

Fundamentos de la Web

# Colección de ejercicios



Universidad  
Rey Juan Carlos

**Micael Gallego**

Correo: [micael.gallego@urjc.es](mailto:micael.gallego@urjc.es)

Twitter: [@micael\\_gallego](https://twitter.com/micael_gallego)

**Nicolás Rodríguez**

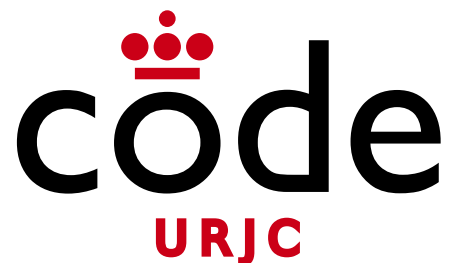
Correo: [nicolas.rodriguez@urjc.es](mailto:nicolas.rodriguez@urjc.es)

**Iván Chicano**

[ivan.chicano@urjc.es](mailto:ivan.chicano@urjc.es)

**Michel Maes**

[michel.maes@urjc.es](mailto:michel.maes@urjc.es)



©2023

Micael Gallego, Nicolás Rodríguez, Iván Chicano, Michel Maes

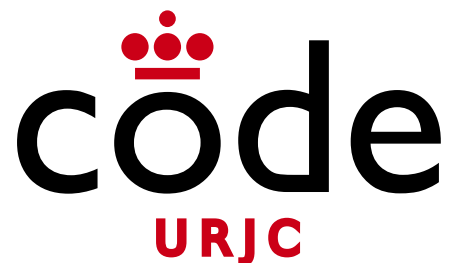
Algunos derechos reservados

Este documento se distribuye bajo la licencia  
“Atribución-CompartirIgual 4.0 Internacional”  
de Creative Commons Disponible en  
<https://creativecommons.org/licenses/by-sa/4.0/deed.es>

# Índice de Pruebas de Evaluación

## Índice

- La asignatura se compone de 5 bloques
  - Bloque I: Introducción a la web (No hay ejercicios)
  - Bloque II: Tecnologías de cliente web
  - Bloque III: Tecnologías de servidor web
  - Bloque IV: Tecnologías de interactividad en el cliente web
  - Bloque V: Web semántica (No hay ejercicios)

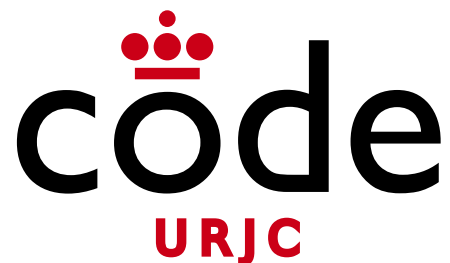


## Fundamentos de la Web

# Bloque II: Tecnologías de cliente web

## Tema 2.1: HTML





©2023

Micael Gallego, Nicolás Rodríguez, Iván Chicano, Michel Maes

Algunos derechos reservados

Este documento se distribuye bajo la licencia  
“Atribución-CompartirIgual 4.0 Internacional”  
de Creative Commons Disponible en  
<https://creativecommons.org/licenses/by-sa/4.0/deed.es>

# Índice de ejercicios

- Ejercicio 1

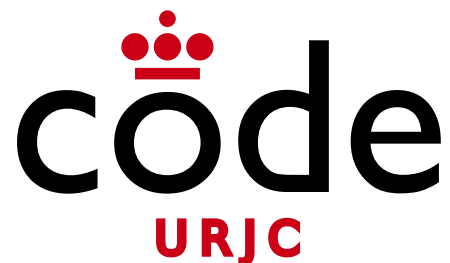
# Ejercicio 1

- Crea una página HTML (sin incluir estilos)
- El contenido puede ser de cualquier temática:  
*Una empresa, un grupo de música, un departamento de la universidad, un producto software, un teléfono móvil, una asociación cultural, etc...*
- El contenido puede ser inventado o real (obtenido de una página de Internet)

# Ejercicio 1

- La página debe incluir, al menos:
  - Varias secciones con subsecciones
  - Fotografías/Imágenes
  - Listas de elementos
  - Una o varias tablas
  - Hiperenlaces



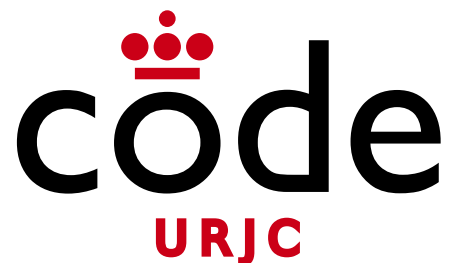


## Fundamentos de la Web

# Bloque II: Tecnologías de cliente web

## Tema 2.2: CSS





©2023

Micael Gallego, Nicolás Rodríguez, Iván Chicano, Michel Maes

Algunos derechos reservados

Este documento se distribuye bajo la licencia  
“Atribución-CompartirIgual 4.0 Internacional”  
de Creative Commons Disponible en  
<https://creativecommons.org/licenses/by-sa/4.0/deed.es>

# Índice de ejercicios

- Ejercicio 1
- Ejercicio 2
- Ejercicio 3

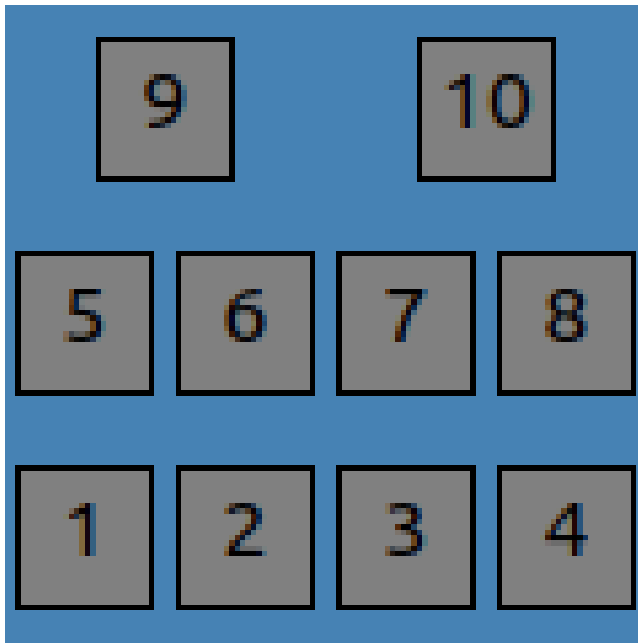
# Ejercicio 1

- Crea una página web con HTML y CSS con el siguiente botón



# Ejercicio 2

- Trata de replicar la siguiente figura utilizando solo CSS (*Flexbox*)



