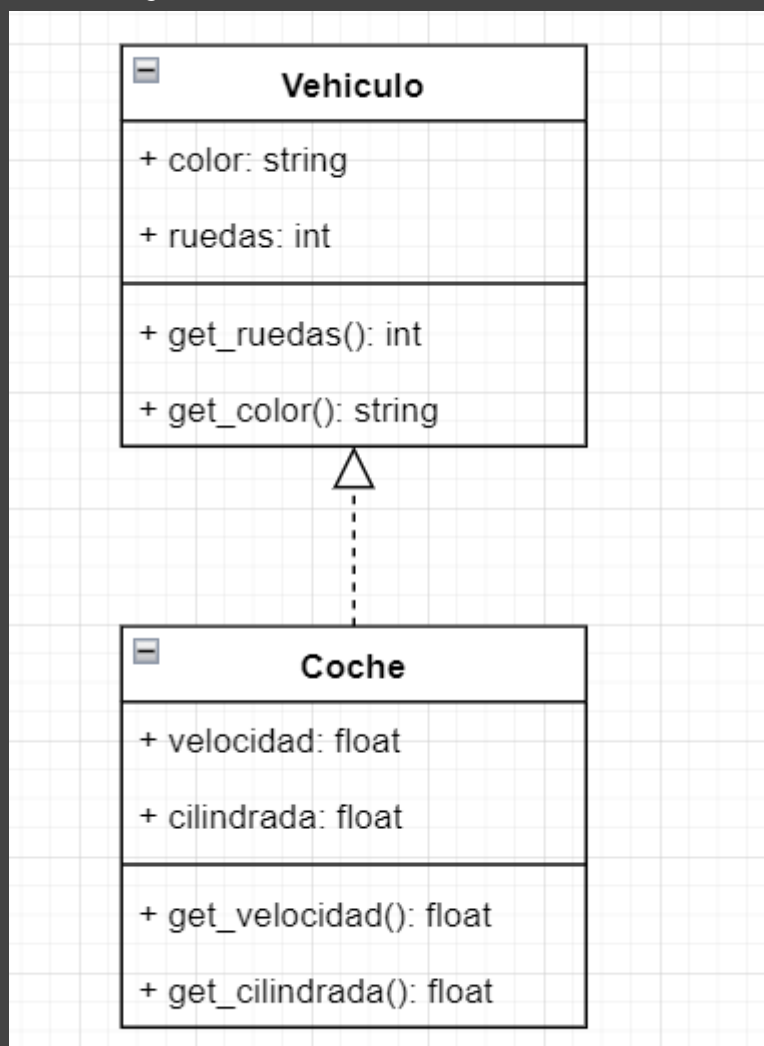


Guía de ejercicios POO - Unidad 5 - Herencia

1. Crear una clase Instrumento que:
 - a. Tenga como atributos el precio (int), marca(string) y modelo(string).
 - b. El método tocar().
 - c. Luego se crearán las clases Guitarra, Batería y Piano, las cuales heredarán de la clase Instrumento.
 - d. Es importante que cada instrumento, al llamarse al método tocar() tenga su PROPIA IMPLEMENTACIÓN, es decir: si creo un objeto de la clase Guitarra y llamo al método tocar(), en la consola de salida voy a visualizar el mensaje "Sonando como una guitarra", y así por cada objeto de su respectiva clase

