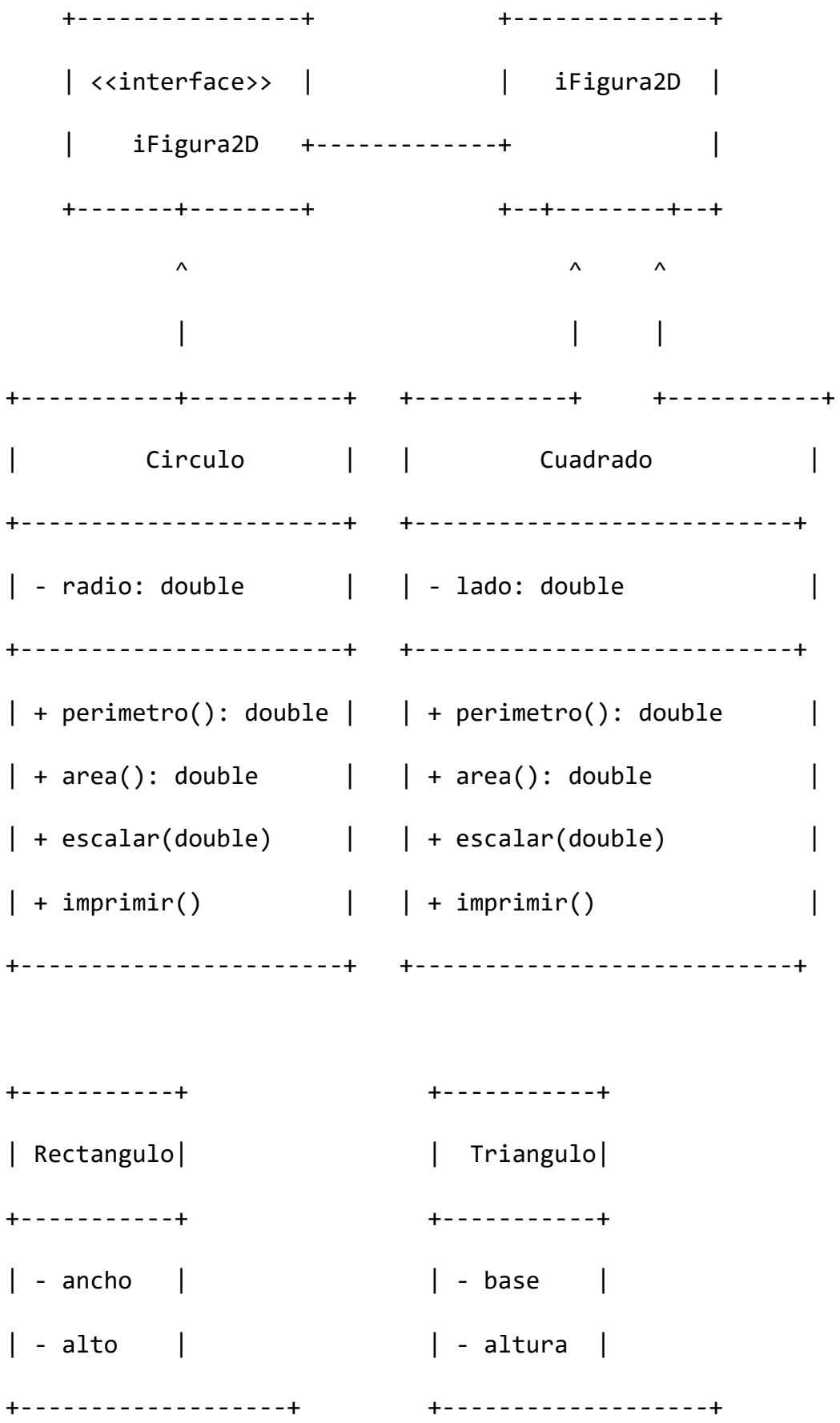

JAAVA

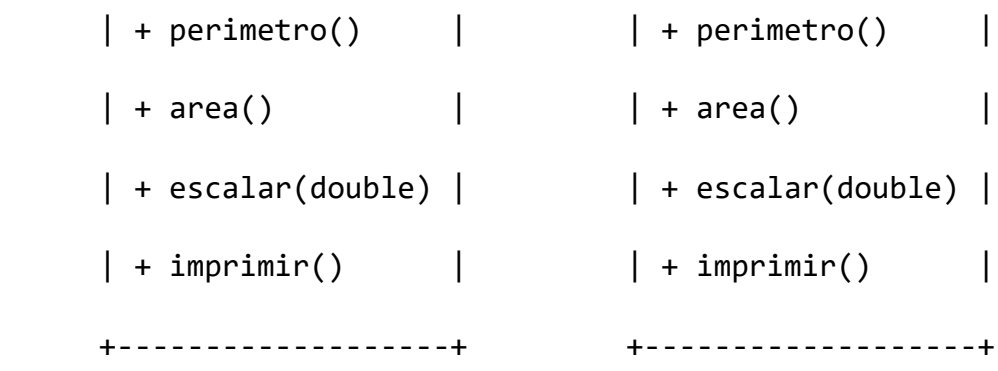
Foto generada con la IA de Canva

Índice

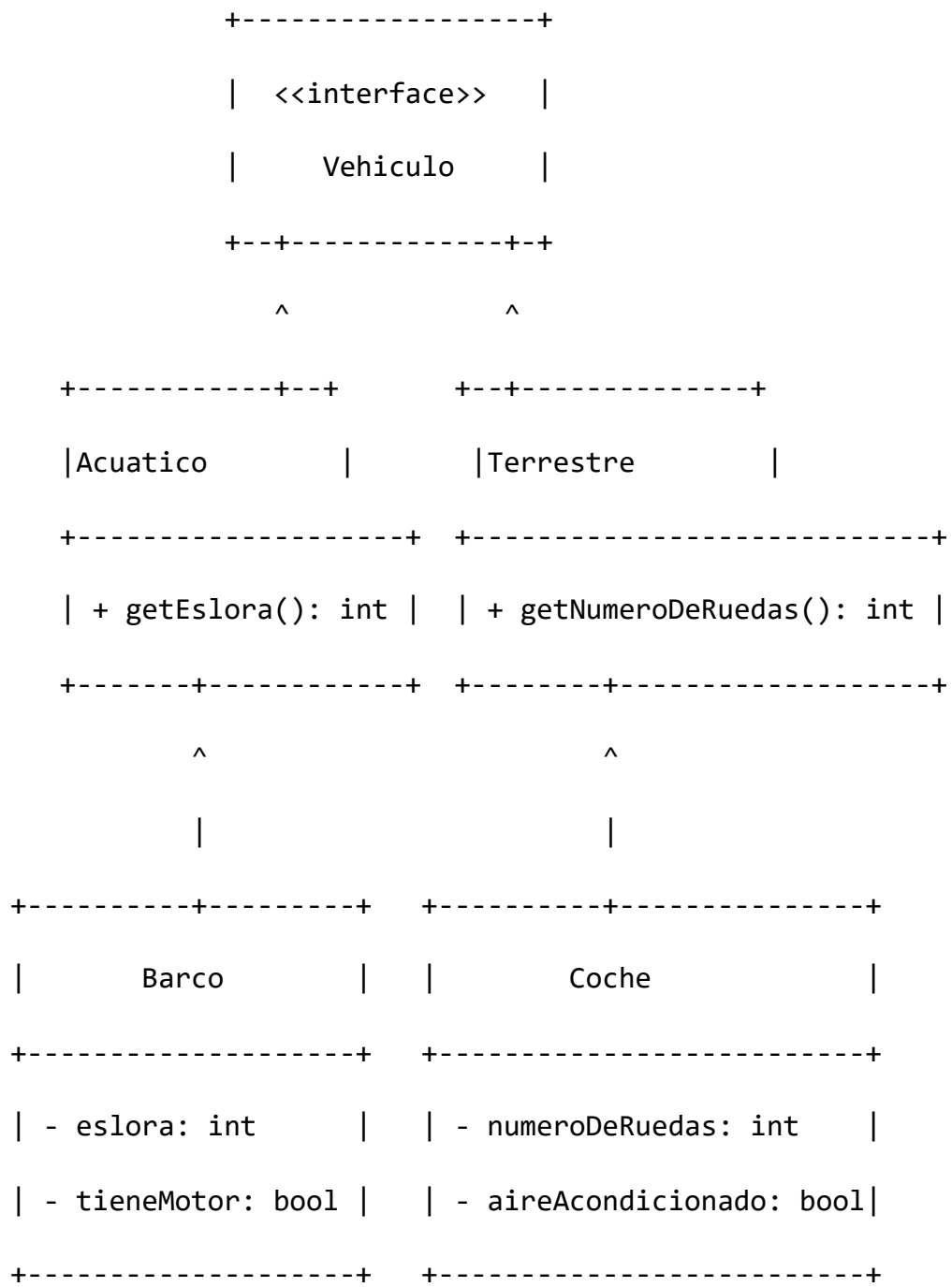
▢Diseño UML

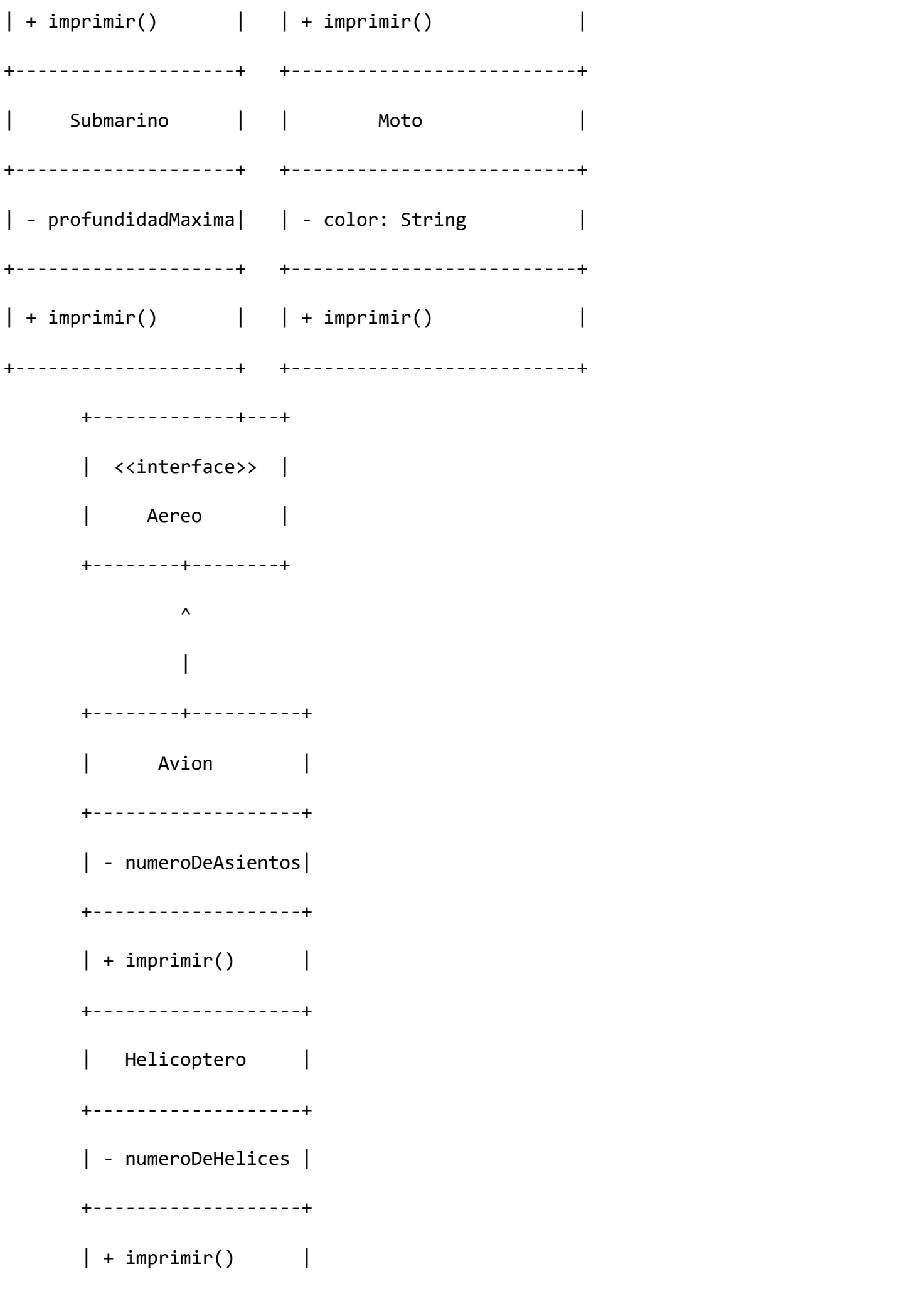
Proyecto Figura





Proyecto Vehículos

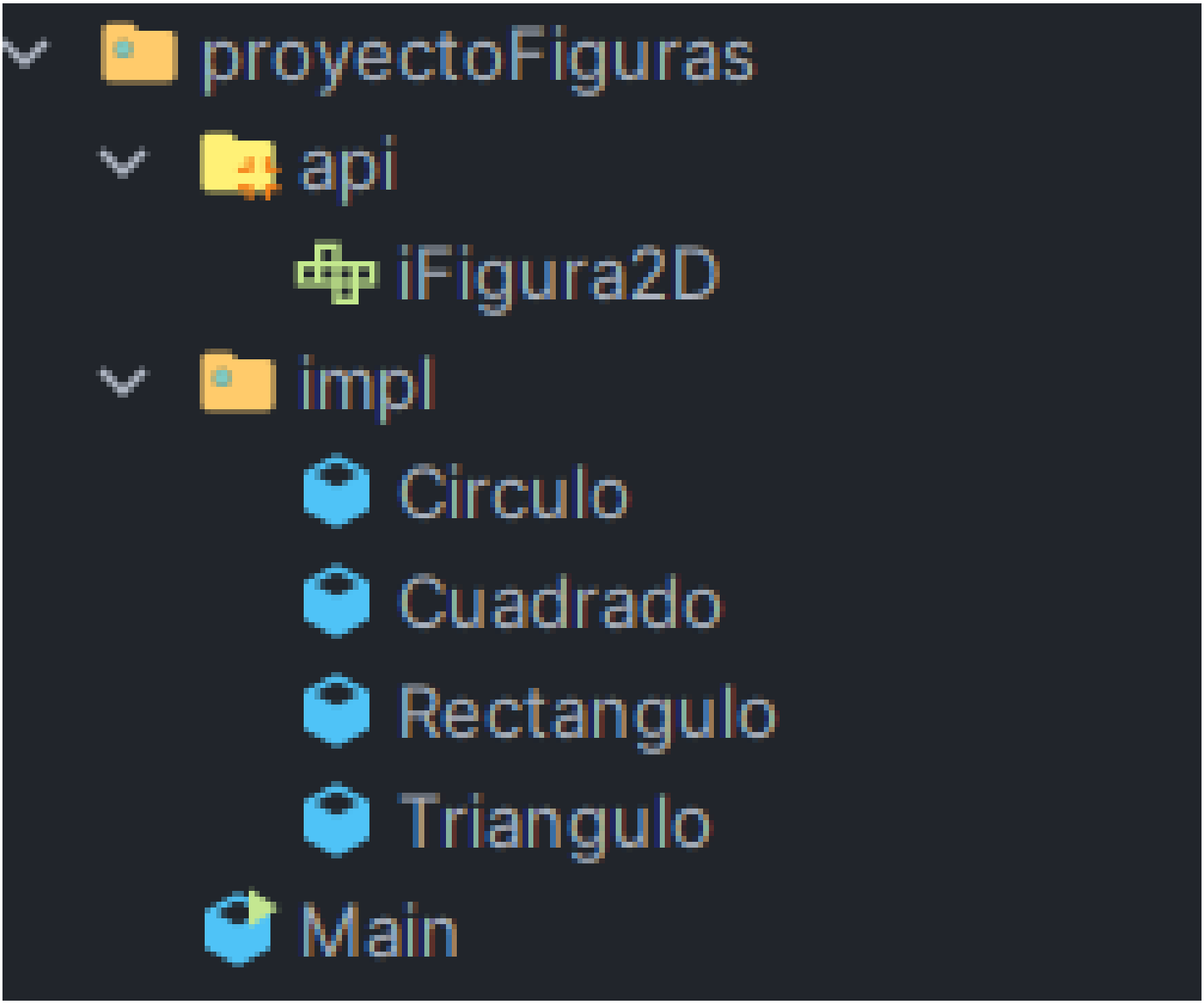



+ imprimir()
Submarino
