

1. Establece el color de texto (etiqueta).

```
function draw() {  
  image(img, 0, 0, 640, 420);  
  fill("#FF0000");  
  text("Perro", 45, 75);  
  noFill();  
  stroke("#FF0000");  
  rect(30, 60, 450, 350 );  
  
  fill("#FF0000");  
}
```

2. Código para el texto (etiqueta).

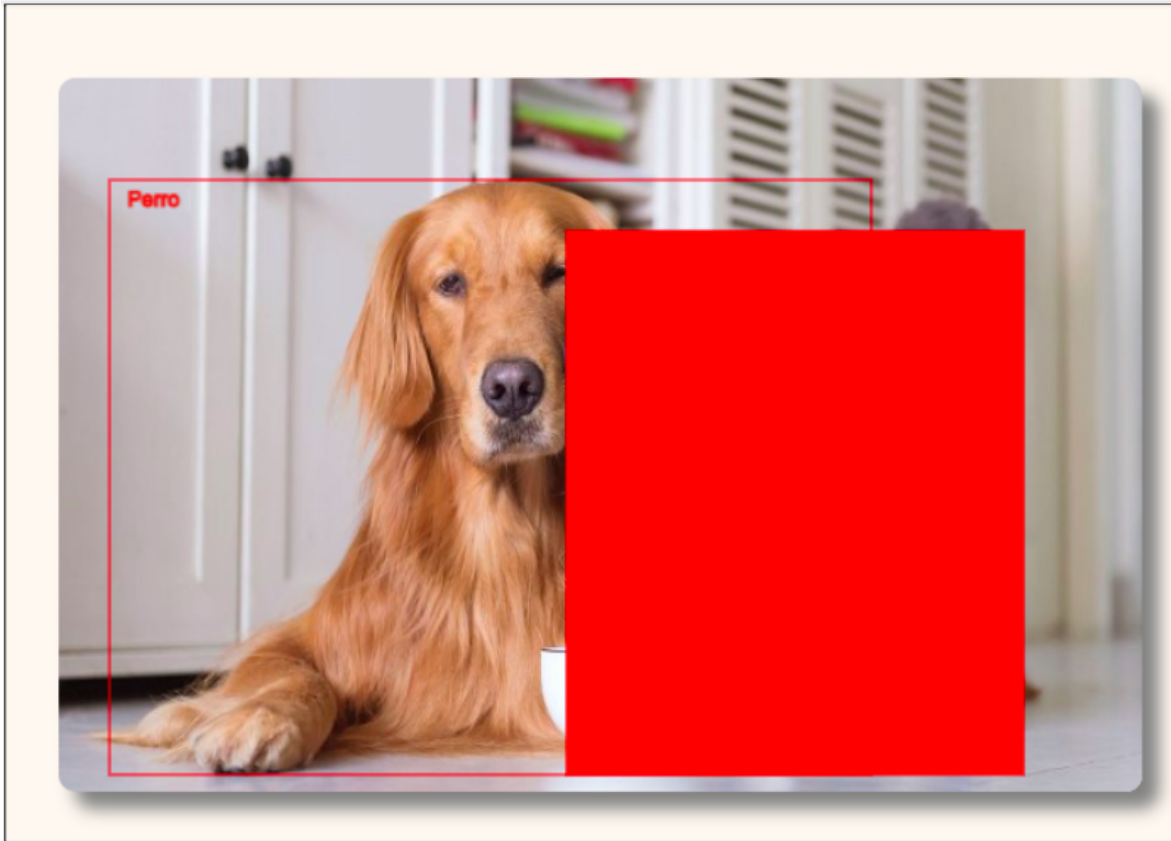
```
function draw() {  
  image(img, 0, 0, 640, 420);  
  fill("#FF0000");  
  text("Perro", 45, 75);  
  noFill();  
  stroke("#FF0000");  
  rect(30, 60, 450, 350 );  
  
  fill("#FF0000");  
  text("Gato", 320, 120);  
}
```

**Output:**



3. **Desestablece la función fill().**

**Si no desestablecemos la función fill(), el output será este:**



**Código para modificar la función fill().**

```
function draw() {  
  image(img, 0, 0, 640, 420);  
  fill("#FF0000");  
  text("Perro", 45, 75);  
  noFill();  
  stroke("#FF0000");  
  rect(30, 60, 450, 350 );  
  
  fill("#FF0000");  
  text("Gato", 320, 120);  
  noFill();  
}
```

4. **Establece el color de borde del rectángulo.**

```
function draw() {
  image(img, 0, 0, 640, 420);
  fill("#FF0000");
  text("Perro", 45, 75);
  noFill();
  stroke("#FF0000");
  rect(30, 60, 450, 350 );

  fill("#FF0000");
  text("Gato", 320, 120);
  noFill();
  stroke("#FF0000");
}
```

## 5. Dibuja el rectángulo.

```
function draw() {
  image(img, 0, 0, 640, 420);
  fill("#FF0000");
  text("Perro", 45, 75);
  noFill();
  stroke("#FF0000");
  rect(30, 60, 450, 350 );

  fill("#FF0000");
  text("Gato", 320, 120);
  noFill();
  stroke("#FF0000");
  rect(300, 90, 270, 320 );
}
```