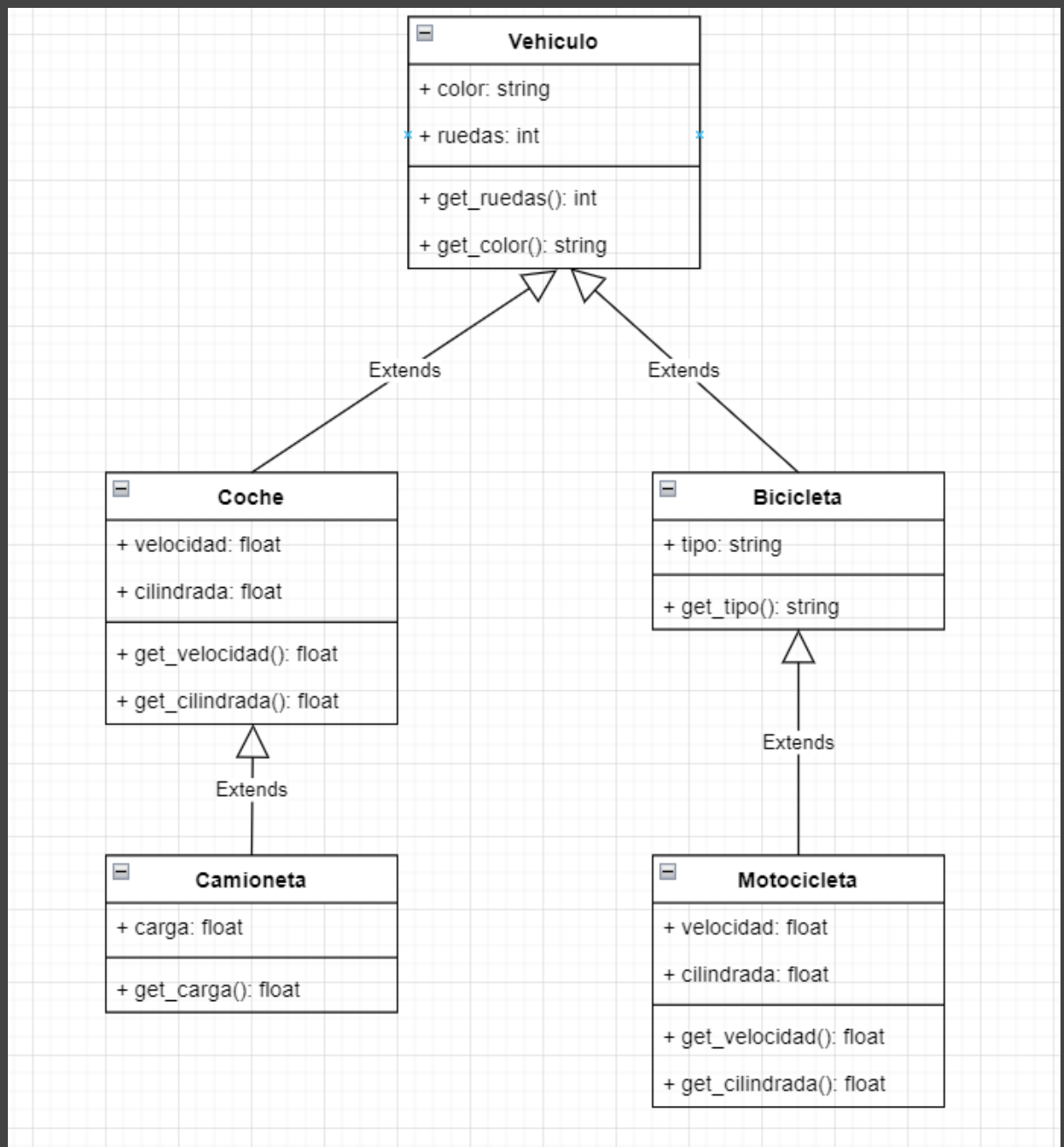
2. A partir del siguiente diagrama escribir la clase Vehiculo y Coche.

- a. Diagrama



- b. Por ahora solamente vamos a hacer herencia de esos atributos y métodos.

