Práctico 1

Por Sergio J. Antozzi

Ejercicio 1: Interfaces

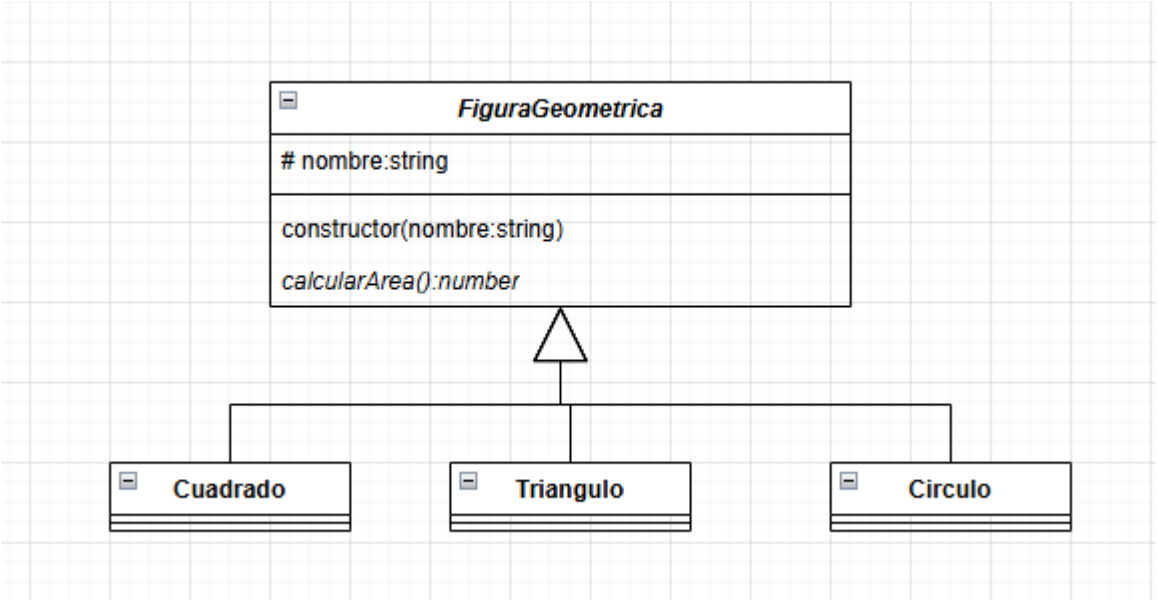
Definir una **interface `Animal`** que tenga los siguientes métodos:

- `hacerSonido(): void`
- `moverse(): void`

Luego, implementar la interface en una clase concreta llamada `Perro` que imprima en consola:

- `"Guau!"` al llamar `hacerSonido()`
- `"El perro corre"` al llamar `moverse()`

Ejercicio 2: Clase Abstracta



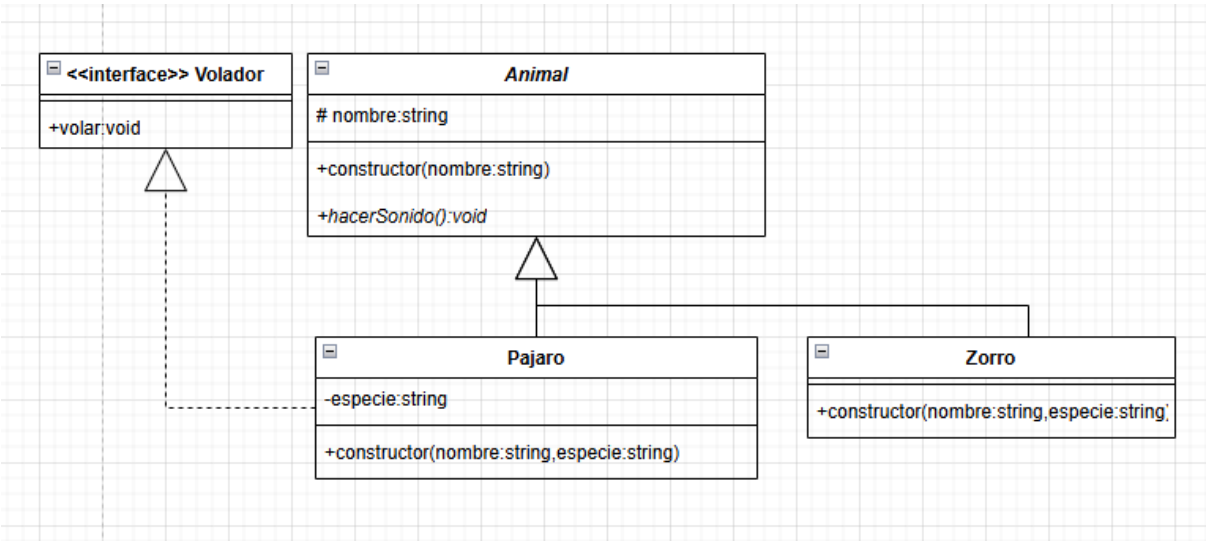
Implementar el método `calcularArea()` usando la fórmula del área de cada uno.

Ejercicio 3: Herencia y Polimorfismo

1. Crear una clase abstracta `Empleado` con:
    - `nombre: string`
    - `salarioBase: number`
    - un método abstracto `calcularSalario(): number`
2. Crear dos subclases:
    - `EmpleadoTiempoCompleto` (suma un bono fijo de $20.000 al salario base).
    - `EmpleadoMedioTiempo` (cobra el 50% del salario base).
3. Crear un arreglo de tipo `Empleado[]` y demostrar **polimorfismo** recorriéndolo y mostrando el salario calculado de cada empleado.
4. Hacer el diagrama UML.

Ejercicio 4: UML

Dado el siguiente diagrama UML, **interpretar** qué representa y escribir el código en TypeScript correspondiente.



Ejercicio 5: Diseño de UML propio

Diseñar un **diagrama UML** para un sistema de **vehículos** con al menos:

- Una clase abstracta `Vehiculo`
- Dos clases concretas que hereden de ella (ej: `Auto`, `Moto`)
- Una interface `Electrico` que pueda ser implementada por alguno de los vehículos.

Luego, **implementar el código en TypeScript** siguiendo tu propio UML.

Entrega: Un txt con el repositorio de github donde se encuentre el TP1, en el readme deben estar los diagramas uml pedidos y los integrantes del grupo, solo entrega el integrante representante.
En el txt puede agregar algún detalle, de como se trabajó y si hubo algún integrante que no aportó.