

# HTML y CSS para trabajar en equipo

*Andros Fenollosa*



**Word-  
Camp**

**Zaragoza**

**2023**

Antes de trabajar en equipo...

# Metas



```
<meta charset="utf-8">
<title>Inicio</title>
<meta
  name="viewport"
  content="width=device-width, initial-
scale=1.0, maximum-scale=1.0, user-scalable=no,
shrink-to-fit=no"
>
```

# Metas



```
<meta charset="utf-8">
<title>Inicio</title>
<meta
  name="viewport"
  content="width=device-width, initial-
scale=1.0, maximum-scale=1.0, user-scalable=no,
shrink-to-fit=no"
>
```

# Metas

```
● ● ●  
  
<link  
  rel="icon"  
  type="image/png"  
  href="favicon.png"  
>  
<link  
  rel="apple-touch-icon"  
  href="touch-icon-iphone.png"  
>  
<link  
  rel="apple-touch-icon"  
  sizes="152x152"  
  href="/static/img/apple-touch-icon/touch-icon-ipad.png"  
>  
<link  
  rel="apple-touch-icon"  
  sizes="180x180"  
  href="/static/img/apple-touch-icon/touch-icon-iphone-retina.png"  
>  
<link  
  rel="apple-touch-icon"  
  sizes="167x167"  
  href="/static/img/apple-touch-icon/touch-icon-ipad-retina.png"  
>
```

# Metas

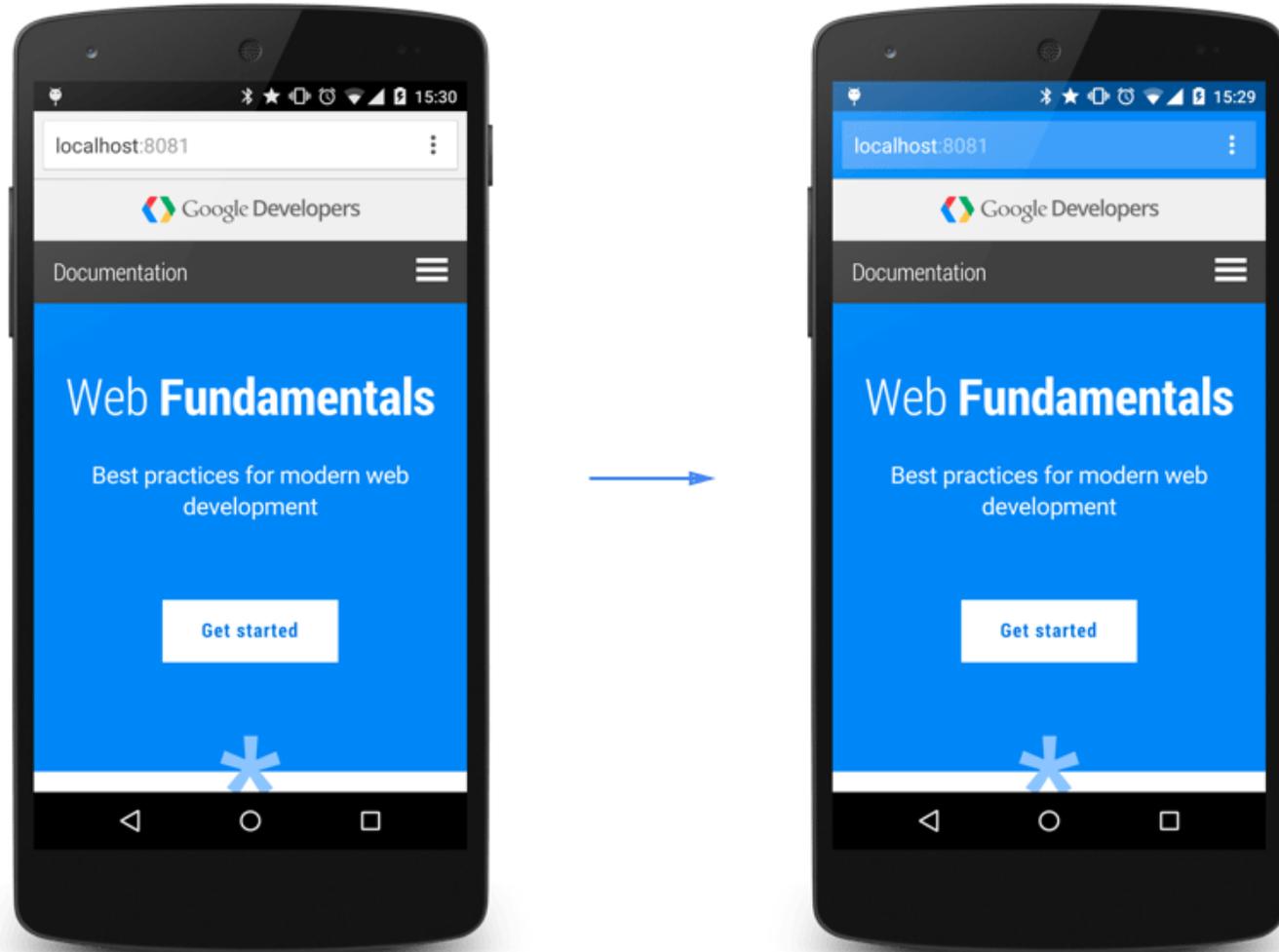


```
<meta
  name="theme-color"
  content="#3c790a"
>

<meta
  name="theme-color"
  content="#ecd96f"
  media="(prefers-color-scheme: light)"
>

<meta
  name="theme-color"
  content="#0b3e05"
  media="(prefers-color-scheme: dark)"
>
```