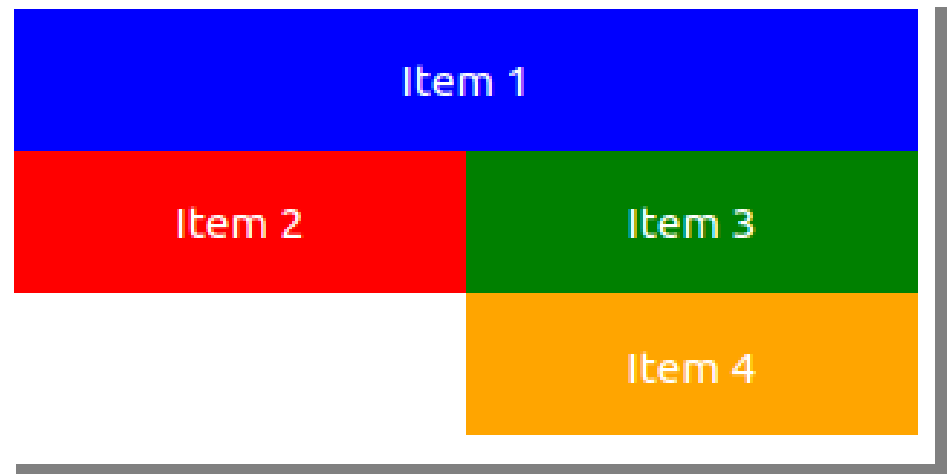
```
<div class="container">  
  <div class="item">1</div>  
  <div class="item">2</div>  
  <div class="item">3</div>  
  <div class="item">4</div>  
  <div class="item">5</div>  
  <div class="item">6</div>  
  <div class="item">7</div>  
  <div class="item">8</div>  
  <div class="item">9</div>  
  <div class="item">10</div>  
</div>
```

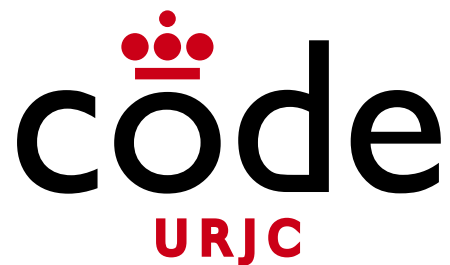
# Ejercicio 3

- Trata de replicar la siguiente figura utilizando solo CSS (Grid)

```
<div class="grid">  
  <div class="item a">Item 1</div>  
  <div class="item b">Item 2</div>  
  <div class="item c">Item 3</div>  
  <div class="item d">Item 4</div>  
</div>
```

```
.grid {  
  display: grid;  
}  
.item {  
  color: white;  
  padding: 1em;  
  text-align: center;  
}  
.a { background: blue; }  
.b { background: red; }  
.c { background: green; }  
.d { background: orange; }
```



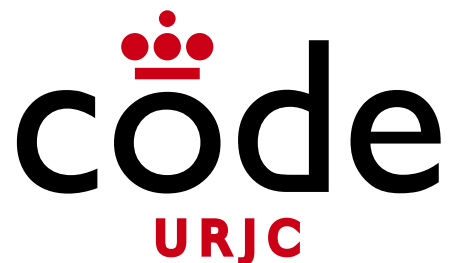


## Fundamentos de la Web

# Bloque II: Tecnologías de cliente web

## Tema 2.3: Bootstrap





©2023

Micael Gallego, Nicolás Rodríguez, Iván Chicano, Michel Maes

Algunos derechos reservados

Este documento se distribuye bajo la licencia  
“Atribución-CompartirIgual 4.0 Internacional”  
de Creative Commons Disponible en  
<https://creativecommons.org/licenses/by-sa/4.0/deed.es>

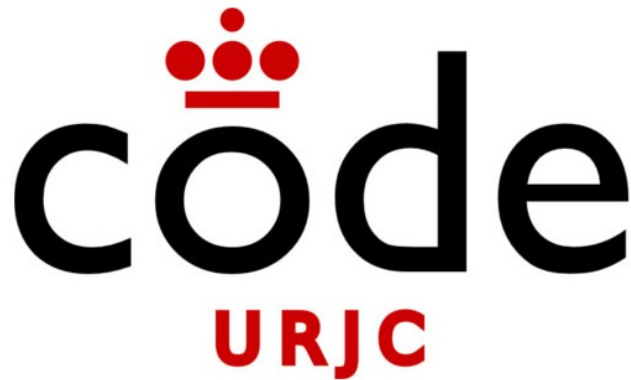


# Índice de ejercicios

- Ejercicio 1

# Ejercicio 1

- Replica la siguiente página web utilizando componentes *Bootstrap*
  - *Url de la imagen: [https://www.codeurjc.es/img/codeurjc\\_logo.jpg](https://www.codeurjc.es/img/codeurjc_logo.jpg)*
  - *Será necesario customizar los estilos de Bootstrap*



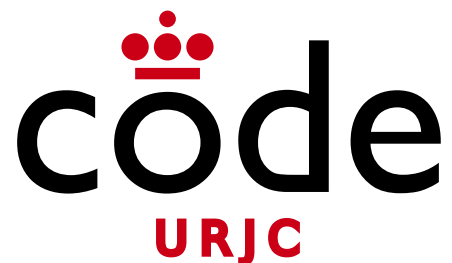
## Bienvenidos

Esta es la página principal de `CodeURJC`

## Profesores

Nombre	Apellido	Email
Micael	Gallego	micael.gallego@urjc.es
Iván	Chicano	ivan.chicano@urjc.es
Michel	Maes	michel.maes@urjc.es

Cargar más

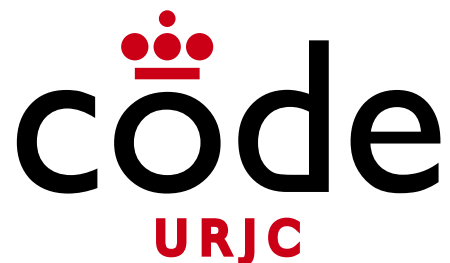


## Fundamentos de la Web

### Bloque III: Tecnologías de servidor web

# Tema 3.1: Introducción a JavaScript y Node





©2023

Micael Gallego, Iván Chicano, Michel Maes, Pablo Fuente

Algunos derechos reservados

Este documento se distribuye bajo la licencia  
"Atribución-CompartirIgual 4.0 Internacional"  
de Creative Commons Disponible en

<https://creativecommons.org/licenses/by-sa/4.0/deed.es>

# Índice de Ejercicios

- Ejercicio 1
- Ejercicio 2
- Ejercicio 3
- Ejercicio 4
- Ejercicio 5
- Ejercicio 5b
- Ejercicio 6
- Ejercicio 7

# Ejercicio 1

## Hello world!

Ejercicio1

- Crear una carpeta
- Crear un fichero **app.js**

```
console.log('Hello world!');
```

