ejecutable) x
--- compiler:3.13.0:compile (default-compile) @ Campeonato ---
Recompiling the module because of changed source code.
Compiling 5 source files with javac [debug target 12] to target\classes
location of system modules is not set in conjunction with -source 12
not setting the location of system modules may lead to class files that cannot run on JDK 12
--release 12 is recommended instead of -source 12 -target 12 because it sets the location of system modules automatically

--- exec:3.1.0:exec (default-cli) @ Campeonato ---
Equipos registrados:
Equipo: Equipo A, Entrenador: Entrenador A
Jugadores: Jugador 1 Jugador 2
Equipo: Equipo B, Entrenador: Entrenador B
Jugadores: Jugador 3 Jugador 4

Historial de partidos:
Partido: Equipo A vs Equipo B | Fecha: 2023-10-01 | Resultado: Pendiente | Estado: Pendiente
-----
BUILD SUCCESS

```

### Resumen de Resolución:

En la actividad se realizó un sistema de gestión de proyectos utilizando Programación Orientada a Objetos, para lograr el objetivo se realizó el diagrama UML que plantea los moldes de las clases que se solicitaron en el ejemplo con sus respectivos atributos y métodos, representando la base para la implementación del código que fue el paso

siguiente asignando tareas hacia las variables, verificando su funcionalidad y requerimientos planteados, permitiendo una estructura funcional, confiable y adaptable hacia distintos contextos proporcionados.

### **Conclusión:**

La implementación de la creación de este sistema dentro de distintas situaciones, como por ejemplo en el fútbol que fue el caso de esta actividad, a permitido profundizar los conocimientos como herencia, encapsulamiento y polimorfismo, fomentando el desarrollo de habilidades dentro de la gestión de proyectos dentro del área de programación, realizando un mejor desarrollo hacia un problema y dando su resolución.

### **Referencias:**

Herrera García, A. (2015). *Aplicación Java de gestión de campeonatos de baloncesto* [Presentación]. Prezi. <https://prezi.com/mashrsf-2tw-/aplicacion-java-de-gestion-de-campeonatos-de-baloncesto/>

González, T. (2024, 8 de febrero). *Creando un Gestor de Tareas Básico en Java*. InfoGonzalez. <https://infogonzalez.com/2024/02/creando-un-gestor-de-tareas-basico-en-java.html>

González Pacheco, E. F. (2019, 4 de febrero). *Programa en Java de torneo de partidos de fútbol* [Video]. YouTube. <https://www.youtube.com/watch?v=4uf9LjOK4wY>