**Output:**



**Ahora empezaremos a programar para inicializar y ejecutar el modelo CocoSsd.**

1. **Código para inicializar el modelo CocoSsd.**

```
function setup() {  
  canvas = createCanvas(640, 420);  
  canvas.center();  
  objectDetector = ml5.objectDetector('cocossd', modelLoaded);  
}
```

2. **Código para actualizar el estado como “Estado: detectando objetos”.**

```
function setup() {
  canvas = createCanvas(640, 420);
  canvas.center();
  objectDetector = ml5.objectDetector('cocossd', modelLoaded);
  document.getElementById("status").innerHTML = "Estatus: detectando objetos";
}
```

3. Define una variable de cadena vacía para contener el estado.

```
img = "";
status = "";

function preload(){
  img = loadImage('dog_cat.jpg');
}
```

4. Código de la función modelLoaded().

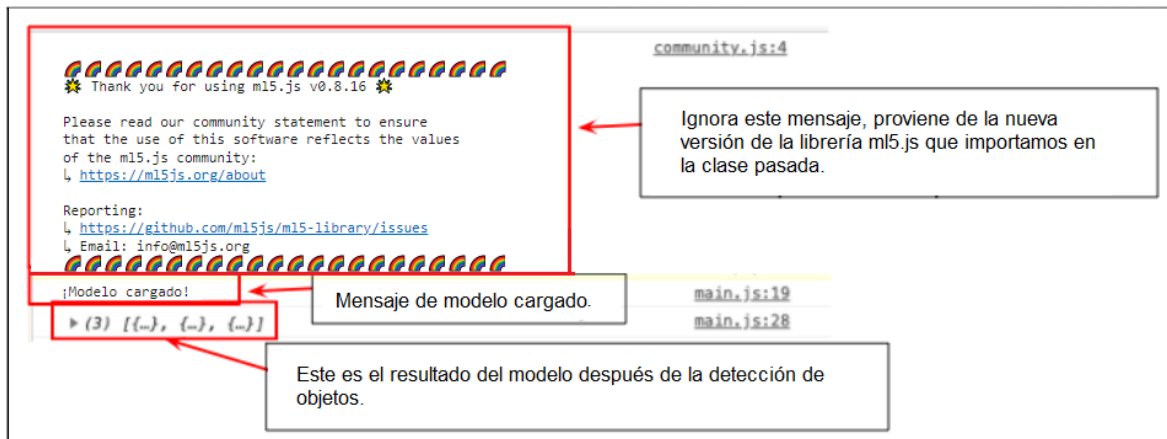
```
function modelLoaded() {
  console.log("¡Modelo cargado!");
  status = true;
  objectDetector.detect(img, gotResult);
}
```

5. Código de la función gotResult().

```
function modelLoaded() {
  console.log("¡Modelo cargado!");
  status = true;
  objectDetector.detect(img, gotResult);
}

function gotResult(error, results) {
  if (error) {
    console.log(error);
  }
  console.log(results);
}
```

Output en la pantalla de la consola:



**NOTA:** Este modelo se va actualizando, por lo que puede que no detecte ninguna de las cosas mencionadas y dé 1, 2 o 3 arrays, así que no te preocupes.

**Expande el array.**



Este array contendrá 3 arrays.

```
main.js:28
▼ (3) [{...}, {...}, {...}] ⓘ
  ▶ 0: {label: "cat", confidence: 0.8548185229301453, x: 375.55742263793945, ...}
  ▶ 1: {label: "dog", confidence: 0.6707387566566467, x: 27.80470848083496, ...}
  ▶ 2: {label: "bowl", confidence: 0.5682403445243835, x: 357.06048011779785, ...}
    length: 3
  ▶ __proto__: Array(0)
```

**Datos dentro del primer array.**

```
main.js:28
▼ (3) [{...}, {...}, {...}]
  ▼ 0:
    confidence: 0.8548185229301453
    height: 352.57424265146255
    label: "cat"
    ▶ normalized: {x: 0.4694467782974243, y: 0.16424188017845154, width: 0.4...
      width: 341.9727325439453
      x: 375.55742263793945
      y: 73.90884608030319
    ▶ __proto__: Object
  ▶ 1: {label: "dog", confidence: 0.6707387566566467, x: 27.80470848083496, ...
  ▶ 2: {label: "bowl", confidence: 0.5682403445243835, x: 357.06048011779785...
    length: 3
  ▶ __proto__: Array(0)
```

El primer array contiene:

1. Confidence (precisión).
2. 'Height' es la altura del objeto.
3. 'Label' (etiqueta) es el nombre del objeto detectado.
4. 'Normalized' es otro array que contiene las coordenadas X y Y, la altura y el ancho del objeto. Pero en forma decimal, y **no** necesitamos eso.
5. 'Width' es el ancho del objeto.
6. X: es el valor de la coordenada X del objeto situado en la imagen.
7. Y: es el valor de la coordenada Y del objeto situado en la imagen.