- Ejecutar el comando

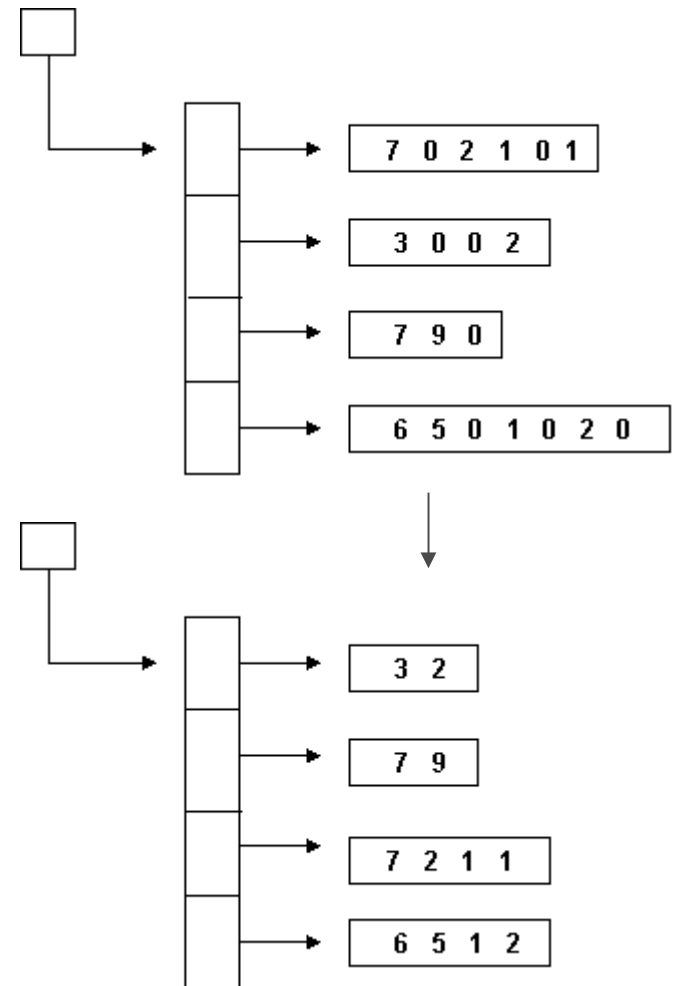
```
$ node app.js
```

## Ejercicio 2

- Crear una función que **reciba un array como parámetro** y devuelva un array de dos elementos
- El primer elemento apuntará a un **array con los números pares** del array que se pasa como parámetro
- El segundo elemento apuntará a un **array con los números impares** del array que se pasa como parámetro
- Para probar la función se implementarán varias llamadas con **diferentes arrays** y el resultado se mostrará en la **consola**

# Ejercicio 3

- Crear una función que **reciba un array bidimensional, le quite los ceros y ordene las filas de menor a mayor longitud**
- Para probar la función se implementarán varias llamadas con **diferentes arrays** y el resultado se mostrará en la **consola**





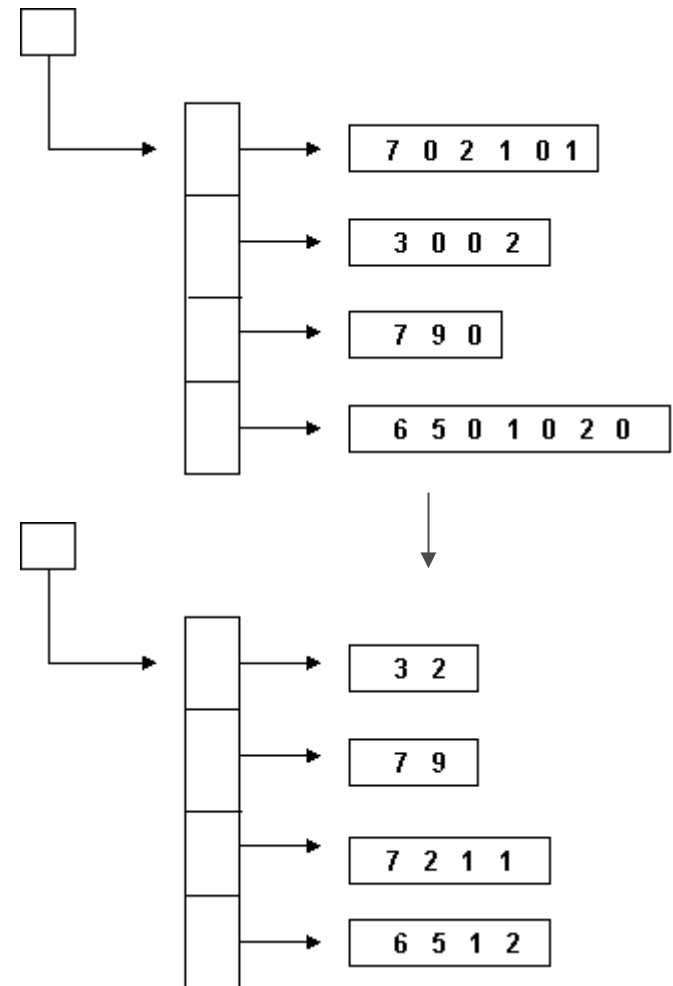
# Ejercicio 3

- Para ordenar los arrays por tamaño se puede usar el **algoritmo de la burbuja** (*bubble sort*)
- Ejemplo: Ordenación de un array de números (tiene que adaptarse para ordenar arrays por su tamaño)

```
function sortNumbers(numbers) {
  for (let i = 0; i < numbers.length; i++) {
    for (let j = 0; j < numbers.length-1; j++) {
      if (numbers[j] > numbers[j + 1]) {
        let temp = numbers[j];
        numbers[j] = numbers[j + 1];
        numbers[j + 1] = temp;
      }
    }
  }
  return numbers;
}
```

# Ejercicio 4

- Haz que el ejercicio 3 sea configurable
- En vez de eliminar los ceros de los arrays, que elimine los números que decida el usuario pasando una función como parámetro
- Usa arrow functions en las pruebas que llaman a `quitaNumerosYOrdena(...)`



# Ejercicio 5

- Se quiere implementar un programa en JavaScript que calcule el área y perímetro de un rectángulo
- El rectángulo se representará como un objeto con los atributos:
  - color
  - alto
  - ancho
- Y métodos:
  - `area(): largo * ancho`
  - `perimetro(): 2 * ancho + 2 * largo`