- profundidadMaxima
+ imprimir()
+ imprimir()
Moto
- color: String
+ imprimir()
<<interface>>
Aereo
Avion
- numeroDeAsientos
+ imprimir()
Helicoptero
- numeroDeHelices
+ imprimir()

```
      +------------------+
```

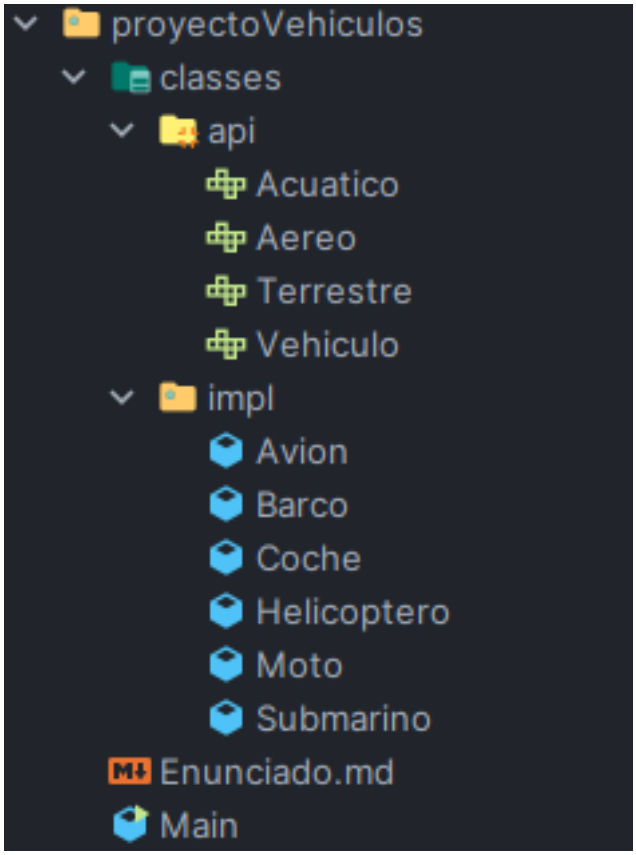
Estructura de Proyecto

Proyecto Figuras



El proyecto se conforma por la carpeta principal del proyecto que ésta a su vez almacena la carpeta api donde se aloja la interface iFigura2D y la carpeta impl donde se alojan los archivos .class del proyecto que explicaré más adelante

Proyecto Vehículos



El proyecto se conforma por: La carpeta principal con el nombre del proyecto > carpeta classes (donde se aloja la carpeta api e impl) > carpeta api (donde se almacenan las interfaces) > carpeta impl (donde almacenamos los archivos .class) que explicaré más adelante

---

▢ Interfaces

Proyecto Figura

Interface iFigura2D:

La interfaz iFigura2D define los métodos fundamentales que deben ser implementados por cualquier figura geométrica bidimensional. Proporciona métodos para calcular el perímetro y el área de la figura, escalar la figura por un factor dado, e imprimir la información de la figura

Proyecto Vehículos

Interface Acuático:

La interface Acuático define el método para obtener la eslora del objeto barco e incluye el método imprimir para mostrar posteriormente las características del objeto barco.