3. Agregar las clases Bicicleta, Motocicleta y Camioneta, siguiendo el diagrama inferior



- Ahora se pide agregar los atributos *combustible*, *distanciaRecorrida* y *relacionConsumoDistancia*(por ejemplo: será un número float o decimal, cada 1 litro de combustible recorro 0,5 km) y los métodos *cargarCombustible* y *recorrerDistancia*.
- El método *cargarCombustible* sumará más combustible al Vehiculo.
- El método *recorrerDistancia* verificará si la distancia por parámetro es viable para realizar (el trayecto o viaje) y si es así, sumará la distancia recorrida y luego restará el combustible que se gastó
- Para determinar el consumo de combustible se realiza la multiplicación de *relacionConsumoDistancia * distancia*

- Por último, ¿Qué pasa con la clase Bicicleta, lleva o no lleva combustible? Plantear a nivel diagrama o código que posible solución pueden realizar para separar la lógica de consumo de combustible con respecto a la clase Bicicleta. (Es decir tratar de generar una clase o dos más para después poder realizar una herencia y que la lógica del consumo de combustible NO esté disponible en la bicicleta, suponiendo que la bicicleta no necesita combustible)
4. Se desea implementar un programa que gestione empleados (clase Empleado).
- a. Los empleados se definen por tener como atributos:
 - i. Nombre
 - ii. Edad
 - iii. Salario
 - iv. También tendremos una constante llamada PLUS, que tendrá un valor de 300€ (¿Atributo de clase o de instancia?)
 - b. Tenemos dos tipos de empleados: Repartidor y Comercial.
 - i. El comercial, aparte de los atributos anteriores, tiene uno más llamado comisión (double).
 - ii. El repartidor, aparte de los atributos de empleado, tiene otro llamado zona (String).
 - iii. Crea sus constructores, getters, setters, __str__ (pensá como aprovechar la herencia).
 - iv. IMPORTANTE: No se podrán crear objetos del tipo Empleado (la clase padre) pero si de sus hijas.
 - v. Las clases tendrán un método llamado plus, que según en cada clase tendrá una implementación distinta.
 - 1. En Comercial, si tiene más de 30 años y cobra una comisión de más de 200 euros, se le aplicara el plus.
 - 2. En Repartidor, si tiene menos de 25 y reparte en la "zona 3", este recibirá el plus.
 - c. Escribir un programa main que compruebe el funcionamiento
5. ~~Vamos a hacer unas mejoras a la clase Baraja del ejercicio de la guía anterior~~
- ~~a. Lo primero que haremos es que nuestra clase Baraja será la clase padre y será abstracta.~~
 - ~~b. Le añadiremos el número de cartas en total y el número de cartas por palo.~~
 - ~~c. El método crearBaraja() será abstracto.~~
 - ~~d. La clase Carta tendrá un atributo genérico que será el palo de nuestra versión anterior.~~
 - ~~e. Creamos clases que sean:~~
 - ~~i. PalosBarEspañola:~~
 - ~~1. OROS~~
 - ~~2. COPAS~~
 - ~~3. ESPADAS~~
 - ~~4. BASTOS~~
 - ~~ii. PalosBarFrancesa:~~

1. ~~DIAMANTES~~
2. ~~PICAS~~
3. ~~CORAZONES~~
4. ~~TREBOLES~~

f. ~~Creamos dos clases hijas:~~

i. ~~BarajaEspañola: tendrá un atributo boolean para indicar si queremos jugar con las cartas 8 y 9 (total 48 cartas) o no (total 40 cartas).~~

ii. ~~BarajaFrancesa: no tendrá atributos, el total de cartas es 52 y el número de cartas por palo es de 13. Tendrá dos métodos llamados:~~

1. ~~cartaRoja(Carta<PalosBarFrancesa> c): si el palo es de corazones y diamantes.~~

2. ~~cartaNegra(Carta<PalosBarFrancesa> c): si el palo es de tréboles y picas.~~

iii. ~~Si el palo es de tipo PalosBarFrancesa:~~

1. ~~La carta número 11 será Jota~~

2. ~~La carta numero 12 será Reina~~

3. ~~La carta numero 13 será Rey~~

4. ~~La carta numero 1 será As~~

iv. ~~Si el palo es de tipo PalosBarFrancesa:~~

1. ~~La carta numero 10 será Sota~~

2. ~~La carta numero 12 será Caballo~~

3. ~~La carta numero 13 será Rey~~

4. ~~La carta numero 1 será As~~