# Ejercicio 5b

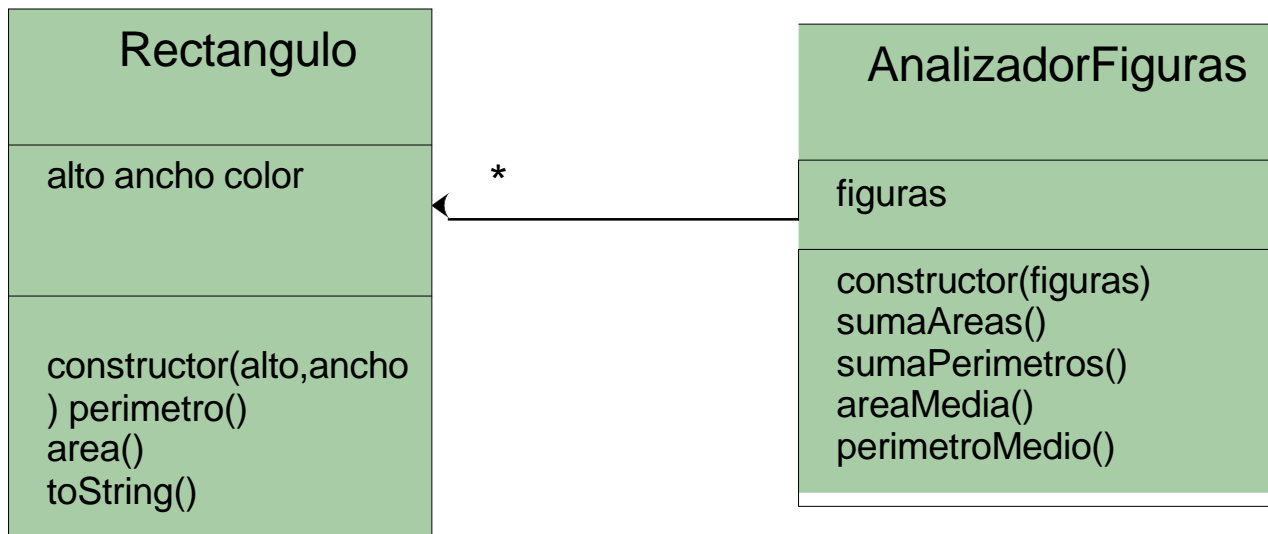
- Se quiere implementar un programa en JavaScript que permita analizar un array de rectángulos
- De cada rectángulo se debe conocer:
  - Color Alto Ancho
  - Área:  $\text{largo} * \text{ancho}$
  - Perímetro:  $2 * \text{ancho} + 2 * \text{largo}$
  - 
  -

# Ejercicio 5b

- Los **análisis** serán:
  - Suma total de áreas de los rectángulos
  - Suma total de perímetros de los rectángulos
  - Área media
  - Perímetro medio
  
- Prueba del **correcto funcionamiento**:
  - Construir un array con diferentes rectángulos
  - Ejecutar todos los análisis
  - Mostrar el resultado en la consola

# Ejercicio 6

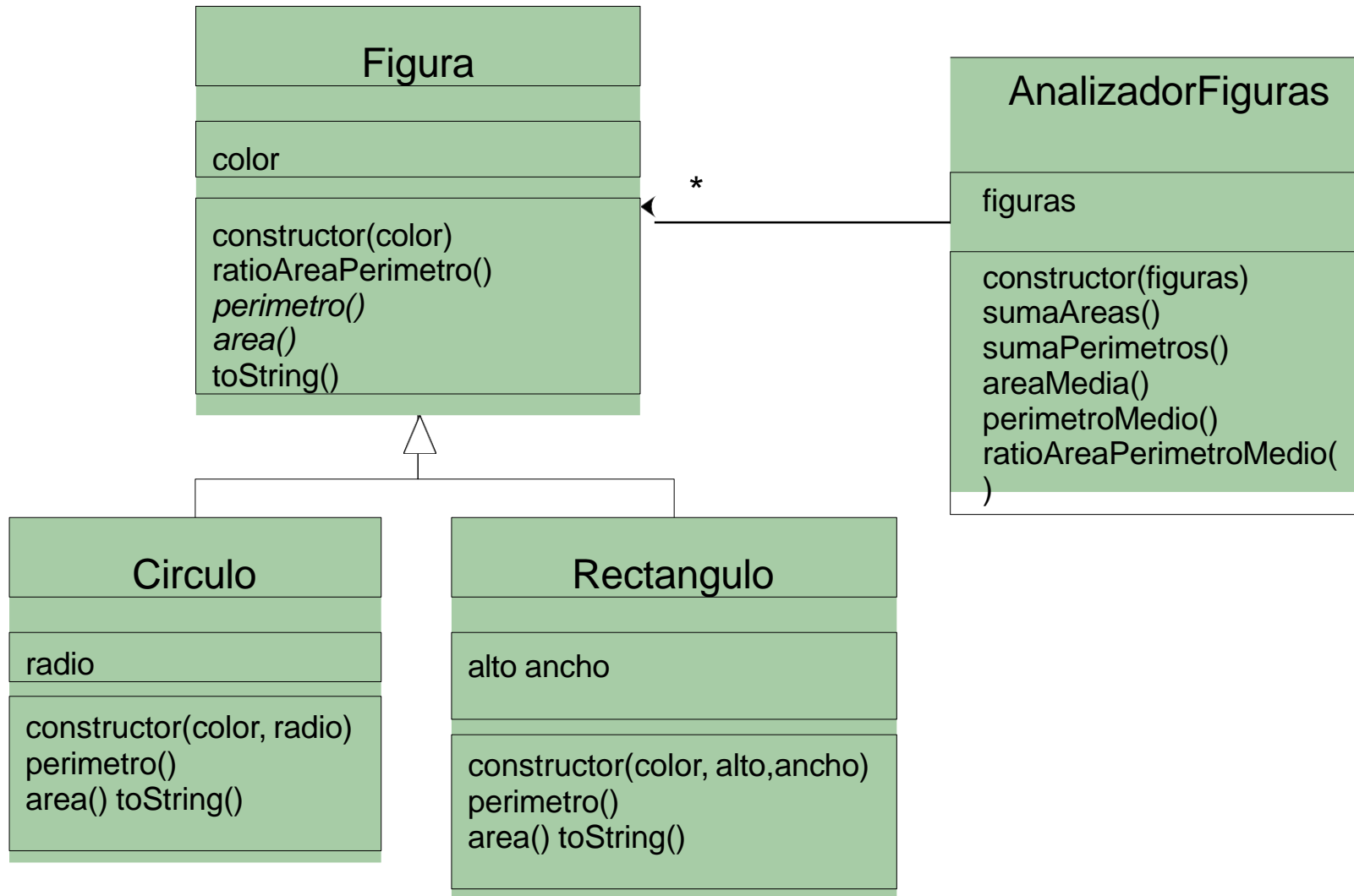
- Transforma el Ejercicio 5 para que use clases
- Agrupa las funciones de análisis en la clase AnalizadorFiguras