Interface Aéreo:

La interface Aéreo define el método para obtener el número de asientos del avión e incluye también el método imprimir para mostrar las características del objeto Avión y Helicóptero

Interface Terrestre:

La interface Terrestre define el método para obtener el número de ruedas de un vehículo terrestre y también incorpora el método imprimir para mostrar las características del objeto coche y moto.

Interface Vehículo:

La interface Vehículo define los métodos obtener matrícula y obtener modelo ya que son cosas que identifican a un vehículo, también incorpora el método imprimir

---

▫ Clases

Proyecto Figura

Clase Círculo:

La clase Circulo implementa la interfaz iFigura2D para representar una figura geométrica circular.
Proporciona métodos para calcular el perímetro y el área de un círculo, escalar su tamaño y
imprimir sus detalles.

Clase Cuadrado:

La clase Cuadrado implementa la interfaz iFigura2D, definiendo los métodos para calcular el perímetro y el área de un cuadrado, escalar sus dimensiones y imprimir
sus detalles. Esta clase modela un cuadrado mediante la longitud de su lado.

Clase Rectángulo:

La clase Rectangulo implementa la interfaz iFigura2D, proporcionando una implementación
concreta para los métodos de cálculo de perímetro y área, escalado de dimensiones,
y presentación de detalles específicos para un rectángulo. Esta clase representa un rectángulo
mediante sus dimensiones de ancho y alto.

Clase Triángulo:

La clase Triangulo implementa la interfaz iFigura2D, ofreciendo una implementación específica para calcular el perímetro y el área de un triángulo, así como para escalar sus dimensiones e imprimir detalles sobre el triángulo. Esta clase representa un triángulo utilizando su base y altura.

Clase Main:

Clase principal para demostrar el uso de figuras geométricas.
Crea una lista de figuras geométricas, imprime sus detalles,
escala sus dimensiones y vuelve a imprimir los detalles actualizados.

Proyecto Vehículo

Clase Vehículo:

La clase Avion implementa las interfaces Aéreo y Vehículo, representando un vehículo aéreo con características específicas como matrícula, modelo, número de asientos y tiempo máximo de vuelo.

Clase Barco:

La clase Barco implementa las interfaces Acuatico y Vehiculo, representando un vehículo acuático con características específicas como matrícula, modelo, eslora y si tiene motor.
Modela las propiedades y comportamientos fundamentales de un barco.

Clase Coche:

La clase Coche implementa las interfaces Terrestre y Vehículo, representando un vehículo terrestre con características específicas como matrícula, modelo, número de ruedas y si tiene aire acondicionado.
Modela las propiedades y comportamientos fundamentales de un coche.

Clase Helicóptero:

La clase Helicóptero implementa las interfaces Aéreo y Vehículo, representando un vehículo aéreo con características específicas como matrícula, modelo, número de asientos y número de hélices. Modela las propiedades y comportamientos fundamentales de un helicóptero.

Clase Moto:

La clase Moto implementa las interfaces Terrestre y Vehiculo, representando un vehículo terrestre con características específicas como matrícula, modelo, número de ruedas y color. Modela las propiedades y comportamientos fundamentales de una moto.

Clase Submarino:

La clase Submarino implementa las interfaces Acuatico y Vehiculo, representando un vehículo acuático con características específicas como matrícula, modelo, eslora y profundidad máxima de inmersión. Modela las propiedades y comportamientos fundamentales de un submarino.

