

Class Animal

```
id;  
race;  
color;  
age;  
gender;  
name;
```

```
contructor (id, race, color, age, gender, name) {
```

```
this.id = id;  
this.name = name;  
this.race = race;  
this.color = color;  
this.age = age;  
}
```

```
mostrarDatos () {
```

```
console.log ('Animal {', this.name, ', race', this.race, ',  
color, ', this.colors, ', age', this.age, ', ', this.gender);
```

```
animal (comida) {
```

```
console.log ('{', this.name, ', esta comiendo ', comida, '}');
```

```
toJSON () return {
```

```
id: this.id,  
name: this.name,  
race: this.race,  
color: this.color,  
age: this.age,  
gender: this.gender};  
}
```

```
get id () { return this.id; }  
get name () { return name; }  
get race () { return race; }  
get color () { return color; }  
get age () { return age; }  
get gender () { return gender; }  
}
```

```
set id (newid) {
```

```
if (newid) {  
this.id = newid;  
this.Hid = newid; }
```

```
set name (newname) {  
this.name = newname; }
```

```
Set race (newrace) {
    this.race = newrace;
}
```

```
Set color (newcolor) {
    this.color = newcolor;
}
```

```
Set age (newage) {
    this.age = newage;
}
```

```
Set gender (newgender) {
    this.gender = newgender;
}
```

```
Const gat = new Animal(1, 'Gato', 'GATO', 3, 'Macho');
gat.mostrarDatos();
gat.Animal('Nada')
console.log(JSON.stringify(gat));
```

Class Estudiante

```
    Id;  
    nombre;  
    apellido;  
    Edad;  
    Genero;
```

Constructor (nombre, Id, apellido, edad, genero) {

```
    this.Id = Id;
```

```
    this.nombre = nombre;
```

```
    this.apellido = apellido;
```

```
    this.edad = edad;
```

```
    this.genero = genero;
```

5

mostrarDatos () {

```
    System.out.println ("Personas [" + this.Id + "], " + this.nombre + ",  
    " + this.apellido + ", " + this.edad + ", " + this.genero);
```

DarMe (Dame me) {

```
    System.out.println ("Nombre " + this.nombre + " Maria Dame mas que  
    " + DameMe);
```

funcion Persona

return {

id: this.id;
nombre: this.nombre,
apellido: this.apellido,
edad: this.edad,
genero: this.genero,

}

get nombre () { return this.nombre; }

get id () { return this.id; }

get apellido () { return this.apellido; }

get edad () { return this.edad; }

get genero () { return this.genero; }

set id (nuevoId) { this.id = nuevoId; }

set apellido (nuevoApellido) { this.apellido = nuevoApellido; }

set nombre (nuevoNombre) { this.nombre = nuevoNombre; }

set edad (nuevaEdad) { this.edad = nuevaEdad; }

set genero (nuevoGenero) { this.genero = nuevoGenero; }

}

}

const juan = new Persona (1, "Juan", "Rodríguez", 17, "M")

console.log (JSON.stringify (juan));

Juan { id: 1, nombre: "Juan", apellido: "Rodríguez", edad: 17, genero: "M" }

Juan.getNombre ()

class Hospital {

 id;
 nombre;
 direccion;
 cantidad_sillas;
 tiene_planta;

 Constructor C id, nombre, direccion, cantidad_sillas,
 tiene_planta)

 this.id = id; id;

 this.nombre = nombre;

 this.direccion = direccion;

 this.cantidad_sillas = cantidad_sillas;

 this.tiene_planta = tiene_planta;

3

mostrarDatos () {

 console.log ('Hospital #E this.id S, nombre
 S this.nombre3, direccion S this.direccion
 cantidad_sillas #E this.cantidad_sillas3,
 tiene_planta #E this.tiene_plantas');

3

planta (tiene_planta) {

 if (tiene_planta) {

 return 'Este hospital si tiene
 planta'

 } else {

 return 'Este hospital no tiene
 planta'

3

3

tojson()

id: this.id,
nombre: this.nombre,
direccion: this.direccion;
cantidad - salas: this.sala,
tiene - planta: this.tiene_planta

