

## **Ejercicios de POO básicos**

1. Crea un objeto literal llamado coche. Este objeto debe tener las propiedades marca, modelo y año. Además, debe incluir un método llamado obtenerInformacion que devuelva una cadena de texto con todos los detalles del coche.
2. Define una función constructora llamada Libro. Esta función debe aceptar tres parámetros: titulo, autor y paginas. Luego, instancia al menos dos objetos Libro diferentes usando esta función constructora.
3. Añade un método llamado mostrarDetalles a la función constructora Libro del ejercicio anterior. Este método debe devolver una cadena de texto que incluya el título, autor y número de páginas del libro.
4. Dentro del método mostrarDetalles de la función constructora Libro, asegúrate de utilizar la palabra clave this correctamente para acceder a las propiedades del objeto que lo invoca.
5. Agrega un nuevo método a la función constructora Libro llamado cambiarPaginas. Este método debe aceptar un nuevo número de páginas como argumento y actualizar la propiedad paginas del objeto.
6. Crea una función constructora llamada Usuario que reciba nombre y email. Si el email no se proporciona al instanciar un nuevo usuario, el valor por defecto de la propiedad email debe ser "desconocido@ejemplo.com".
7. Refactoriza la función constructora Libro a una clase ES6 llamada Book. La clase debe tener un constructor que acepte titulo, autor y paginas, y asignarlos a las propiedades del objeto. Instancia un objeto de esta nueva clase.
8. Añade un método llamado getDescription a la clase Book (ES6) del ejercicio anterior. Este método debe devolver una cadena de texto similar a la que devolvía el método mostrarDetalles de la función constructora original.
9. Define una nueva clase ES6 llamada Revista que herede de la clase Book. La clase Revista debe añadir una propiedad adicional llamada numero y un método propio llamado getRevistaInfo que devuelva la información completa de la revista, incluyendo su número.
10. Crea una clase llamada Biblioteca. Esta clase debe tener un constructor que inicialice una propiedad interna, un array vacío llamado coleccionDeLibros. Añade un

método a la clase Biblioteca llamado agregarLibro que reciba un objeto Book y lo añada a la coleccionDeLibros. Luego, añade otro método llamado listarLibros que imprima en consola los detalles de cada libro en la colección.

## **Ejercicios con Clases**

EJERCICIO 1: Clase Persona Crea una clase llamada Persona con las propiedades nombre y edad. El constructor debe inicializar estas propiedades. Añade un método llamado saludar que devuelva un string con un saludo, incluyendo el nombre de la persona. Crea dos instancias de la clase Persona y llama a su método saludar.

EJERCICIO 2: Clase Rectangulo Define una clase Rectangulo con las propiedades ancho y alto. El constructor debe aceptar estos dos valores. Implementa un método llamado calcularArea que retorne el área del rectángulo. Implementa un método llamado calcularPerimetro que retorne el perímetro del rectángulo. Crea una instancia de Rectangulo y muestra su área y perímetro.

EJERCICIO 3: Clase CuentaBancaria Crea una clase CuentaBancaria con las propiedades numeroCuenta y saldo. El constructor debe inicializar el numeroCuenta y establecer el saldo inicial en 0. Añade un método depositar(cantidad) que incremente el saldo. Añade un método retirar(cantidad) que decremente el saldo, solo si hay suficiente saldo. Si no hay suficiente, debe imprimir un mensaje de error. Añade un método obtenerSaldo que retorne el saldo actual. Crea una cuenta, realiza depósitos y retiros, y muestra el saldo.

EJERCICIO 4: Clase Coche Define una clase Coche con las propiedades marca, modelo y velocidad. El constructor debe inicializar marca y modelo, y la velocidad debe comenzar en 0. Implementa un método acelerar() que aumente la velocidad en 10. Implementa un método frenar() que disminuya la velocidad en 5, asegurándose de que la velocidad no sea negativa. Crea una instancia de Coche, acelera varias veces, frena y muestra la velocidad actual.

EJERCICIO 5: Clase Libro y Clase Biblioteca Crea una clase Libro con las propiedades título y autor. Crea una clase Biblioteca. Esta clase debe tener una propiedad libros, que será un array vacío en el constructor. Implementa un método agregarLibro(libro) en la clase Biblioteca que reciba un objeto Libro y lo añada al array libros. Implementa un método listarLibros() en la clase Biblioteca que imprima el título y autor de cada libro en la biblioteca. Crea algunos objetos Libro, agrégalos a una instancia de Biblioteca y luego lista todos los libros.

EJERCICIO 6: Herencia - Animal Básico Crea una clase base Animal con las propiedades nombre y especie. El constructor debe inicializar estas propiedades. Añade un método emitirSonido() que imprima "El animal hace un sonido genérico".

EJERCICIO 7: Herencia - Perro y Gato Extiende la clase Animal para crear una clase Perro. La clase Perro debe tener una propiedad adicional raza. El constructor de Perro debe llamar al constructor de Animal usando super() y luego inicializar raza. Sobrescribe el método emitirSonido() en Perro para que imprima "El perro ladra.". Extiende la clase Animal para crear una clase Gato. La clase Gato debe tener una propiedad adicional colorPelo. El constructor de Gato debe llamar al constructor de Animal usando super() y luego inicializar colorPelo. Sobrescribe el método emitirSonido() en Gato para que imprima "El gato maúlla.". Crea instancias de Perro y Gato y llama a emitirSonido en cada una.

EJERCICIO 8: Herencia - Forma y sus Subclases Crea una clase base abstracta (o con un método genérico) Forma con un método llamado calcularArea(). Este método debe imprimir un mensaje indicando que debe ser implementado por las subclases. Crea una clase Circulo que extienda de Forma. Debe tener una propiedad radio. El constructor de Circulo debe inicializar el radio. Implementa el método calcularArea() en Circulo para que retorne el área de un círculo ( $\pi * \text{radio} * \text{radio}$ ). Crea una clase Cuadrado que extienda de Forma. Debe tener una propiedad lado. El constructor de Cuadrado debe inicializar el lado. Implementa el método calcularArea() en Cuadrado para que retorne el área de un cuadrado ( $\text{lado} * \text{lado}$ ). Crea instancias de Circulo y Cuadrado y muestra sus áreas.

EJERCICIO 9: Clase Usuario y Administrador Crea una clase Usuario con las propiedades nombreUsuario y email. El constructor debe inicializar estas propiedades. Añade un método mostrarInfo() que imprima el nombre de usuario y el email. Crea una clase Administrador que extienda de Usuario. El constructor de Administrador debe llamar al constructor de Usuario usando super() y añadir una propiedad adicional nivelAcceso (por ejemplo, "total"). Sobrescribe el método mostrarInfo() en Administrador para que también imprima el nivel de acceso. Crea una instancia de Usuario y otra de Administrador, y llama a mostrarInfo() en ambas.

EJERCICIO 10: Tienda de Productos y Carrito de Compras Crea una clase Producto con las propiedades nombre, precio y id. El constructor debe inicializar estas propiedades. Crea una clase CarritoCompras. Esta clase debe tener una propiedad productos, que será un array vacío en el constructor. Implementa un método agregarProducto(producto) en CarritoCompras que reciba un objeto Producto y lo añada al array productos. Implementa un método eliminarProducto(idProducto) en CarritoCompras que elimine un producto del array por su id. Implementa un método calcularTotal() en CarritoCompras que retorne la suma de los precios de todos los

productos en el carrito. Crea varios productos, agrégalos a un carrito de compras, elimina uno y luego calcula y muestra el total.