**Principio de Sustitución de Liskov (LSP)**

**Ejercicio1**

**BEFORE**

Tenemos un programa donde vamos a listar deportistas con su equipo y su categoría, para ello vamos a tener una clase **Player** que tiene los siguientes atributos: **FirstName**, **LastName**, **team**, **categoría** y un método que nos **obtenga el equipo** y un método que **nos asigne el equipo** y un método para **cambiar la categoria**, tiene a su vez también un **enum** con las **categorías (amateur, Profesional)**.

Luego tendremos una clase hija **FootbalPlayer** que herede de **Player** y tiene un método **GetMarketValue** donde obtenemos el valor comercial de un deportista según su categoría, luego tendremos una clase **TennispLayer** también que heredara de Player.

Vemos que al cambiar de **footbalPlayer** a **tennisPlayer** se rompe con dicho principio ya que no se cumple con los métodos mínimos necesarios de la clase base.

**AFTER**

Lo primero que se tendría que realizar es determina todo lo que es común en ambas clases para tenerlo presente en la clase base, como vemos la propiedad **Team**, los métodos de **obtener Team y asignarTeam** son solo propias de la clase **footbalPlayer,** por lo que vemos que las demás propiedades y métodos se asignaran a una **interface** para así poder heredarla, lo mismo se hará con todo lo relacionado a **Team**, se asignara a una interface y así poder heredarla.

**Ejercicio 2**

**BEFORE**

En este ejercicio tenemos una clase **Animal** y sus métodos **makeNoise** y un método **fly,** a su vez tenemos una clase hija **Dog** y una clase hija **Bird**, como vemos la clase hija **Dog** no tiene el método **Fly.**

**AFTER**

Diferenciar todo aquello que sea común a ambas clases y llevarlo a una interfaz, posteriormente implementar los métodos específicos a cada clase si es que hubiese.