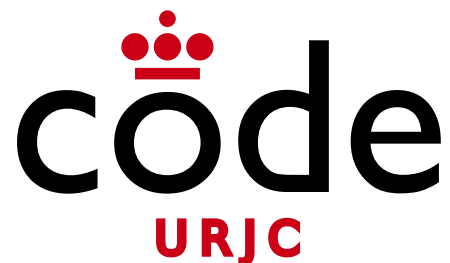
# Ejercicio 7

- Amplía el Ejercicio 6 con las siguientes funcionalidades:
  - Añadir el círculo como nuevo tipo de figura
    - Área:  $\text{Math.PI} * \text{radio} * \text{radio}$
    - Perímetro:  $2 * \text{Math.PI} * \text{radio}$
  - Añadir un nuevo análisis:
    - Por cada figura saber el ratio area/perímetro
    - La media del ratio área/perímetro para todas las figuras

# Ejercicio 7



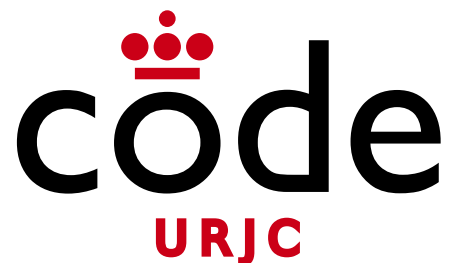




## Fundamentos de la Web

# Bloque III: Tecnologías de servidor web

## Tema 3.2: Estructuras de datos en JavaScript



©2023

Micael Gallego, Iván Chicano, Michel Maes

Algunos derechos reservados

Este documento se distribuye bajo la licencia  
“Atribución-CompartirIgual 4.0 Internacional”  
de Creative Commons Disponible en  
<https://creativecommons.org/licenses/by-sa/4.0/deed.es>

# Índice de ejercicios

- Ejercicio 1
- Ejercicio 2
- Ejercicio 3

# Ejercicio 1

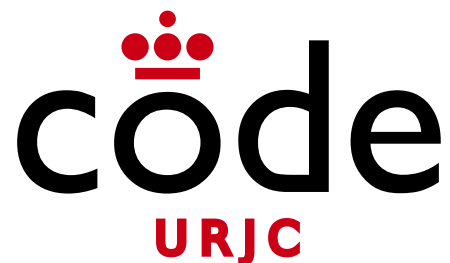
- Implementar una función **cuentaDistintas** que reciba un array con nombres de provincias (que podrían estar repetidas) e indique cuántas provincias sin repetir hay en ese array
- Ejecuta la función con un array de ejemplo para verificar que funciona como se espera

## Ejercicio 2

- Se pide crear una clase **GestorAeropuertos** que permita almacenar aeropuertos (nombre y ciudad)
- Se desea poder acceder a la información completa de un aeropuerto por su nombre de la forma más eficiente posible
- Se debe declarar como atributo de GestorAeropuertos la estructura de datos más eficiente
- Introducir varios aeropuertos y verificar el funcionamiento correcto cuando se accede a uno de ellos usando su nombre

# Ejercicio 3

- Implementar una aplicación que permita gestionar en memoria un conjunto de viajes de una aerolínea
- Cada viaje se representa con la ciudad origen, destino y la duración del viaje
- Se dan de alta los viajes en un gestor (clase **GestorViajes**)
- Al gestor de viajes se le pueden pedir:
  - Devolver un array con los viajes que tienen una determinada ciudad origen
  - Devolver un array con los viajes que tienen una determinada ciudad destino
  - Devolver un array con los viajes
  - Devolver un array con las ciudades en las que hay viajes
- Hay que conseguir el **menor tiempo de ejecución** de las consultas, aunque sean necesarias varias estructuras de datos

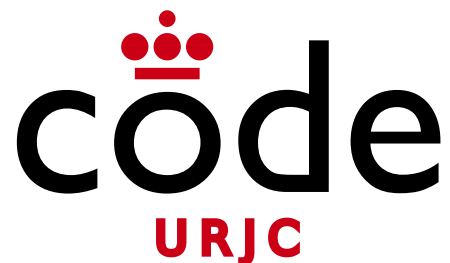


## Fundamentos de la Web

# Bloque III: Tecnologías de servidor web

## Tema 3.3: Módulos en JavaScript





©2023

Micael Gallego, Iván Chicano, Michel Maes

Algunos derechos reservados

Este documento se distribuye bajo la licencia  
"Atribución-CompartirIgual 4.0 Internacional"  
de Creative Commons Disponible en

<https://creativecommons.org/licenses/by-sa/4.0/deed.es>

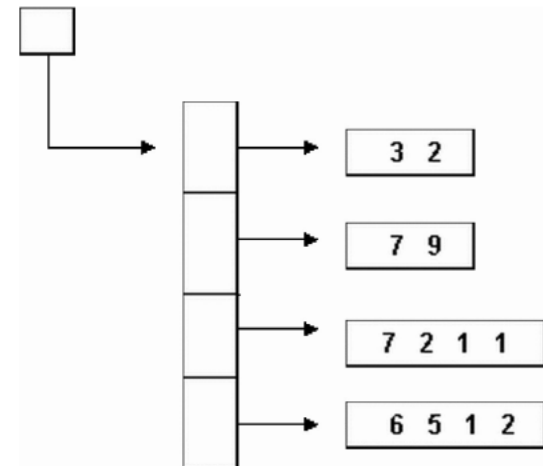
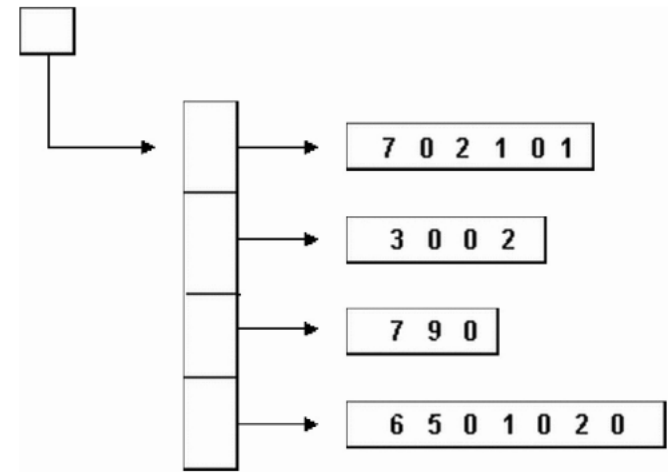


