

# Ejercicios POO lvl 1

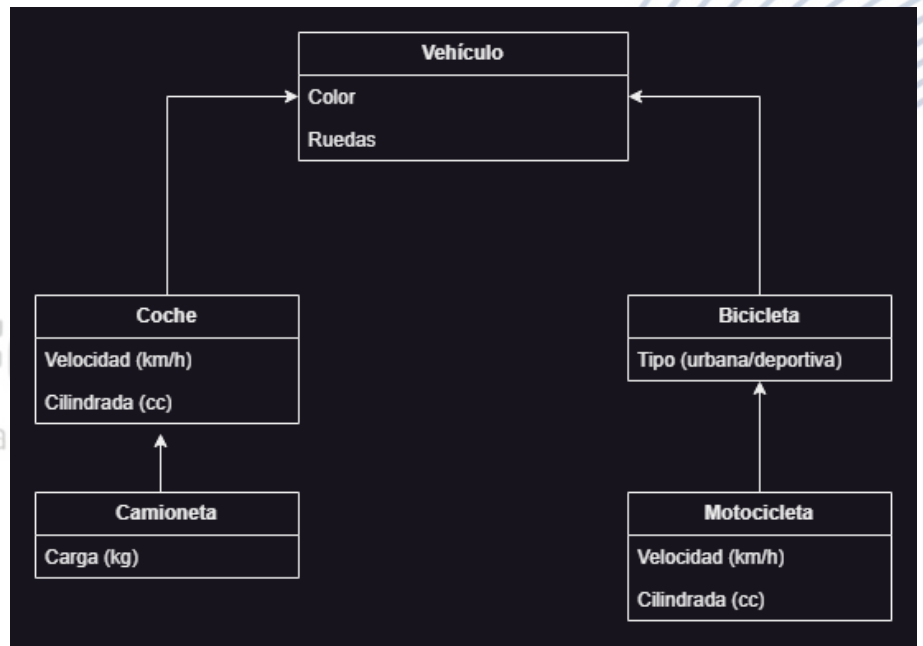


## Ejercicio 1: Vehículo pt.1

A partir del siguiente diagrama de clases, implementá clases y métodos para mostrar atributos.

## Ejercicio 2: Vehículo pt.2

- Crea al menos un objeto de cada subclase y añádelos a una lista llamada vehículos.
- Realiza una función llamada **catalogar()** que reciba la lista de vehículos y los recorra mostrando el nombre de su clase y sus atributos.
- Modifica la función **catalogar()** para que reciba un argumento optativo **ruedas**, haciendo que muestre únicamente los que su número de ruedas concuerde con el valor del argumento. También debe mostrar un mensaje "Se han encontrado {} vehículos con {} ruedas:" únicamente si se envía el argumento **ruedas**. Ponla a prueba con 0, 2 y 4 ruedas como valor.



## Ejercicio 3: Triángulo

Desarrollar un programa que cargue los datos de un triángulo.

Implementar una clase con los métodos para inicializar los atributos, imprimir el valor del lado con un tamaño mayor y el tipo de triángulo que es (equilátero, isósceles o escaleno).

## Ejercicio 4: Tiempo ✓

Crear una clase Tiempo, con atributos hora, minuto y segundo, que pueda ser instanciada indicando: los tres atributos, sólo la hora y minuto, o sólo la hora. Crear además los métodos necesarios para modificar la hora en cualquier momento de forma manual. Mantenga la integridad de los datos en todo momento definiendo de tipo "private". Crear una clase prueba\_tiempo que prueba una hora concreta y la modifique a su gusto mostrándola por pantalla.

## Ejercicio 5: Gestión de Donaciones ✓

Nos piden que diseñemos un programa para gestionar donaciones de alimentos. Los productos tienen los siguientes atributos:

- Nombre
- Cantidad

Tenemos dos tipos de productos:

- Perecedero: tiene un atributo llamado días a caducar.
- No perecedero: tiene un atributo llamado tipo.

Tendremos una función llamada Calcular, que según cada clase hará una cosa u otra, a esta función se le envía la cantidad por producto y entidades a las cuáles se entregarán donaciones.

- Debe obtener la cantidad que se enviará a cada entidad, sabiendo que la distribución debe ser lo más equitativa posible. En caso que sobren productos, se almacenan para el próximo ciclo de donación.
- Además si el producto es perecedero, se informará:
  - Si le queda menos de 10 días para caducar, la entrega debe hacerse en el día actual.
  - Si le queda 1 mes para caducar, la entrega debe hacerse en el plazo de 1 semana.
- Si fuera No Perecedero, se informa cuántos productos se entregarán por entidad y que queda libre la elección de la fecha de entrega siempre que no supere el mes.