# Metas



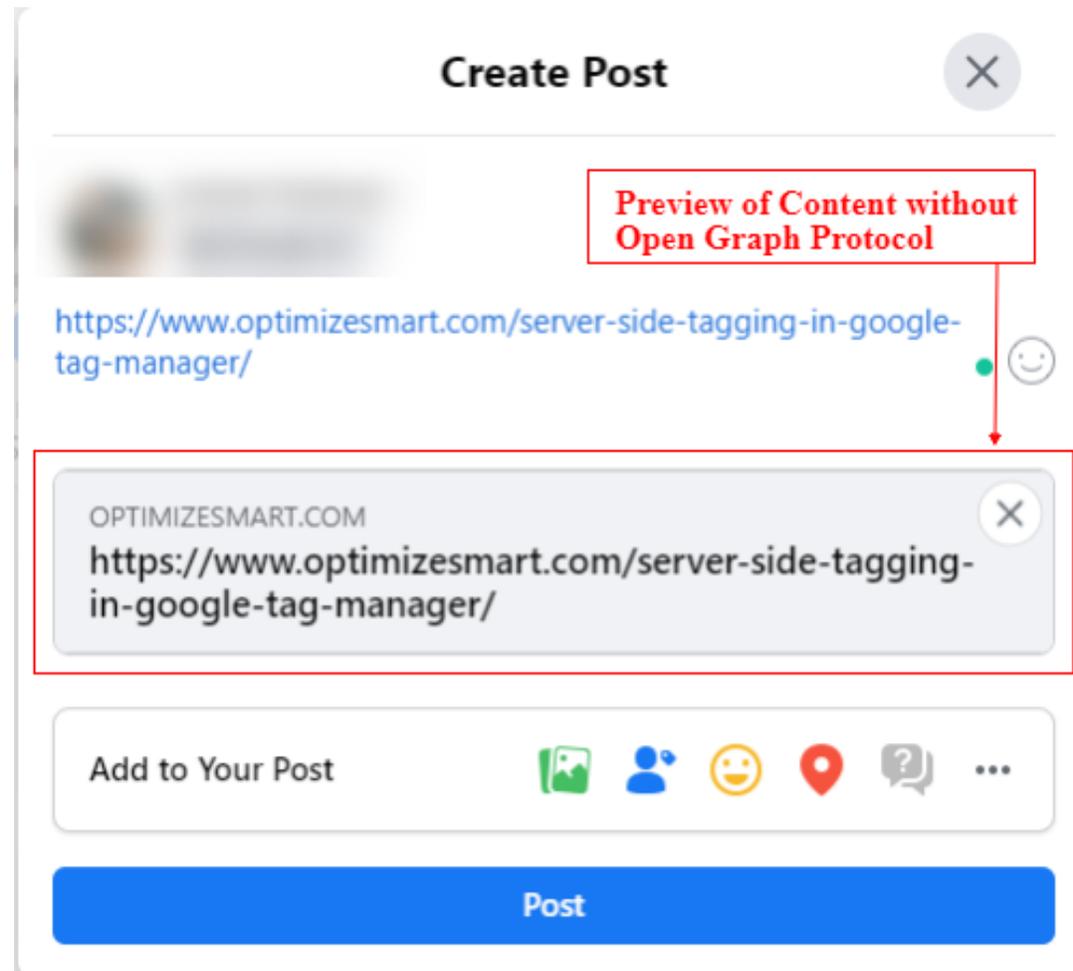
#WCZGZ23

# Metas



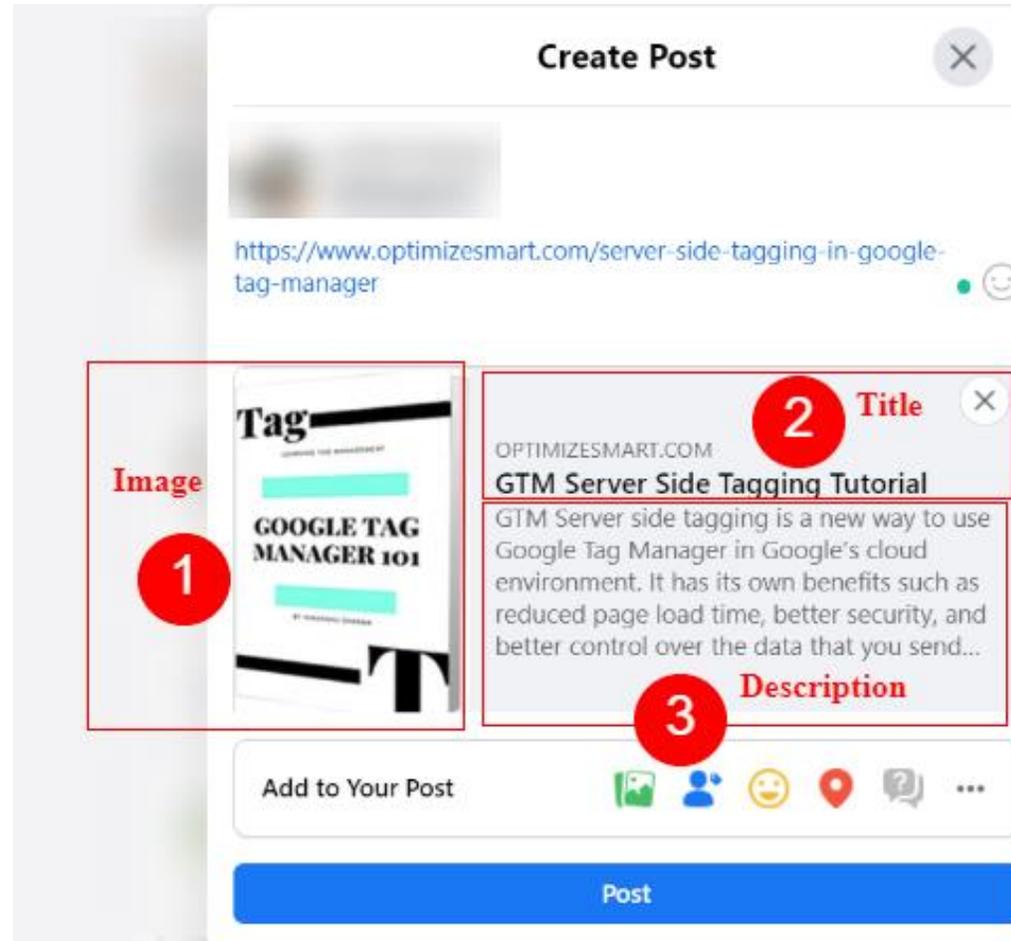
```
<meta
  property="og:image"
  content="img/miniatura.png"
>
<meta
  property="og:title"
  content="The Rock"
>
<!-- website, video, audio, article, book o profile -->
<meta
  property="og:type"
  content="website"
>
<meta
  property="og:url"
  content="http://www.imdb.com/title/tt0117500/"
>
```

# Metas



Optimize Smart

# Metas



Optimize Smart

# Metas



```
<meta
  name="twitter:card"
  content="summary"
>
<meta
  name="twitter:site"
  content="@cuenta"
>
<meta
  name="twitter:creator"
  content="@cuenta"
>
```

# ¡Dame una plantilla!



(Estadísticamente la gente que deja una  suele ser un buen profesional)

<img ¿/?>



```

```

```

```

`<img ¿/?>`

W3C: Los elementos vacíos sólo tienen una etiqueta de inicio; las etiquetas de fin no deben especificarse para los elementos vacíos.

