**Unidad 3: Objetos y funciones. Avanzado**

#### CLASES

#### EJERCICIO 48: Vector

Crea una clase **Vector** que representa un vector. Al crearlo, se pasarán como parámetros dos valores numéricos que serán la **x** (distancia al punto 0,0 en el eje de las x), y la **y** ((distancia al punto 0,0 en el eje de las y).

Esta clase tendrá dos métodos más: **sumar** y **restar**, que toman otro vector como parámetro, y devuelve un nuevo vector resultado de la suma o diferencia de los valores x e y de los dos vectores (el **this** y el parámetro).

Además, tendrá una propiedad **longitud**, que devuelve la longitud del vector, esto es, la distancia desde el origen (0,0) al punto (x, y).

console.log(new Vector(1,2).sumar(2, 3)); // => {x: 3, y: 5};

console.log(new Vector(1,2).restar(2, 3)); // => {x: -1, y: -1};

console.log(new Vector(3, 4).longitud); // => 5

#### EJERCICIO 49: Vehículo

Crea una clase denominada **Vehículo** que tenga algunas características como el **color**, **marca**, **modelo** y la **velocidad máxima** que puede alcanzar.

También tendrá tres métodos además del constructor: **comenzar** que mostrará el mensaje “Encender motor”, **parar** que mostrará “Apagar motor” y **distanciaMax** que recibirá un parámetro con indicando el tiempo de funcionamiento y calculará la distancia máxima que puede haber recorrido, que será el resultado de multiplicar la velocidad máxima por el tiempo.

#### EJERCICIO 50: Reloj

La clase **Clock** (ver archivo) está escrita en estilo funcional. Reescríbela en sintaxis de clase.

P.D. El reloj escribe en la consola, ábrela para verlo.

#### EJERCICIO 51: Error al crear una instancia

Tienes el código en una carpeta.

En el código, la clase **Rabbit** extiende a **Animal**.

Sin embargo, al intentar crear un objeto de la clase **Rabbit**, se produce un error. ¿Por qué? Arréglalo.

#### EJERCICIO 52: Reloj extendido

Tienes el código en una carpeta.

En el código, la clase **Clock** muestra la hora cada segundo.

Crea una nueva clase **ExtendedClock** que herede de **Clock** y agregue el parámetro **precisión**: este es el número de milisegundos entre “tics”. El valor por defecto es **1000** (1 segundo).

* Tu código debe estar en el archivo **extended-clock.js**.
* No modifiques el **clock.js** original. Extiéndelo.

#### EJERCICIO 53: Heredando de Vehículo

Partimos de la clase **Vehiculo** que has creado en un ejercicio anterior.

Crea dos clases más, ambas deben heredar de **Vehículo**:

* **Coche**: cuya velocidad máxima será 120 y sobrescribirá el método **parar** para que muestre “Aparcar”.
* **Avión**: cuya velocidad máxima será 1000 y sobrescribirá los métodos **comenzar**, para que muestre “Despegar”, y **parar**, para que muestre “Aterrizar”.