



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

UNIVERSIDAD DE LAS FUERZAS
ARMADAS

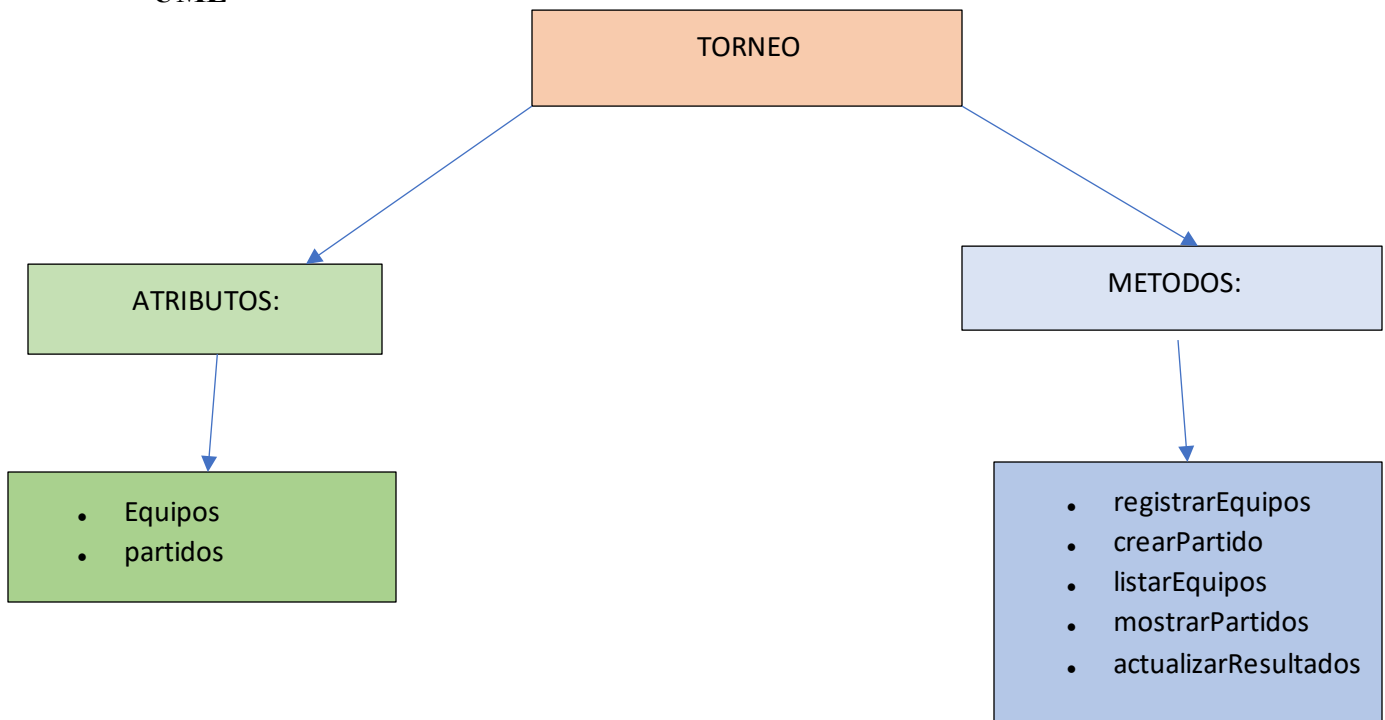
Actividad P00 Experimental **NOMBRES:**

Andrea Castro
Alexis Murminacho
Emerson Coro

NRC:

1323

UML



Actividad Experimental 1

Resolución

El propósito de la clase de torneo en Java es gestionar equipos y partidos de forma estructurada. Los dos componentes principales del sistema son una lista dinámica que puede almacenar nombres de equipos y una lista de programación que puede registrar partidos programados, lo que hace que la gestión de datos sea fácil y eficiente. Las técnicas implementadas incluyen registrar equipos, crear partidos entre equipos existentes, enumerar todos los equipos registrados, mostrar partidos programados y actualizar los resultados de los partidos. Este diseño admite la encapsulación, donde las propiedades solo se cambian entre métodos, lo que garantiza coherencia y modularidad en el sistema.

1. Definición de la clase:

Se creó la clase Torneo con dos atributos principales: una lista de equipos y una lista de partido. Las listas permiten un manejo dinámico de elementos, lo cual es ideal para registrar múltiples equipos y partidos.

2. Métodos:

- **registrarEquipos:** Permite añadir equipos al torneo.
- **crearPartido:** Agrega partidos formados por equipos registrados.
- **listarEquipos:** Muestra la lista de equipos registrados.
- **mostrarPartidos:** Muestra los partidos programados.
- **actualizarResultados:** Permite actualizar el estado de un partido.

3. Diagrama UML:

- Se creó un diagrama UML representando la clase Torneo, con sus atributos y métodos, destacando las relaciones entre ellos.
- El UML incluye:
 - ✦ Clase Torneo con sus atributos y métodos.
 - ✦ Uso de colecciones dinámicas para los atributos equipos y partidos.

Ventajas de Este Diseño

- **Flexibilidad:** El uso de listas dinámicas permite manejar un número variable de equipos y partidos sin restricciones predefinidas.
- **Modularidad:** Los atributos están encapsulados, y su manipulación está limitada a métodos específicos, lo que reduce errores y asegura consistencia.
- **Escalabilidad:** Es sencillo extender la clase para manejar más atributos (por ejemplo, fechas de partidos o ubicaciones) o incluir nuevas funcionalidades.