# Índice de ejercicios

- Ejercicio 1
- Ejercicio 2
- Ejercicio 3

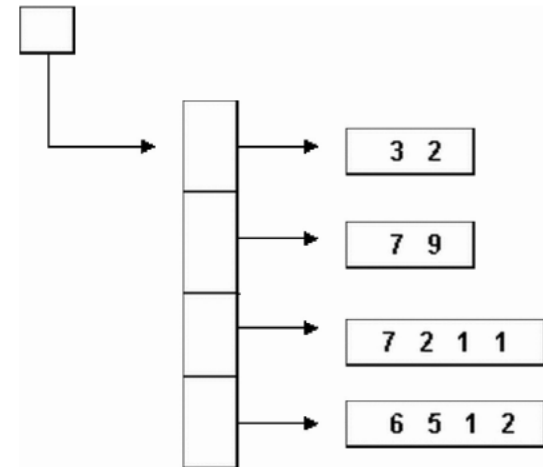
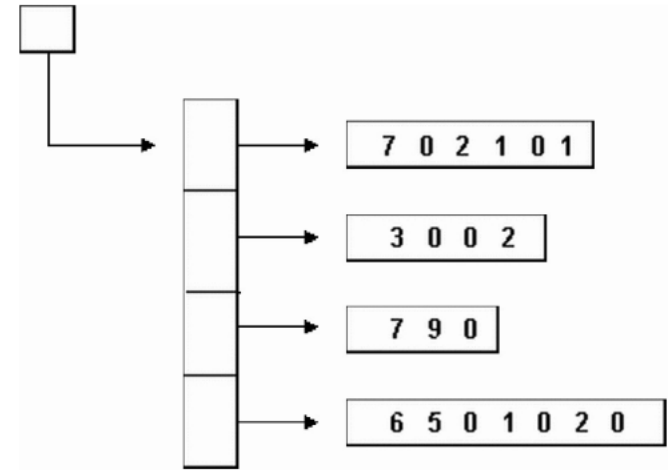
# Ejercicio 1

- Modifica el enunciado para incluir la función de quitar ceros en un módulo
- En el fichero **app.js** se usará ese módulo
- Exporta por **defecto** la función "quitaCeros" desde un módulo



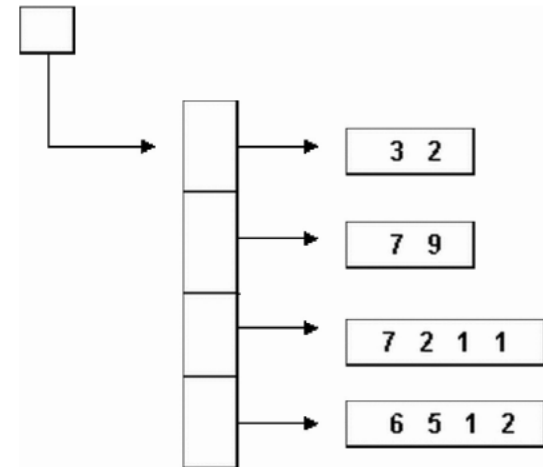
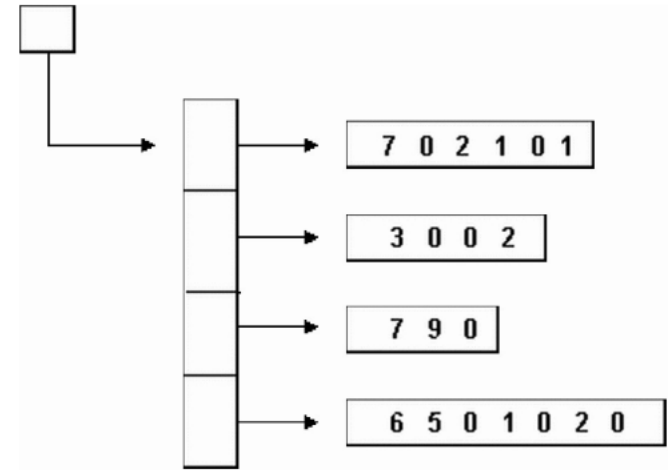
# Ejercicio 2

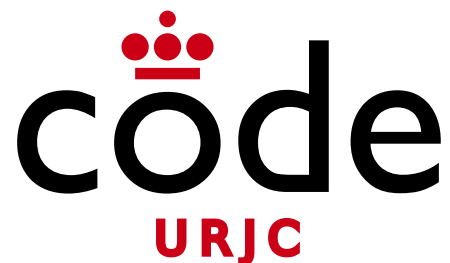
- Reimplementa el ejercicio 1 para usar lodash para eliminar los ceros de los arrays



# Ejercicio 3

- Cambia el ejercicio 2 para que use módulos ES en vez de CommonJS



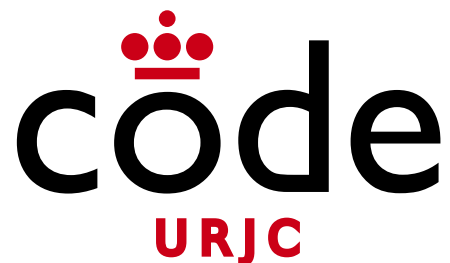


Fundamentos de la Web

Bloque III: Tecnologías de servidor web

# Tema 3.2: Estructuras de datos en JavaScript





©2023

Micael Gallego, Iván Chicano, Michel Maes

Algunos derechos reservados

Este documento se distribuye bajo la licencia  
"Atribución-CompartirIgual 4.0 Internacional"  
de Creative Commons Disponible en

<https://creativecommons.org/licenses/by-sa/4.0/deed.es>

# Ejercicio 1 – Tablón de mensajes

- Crear una aplicación web para gestionar un tablón de anuncios con varias páginas
- La página principal muestra los anuncios existentes (sólo nombre y asunto) y un enlace para insertar un nuevo anuncio
- Si pulsamos en la cabecera de un anuncio se navegará a una página nueva que muestre el contenido completo del anuncio

# Ejercicio 1 – Tablón de mensajes

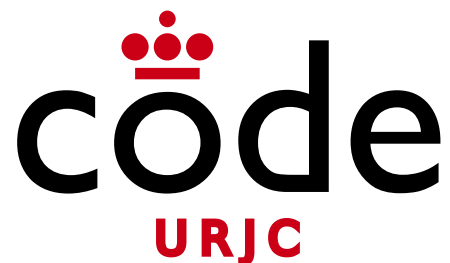
- Si se pulsa el enlace para añadir el anuncio se navegará a una nueva página que contenga un formulario
- Al enviar el formulario se guardará el nuevo anuncio y se mostrará una página indicando que se ha insertado correctamente y un enlace para volver
- En la página del anuncio se podrá borrar