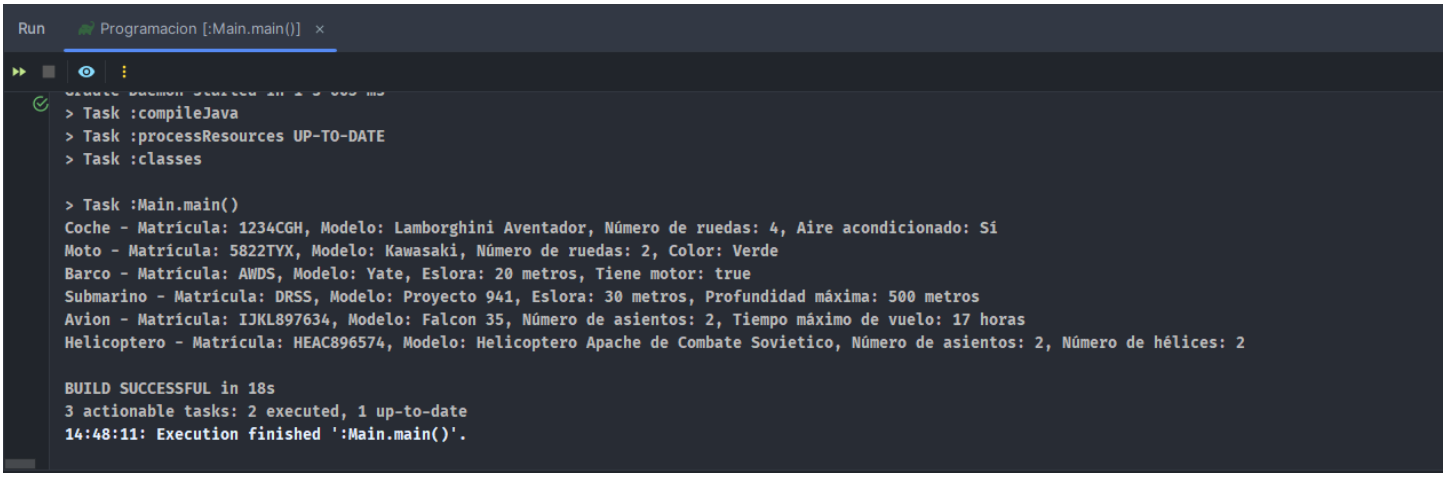
Clase Main:

Clase Principal para instanciar vehículos de clase Aéreo, Terrestre y Acuático, haciendo uso de un ArrayList almacenamos los vehículos y después con ella función .add agregada a vehiculos podremos agregarlos al ArrayList. Después haciendo uso de un bucle for recorremos el ArrayList y mostramos los vehículos almacenados

---

▢ Pruebas de funcionamiento

Proyecto Vehículo



Proyecto Figura

```
Datos de las figuras:
Cuadrado: Lado = 4.0, Perimetro = 16.0, Area = 16.0
Rectangulo: Ancho = 2.0, Alto = 3.0, Perimetro = 10.0, Area = 6.0
Triangulo: Base = 4.0, Altura = 5.0, Perimetro = 14.0, Area = 10.0
Circulo: Radio = 3.0, Perimetro = 18.84955592153876, Area = 28.274333882308138
--------------------
Aumentar el tamaño:
Cuadrado: Lado = 8.0, Perimetro = 32.0, Area = 64.0
Rectangulo: Ancho = 4.0, Alto = 6.0, Perimetro = 20.0, Area = 24.0
Triangulo: Base = 8.0, Altura = 10.0, Perimetro = 28.0, Area = 40.0
Circulo: Radio = 6.0, Perimetro = 37.69911184307752, Area = 113.09733552923255
--------------------
Disminuir el tamaño:
Cuadrado: Lado = 0.8, Perimetro = 3.2, Area = 0.6400000000000001
Rectangulo: Ancho = 0.4, Alto = 0.6000000000000001, Perimetro = 2.0, Area = 0.24000000000000005
Triangulo: Base = 0.8, Altura = 1.0, Perimetro = 2.8, Area = 0.4
Circulo: Radio = 0.6000000000000001, Perimetro = 3.769911184307752, Area = 1.1309733552923258

BUILD SUCCESSFUL in 431ms
3 actionable tasks: 1 executed, 2 up-to-date
14:50:07: Execution finished ':Main.main()'.
```

Realizado por: Ángel García Pérez 1DAW