;

3.

get id() {
return this.id;

get nombre() { return this.nombre;

3 get direccion() { return this.direccion;

3 get cantidad() { return this.cantidad;

3 get tiene_planta() { return this.tiene_planta;

3 set id(record) { this.id = record;

3 set nombre(record) { this.nombre = record;

const Coramios = new Hospital ('C9', 'coramios')
calle 43 C Sur c3-c4', 30, true);

Const Pulmios = new Hospital ('C', 'pulmios',
Calle, 25, 26 "", 44), false);

Cancunia. mostrarDatos();

Console.log (los_Pulmos.tienePlanta));

Console.log (JSON.stringify(Cancunia.tiene_plant

clase Instrumento {

 id:
 costo;
 nombre;
 electrico;

 constructor (id, costo, nombre, electrico) {
 this.id = id;
 this.costo = costo;
 this.nombre = nombre;
 this.electrico = electrico;
 }

5

mostrar informacion ()

 Console.log (id + " " + this.id)

6

 electrico (es-electrico)

 return \$&this.nombre es un instrumento electrico

7 else 3

 return \$&this.nombre " no es electrico";

8

9

10 son 6

 return 5

 id: this.id,
 nombre: this.nombre;
 costo: costo & costo;
 electrico: this.electrico;

3,

3 /

clase Instrumento {

 id;
 costo;
 nombre;
 electrico;

 constructor (id, costo, nombre, electrico) {

 this.id = id;

 this.costo = costo;

 this.nombre = nombre;

 this.electrico = electrico;

5

mostrar informacion ()

 Console.log (C.id + " " + this.id)

6

electrico (es-electrico)

 return \$&this.nombre es un instrumento electrico

7 es 3

 return \$&this.nombre \$ no es electrico;

8

S

9 SON S

 return S

 id: this.id,

 nombre: this.nombre;

 costo: costo + costo;

 electrico: this.electrico;

10 :

11 /

```

get id () {
    return this.id = id;
}

get nombre () {
    return this.nombre = nombre;
}

get costo () {
    return this.costo = costo;
}

set electrico () {
    return this.electrico = electrico;
}

set id (nuevoId) {
    this.id = nuevoId;
}

set nombre (nuevoNombre) {
    this.nombre = nuevoNombre;
}

set costo (nuevoCosto) {
    this.costo = nuevoCosto;
}

set electrico (nuevoElectrico) {
    this.electrico = nuevoElectrico;
}

const guitarra = new Instrumento ('Guitarra clásica', 'Guitarra eléctrica', 5000);
const violin = new Instrumento ('Violín', 'Violín eléctrico', false, 300);

guitarra.mostrarInformacion();
violin.mostrarInformacion();

console.log (guitarra.electrico);
console.log (violin.electrico);

console.log (JSON.stringify (guitarra));

```

id
nombre;
apellido;
edad;
genero;

contructor (id, apellido, nombre, edad, genero)
this.id = id;
this.nombre = nombre;
this.apellido = apellido;
this.edad = edad;
this.genero = genero;

3
mostrar datos () {

console.log ('Persona #' + this.id + ', nombre ' + this.nombre + ', apellido ' + this.apellido + ', edad ' + this.edad + ', genero ' + this.genero) ;

Dicime (Dime me) {

console.log ('Este nombre ' + this.nombre + ' mana así en mi ge' +
'f' + this.nombre + ',');

4

toJSON () {

return {
id: this.id;
nombre: this.nombre;
apellido: this.apellido;
edad: this.edad;
genero: this.genero};

get id () {

return this.edad; id;

```
get nombre () {  
    return nombre; this.nombre}
```

```
set id (creo id) {  
    this.id = newid}
```

```
get Nombre (creaNombre) {  
    this.nombre = newNombre; }
```

5

```
const Maria = new Person (1, 'Maria', 'Rodriguez',  
console.log  
    'Femenino);  
JSON.stringify (Maria));
```

```
Maria.mostrarDatos();
```

```
Maria.dormir ('nada')
```