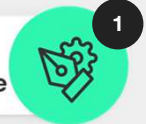


# CLASES

**Formato:** Un documento de texto con nombre de archivo “ApellidoNombre” con que cumpla la siguiente consigna.

Desafío  
entregable



## >> Consigna:

1) Declarar una clase Usuario

2) Hacer que Usuario cuente con los siguientes atributos:

- *nombre*: String
- *apellido*: String
- *libros*: Object[]
- *mascotas*: String[]

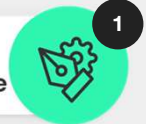
Los valores de los atributos se deberán cargar a través del constructor, al momento de crear las instancias.

**CODER HOUSE**

# CLASES

**Formato:** Un documento de texto con nombre de archivo "ApellidoNombre" con que cumpla la siguiente consigna.

Desafío  
entregable



3) Hacer que Usuario cuente con los siguientes métodos:

- `getFullName(): String`. Retorna el completo del usuario. *Utilizar template strings.*
- `addMascota(String): void`. Recibe un nombre de mascota y lo agrega al array de mascotas.
- `countMascotas(): Number`. Retorna la cantidad de mascotas que tiene el usuario.
- `addBook(String, String): void`. Recibe un string 'nombre' y un string 'autor' y debe agregar un objeto: { nombre: String, autor: String } al array de libros.
- `getBookNames(): String[]`. Retorna un array con sólo los nombres del array de libros del usuario.

4) Crear un objeto llamado usuario con valores arbitrarios e invocar todos sus métodos.

**CODER HOUSE**

# CLASES

## >> Ejemplos:

- ***countMascotas***: Suponiendo que el usuario tiene estas mascotas: ['perro', 'gato']  
usuario.countMascotas() debería devolver 2.
- ***getBooks***: Suponiendo que el usuario tiene estos libros: [{nombre: 'El señor de las moscas', autor: 'William Golding'}, {nombre: 'Fundacion', autor: 'Isaac Asimov'}]  
usuario.getBooks() debería devolver ['El señor de las moscas', 'Fundacion'].
- ***getFullName***: Suponiendo que el usuario tiene: nombre: 'Elon' y apellido: 'Musk'  
usuario.getFullName() debería devolver 'Elon Musk'