

## 102. Generics

### a. Crear un proyecto del tipo Biblioteca de Clases:

- I. Generar la clase Torneo con un tipo genérico.
  1. Restringir el tipo genérico para que deba ser del tipo Equipo o sus derivados. (where)
  2. Tendrá un atributo equipos: List y otro nombre: string.
  3. Sobrecargar el operador == para que controle si un equipo ya está inscripto al torneo.
  4. Sobrecargar el operador + para agregar un equipo a la lista, siempre y cuando este no se encuentre ya en el torneo.
  5. El método Mostrar retornará los datos del torneo y de los equipos participantes.
  6. El método privado CalcularPartido(T, T): string recibirá dos elementos del tipo T, que deberán ser del tipo Equipo o sus herencias, y calculará utilizando Random un resultado del partido. Retornará el resultado con el siguiente formato: "[EQUIPO1] [RESULTADO1] – [RESULTADO2] [EQUIPO2]", siendo EQUIPOX el nombre del equipo y RESULTADOX la cantidad de goles/puntos.
  7. La propiedad pública JugarPartido tomará dos equipos de la lista al azar y calculará el resultado del partido a través del método CalcularPartido.
- II. Generar la clase Equipo abstracta.
  1. Agregar un atributo nombre: string y otro fechaCreacion: datetime.
  2. Dos equipos serán iguales si comparten el mismo nombre y fecha de creación.
  3. El método Ficha retornará el nombre del equipo y su fecha de creación con el siguiente formato "[EQUIPO] fundado el [FECHA]".
- III. Generar la clase EquipoFutbol que herede de Equipo.
- IV. Generar la clase EquipoBasquet que herede de Equipo.

### b. Crear un proyecto del tipo Console:

- I. Crear un programa que genere 2 torneos, uno de Futbol y otro de Basquet.
- II. Crear 3 equipos de cada tipo.

- III. Agregar los equipos en tantos torneos como se pueda.
- IV. Llamar al método `Mostrar de Torneo` e imprimir su retorno por pantalla.
- V. Llamar al menos 3 veces a la propiedad `JugarPartido` de cada torneo e imprimir su respuesta por pantalla.