# Ejercicio 1 – Tablón de mensajes

- **Implementación**

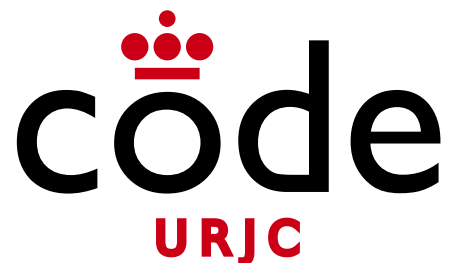
- Se usará un único router con varias funciones (cada una atendiendo una URL diferente)
- Se creará un módulo independiente para la gestión de los posts
- Cada post tendrá asociado un id generado con un contador



## Fundamentos de la Web

### Bloque IV: Tecnologías de interactividad en el cliente web

# Tema 4.1: HTML interactivo con JavaScript



©2023

Micael Gallego, Iván Chicano, Michel Maes

Algunos derechos reservados

Este documento se distribuye bajo la licencia  
"Atribución-CompartirIgual 4.0 Internacional"  
de Creative Commons Disponible en

<https://creativecommons.org/licenses/by-sa/4.0/deed.es>

# Índice de ejercicios

- Ejercicio 1
- Ejercicio 2
- Ejercicio 3
- Ejercicio 4
- Ejercicio 5
- Ejercicio 6
- Ejercicio 7

# Ejercicio 1

- Crea una página web que muestre en la consola "Hola Mundo!" al cargarse en un navegador web

## Ejercicio 2

- Implementa una web que muestre los títulos de unos libros
  - Cien años de soledad
  - El señor de los anillos
  - 1984
  - Un mundo feliz
- Los títulos de los libros estarán almacenados en un array de JavaScript

# Ejercicio 3

- Crea una página con un botón que oculte el texto “Hola Mundo” cuando se pulse
- Si se pulsa de nuevo, se volverá a mostrar el texto “Hola Mundo”

## Ejercicio 4

- Amplía el Ejercicio 2 para que al lado de cada libro aparezca un botón [Más info]
- Al pulsar el botón aparecerá debajo del título el autor del libro y su año de publicación
- Un elemento se puede ocultar con el estilo "display:none" y mostrarse de nuevo con "display:block"



## Ejercicio 4

- Cuando el código HTML es complejo se pueden usar *template literals*
  - Se usan comilla ` (*backtick*) para multilinea
  - Se usa `${...}` para las variables

```
let html =  
  `

<p>${libro.titulo}</p>  
    <p> ${libro.autor} (${libro.año})</p>  
  </div>`;


```

# Ejercicio 4

- Información libros

- Cien años de soledad, de Gabriel García Márquez (1967)

El señor de los anillos, de J. R. R. Tolkien (1954)

- 1984, de George Orwell (1949)

Un mundo feliz, de Aldous Huxley (1932)

# Ejercicio 5

- Actualiza el ejercicio 3 para no generar el HTML como texto y usar la API de nodos del DOM

# Ejercicio 6

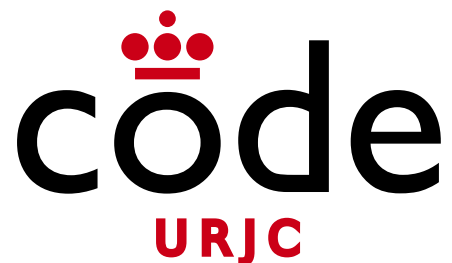
- Amplía el ejercicio 5 para que se puedan dar de alta nuevos libros con un formulario

# Ejercicio 7

- Crea una página que muestre el título de varios libros
- Al lado de cada título aparecerá el botón “Más info”
- Al pulsar el botón se borrará la lista de libros y aparecerá la información del libro junto con un botón “Volver a la lista”

# Ejercicio 7

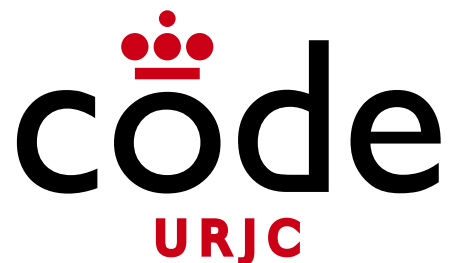




## Fundamentos de la Web

# Bloque III: Tecnologías de interactividad en el cliente web

## Tema 4.2: AJAX



©2023

Micael Gallego, Iván Chicano, Michel Maes

Algunos derechos reservados

Este documento se distribuye bajo la licencia  
“Atribución-CompartirIgual 4.0 Internacional”  
de Creative Commons Disponible en  
<https://creativecommons.org/licenses/by-sa/4.0/deed.es>

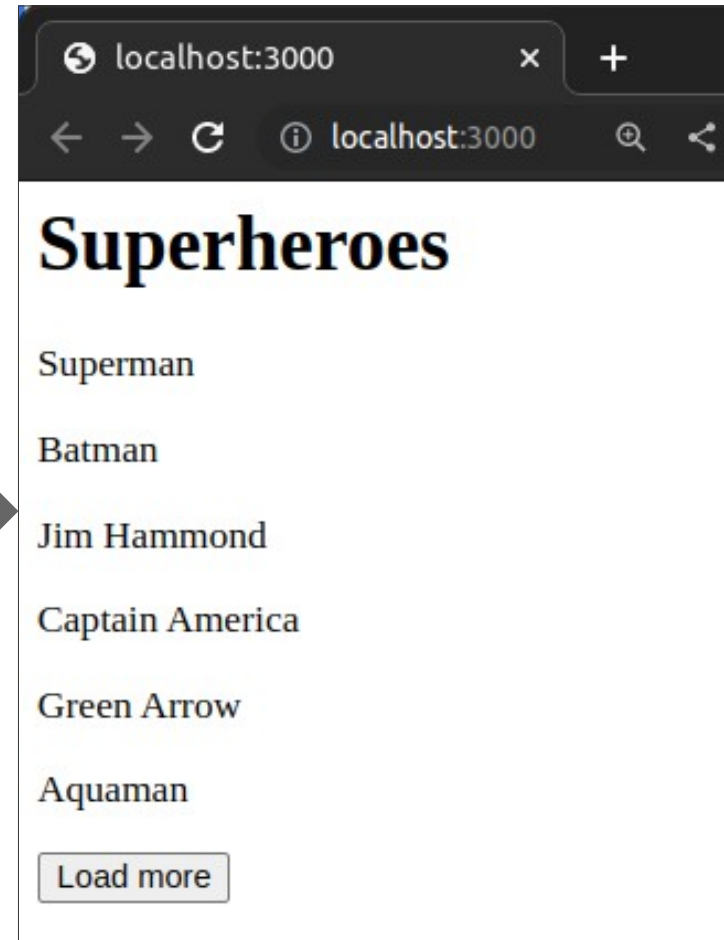
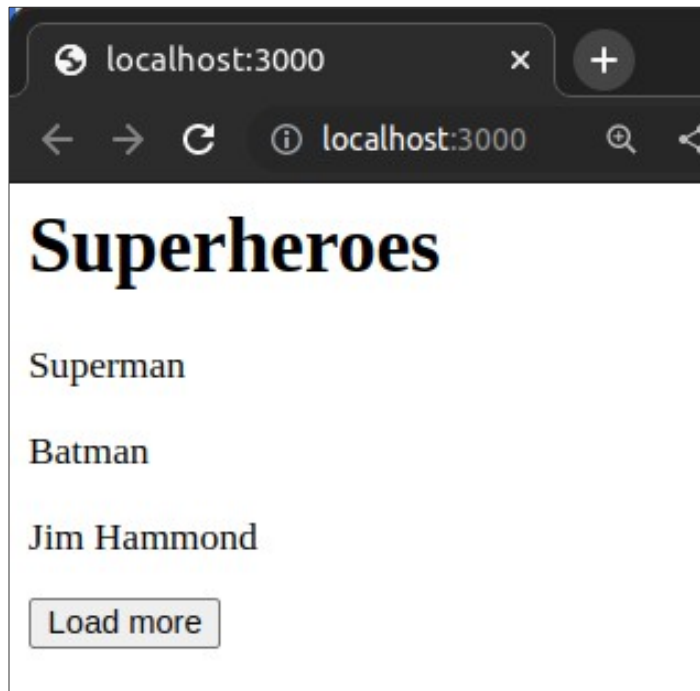