## Ejercicio 6: Cuentas Electrónicas ✓

Crea una clase llamada Cuenta que tendrá los siguientes atributos: titular (que es una persona) y cantidad (puede tener decimales). El titular será obligatorio y la cantidad es opcional.

Implementa los siguientes métodos:

- mostrar(): Muestra los datos de la cuenta.
- ingresar(cantidad): se ingresa una cantidad a la cuenta, si la cantidad introducida es negativa, no se hará nada.
- retirar(cantidad): se retira una cantidad a la cuenta. La cuenta puede estar en números rojos.

## Ejercicio 7: Bebidas Online ✓

Vamos a administrar un ecommerce de bebidas.

En un depósito se guardan las bebidas a comercializar.

Estos productos son bebidas como agua mineral y gaseosas.

De los productos nos interesa saber su identificador (cada uno tiene uno distinto), cantidad de litros, precio y marca.

Si es agua mineral nos interesa saber también el origen (Manantial, Ciudad, etc).

Si es una gaseosa queremos saber el porcentaje que tiene de azúcar y si tiene o no alguna promoción (si la tiene tendrá un descuento del 10% en el precio).

Las operaciones del almacén son las siguientes:

- Calcular precio de todas las bebidas: calcula el precio total de todos los productos del almacén.
- Calcular el precio total de una marca de bebida: dada una marca, calcular el precio total de esas bebidas.
- Agregar producto: agrega un producto, si el identificador esta repetido en alguno de las bebidas, no se agregará esa bebida.
- Eliminar un producto: dado un ID, eliminar el producto del depósito.
- Mostrar información: mostramos para cada bebida toda su información.

## Ejercicio 8: Mis Libros Favoritos ✓

Vamos a crear un programa para nuestros libros favoritos.

Queremos mantener una lista de los libros que hemos ido leyendo, calificando según nos haya gustado más o menos al leerlo.

- Para ello, crear la clase Libro, cuyos atributos son el título, el autor, el número de páginas y la calificación que le damos entre 0 y 10.
- Crear los métodos para poder modificar y obtener los atributos si tienen sentido.
- Posteriormente, crear una clase ConjuntoLibros, que almacena un conjunto de libros. Se pueden añadir libros que no existan (siempre que haya espacio), eliminar libros por título o autor, mostrar por pantalla los libros con la mayor y menor calificación dada y, por último, mostrar un contenido de todo el conjunto.

## Ejercicio 9: Cafetera robot ✓

Cómo diseñaríamos el comportamiento de una cafetera robot?

Desarrolla una clase Cafetera con atributos:

- `_capacidadMaxima` (la cantidad máxima de café que puede contener la cafetera)
- `_cantidadActual` (la cantidad actual de café que hay en la cafetera).



**Luego desarrollar los siguientes métodos:**

- **llenarCafetera():** pues eso, hace que la cantidad actual sea igual a la capacidad.
- **servirTaza(int):** simula la acción de servir una taza con la capacidad indicada. Si la cantidad actual de café “no alcanza” para llenar la taza, se sirve lo que quede.
- **vaciarCafetera():** pone la cantidad de café actual en cero.
- **agregarCafe(int):** añade a la cafetera la cantidad de café indicada.

## **Ejercicio 10: Tomamos un Mate**

**Modelar una clase Mate que describa el funcionamiento de la conocida bebida tradicional argentina. La clase debe contener como miembros:**

- **Un atributo para la cantidad de cebadas restantes hasta que se lava el mate (representada por un número).**
- **Un atributo para el estado (lleno o vacío).**
- **El constructor debe recibir como parámetro n, la cantidad máxima de cebadas en base a la cantidad de yerba vertida en el recipiente.**
- **Un método cebar, que llena el mate con agua. Si se intenta cebar con el mate lleno, se debe lanzar una excepción que imprima el mensaje ¡Cuidado! ¡Te quemaste!**
- **Un método beber, que vacía el mate y le resta una cebada disponible. Si se intenta beber un mate vacío, se debe lanzar una excepción que imprima el mensaje. El mate está vacío!**
- **Es posible seguir cebando y bebiendo el mate aunque no haya cebadas disponibles. En ese caso la cantidad de cebadas restantes se mantendrá en 0, y cada vez que se intente beber se debe imprimir un mensaje de aviso: Advertencia: "el mate está lavado", pero no se debe lanzar una excepción.**