# Ejercicio 1

- Se tiene una web que muestra los superhéroes guardados en un mapa
- Se quiere implementar un botón de “Cargar más” con AJAX
- Los superhéroes se mostrarán de 3 en 3

# Ejercicio 1

## Ejercicio1



# Ejercicio 1

## Ejercicio1

- Al cargar la página se mostrarán los superhéroes 0,1 y 2.
- Cuando se pulse el botón la primera vez se cargarán los superhéroes 3, 4 y 5.
- Cuando se pulse el botón la segunda vez se cargarán los superhéroes 6, 7 y 8.
- La URL que se usa para cargar más resultados deberá indicar qué resultados cargar

# Ejercicio 1

- Se proporciona una web en la que se muestran inicialmente todos los libros

```
import express from 'express';
import { getSuperheroes } from './superheroes.js';
const router = express.Router();
router.get('/', (req, res) => {
  const superheroes = getSuperheroes();
  res.render('index', {
    superheroes: superheroes
  });
});
export default router;
```

# Ejercicio 1

- Se proporciona una web en la que se muestran inicialmente todos los libros

```
const superheroes = new Map();
let id = 0;

export function addSuperhero(superhero) {
  superheroes.set(id, superhero);
  superhero.id = id;
  id++;
}

export function getSuperhero(id) {
  return superheroes.get(id);
}
```

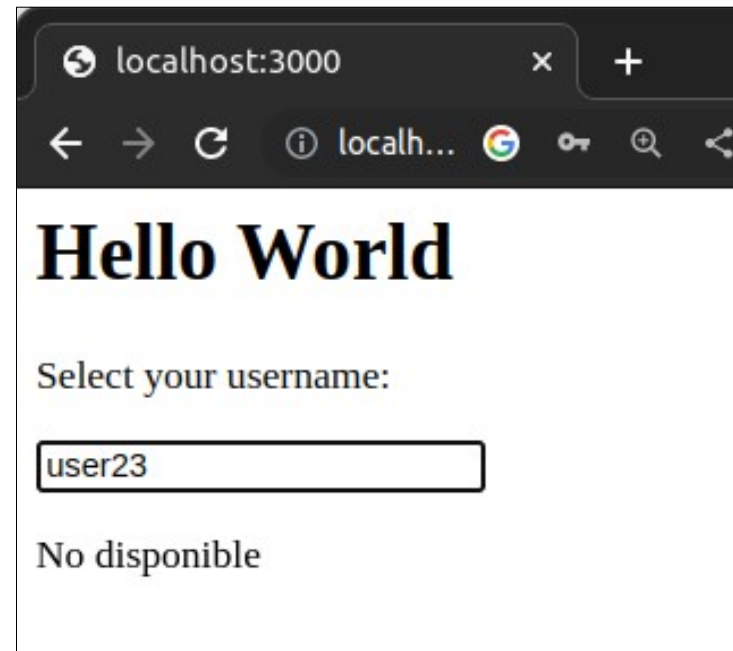
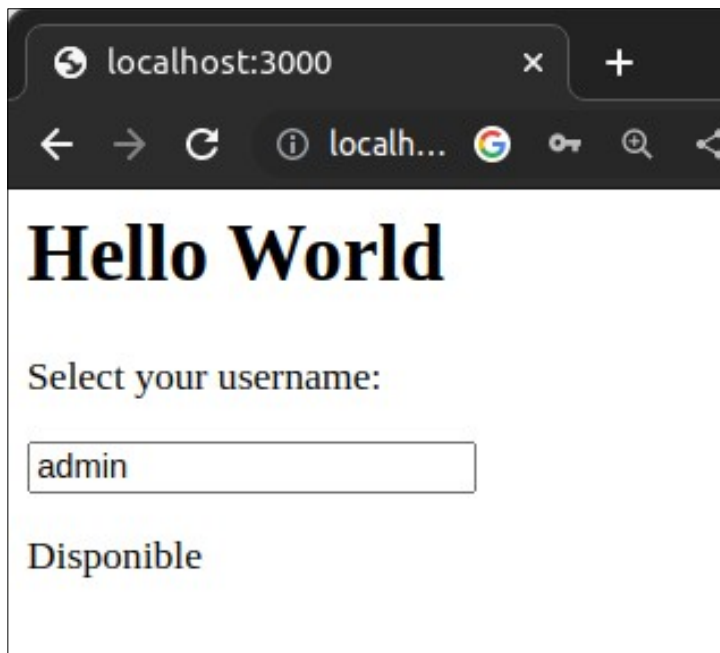
```
export function getSuperheroes(from, to) {
  let values = [...superheroes.values()];
  if (from !== undefined) {
    return values.slice(from, to);
  } else {
    return values;
  }
}

export function loadSampleData() {
  addSuperhero({ name: 'Superman' });
  addSuperhero({ name: 'Batman' });
  ...
}

loadSampleData();
```

# Ejercicio 2

- Web con input de texto que consulta disponibilidad al servidor según va escribiendo el usuario



# Ejercicio 2

- El evento “input” permite ejecutar una función cada vez que el texto cambia

```
<input id='username' oninput='checkUsernameAvailability()' type='text'>
```

- Nota: En las web reales sólo se hace la petición cuando el usuario deja de escribir durante un tiempo para no saturar al servidor