Elementos vacíos: area, base, br, col, command, embed, hr, img, input, keygen, link, meta, param, source, track, wbr

# HTML para trabajar en equipo

# Como se escribe HTML



```
<input id="email" name="email" type="email"  
placeholder="ejemplo@correo.com" required class="input input--small">
```

# Como se escribe HTML



```
<input id="email"  
  name="email"  
  type="email"  
  placeholder="ejemplo@correo.com"  
  required  
  class="input input--small">
```

# Como se escribe HTML



```
<input  
  id="email"  
  name="email"  
  type="email"  
  placeholder="ejemplo@correo.com"  
  required  
  class="input input--small"  
>
```

# Como se escribe HTML



```
<input
  id="email"
  name="email"
  type="email"
  placeholder="ejemplo@correo.com"
  required
  class="
    input
    input--small
  "
>
```

# ASI se escribe HTML



```
<!--  
    Campo de e-mail  
  
    Estados:  
    "input--success" Diseño confirmando formato.  
    "input--error" Diseño de formato equivocado.  
-->  
<input  
    id="email"  
    name="email"  
    type="email"  
    placeholder="ejemplo@correo.com"  
    required  
    class="  
        input  
        input--small  
    "  
>
```

#WCZGZ23

# ASI se escribe HTML

```
<input id="email" name="email" type="email"  
placeholder="ejemplo@correo.com" required  
class="input input--small">
```

```
<!--  
    Campo de e-mail  
  
    Estados:  
    "input--success" Diseño confirmando formato.  
    "input--error" Diseño de formato equivocado.  
-->  
<input  
    id="email"  
    name="email"  
    type="email"  
    placeholder="ejemplo@correo.com"  
    required  
    class="  
        input  
        input--small  
    "  
>
```

# Esto es una tabla



```
<table>
  <tr>
    <th>Nombre</th>
    <th>Apellido</th>
  </tr>
  <tr>
    <td>Francisco</td>
    <td>Ibañez</td>
  </tr>
  <tr>
    <td>Paco</td>
    <td>Roca</td>
  </tr>
</table>
```

```
▼ <table>
  ▼ <tbody>
    ▼ <tr>
      <th>Nombre</th>
      <th>Apellido</th>
    </tr>
    ▼ <tr>
      <td>Francisco</td>
      <td>Ibañez</td>
    </tr>
    ▼ <tr>
      <td>Paco</td>
      <td>Roca</td>
    </tr>
  </tbody>
</table>
```

# Esto es una buena tabla

```
<table>
  <thead>
    <tr>
      <th>Nombre</th>
      <th>Apellido</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Francisco</td>
      <td>Ibañez</td>
    </tr>
    <tr>
      <td>Paco</td>
      <td>Roca</td>
    </tr>
  </tbody>
</table>
```

# Etiquetas de contenido

- `<header>`
- `<main>`
- `<section>`
- `<nav>`
- `<time>`
- `<address>`
- `<article>`
- `<aside>`
- `<footer>`
- `<hgroup>`

Dejad de dar trabajo a los Frontends que tienen sus movidas, usad etiquetas interactivas.

# Etiquetas interactivas

<abbr>



```
<p>Estoy aprendiendo <abbr title="Hypertext Markup Language">HTML</abbr> por  
las mañanas.</p>
```

# Etiquetas interactivas

<details>



```
<details>
  <summary>Factura</summary>
  <ul>
    <li>Vino blanco: 12€</li>
    <li>Aceitunas: 3€</li>
  </ul>
  <p>Total: 15€</p>
</details>
```

# Etiquetas interactivas

<dialog>



```
<dialog open>iHas sido registrado con éxito!</dialog>
```

# Etiquetas interactivas

<datalist>

```
● ● ●  
  
<label>  
  ¿Cual es tu personaje favorito de Harry Potter?:  
  <input  
    list="personajes-harry-potter"  
    type="text"  
    name="personaje-favorito"  
  >  
</label>  
  
<datalist id="personajes-harry-potter">  
  <option value="Harry Potter"></option>  
  <option value="Ron Weasley"></option>  
  <option value="Hermione Granger"></option>  
  <option value="Lord Voldemort"></option>  
  <option value="Albus Dumbledore"></option>  
</datalist>
```

# CSS para trabajar en equipo

# Todo classes



```
<!--  
    Button that deletes the account  
  
    States:  
    "button--disable": The user is not logged in  
-->  
<a  
    href="#"  
    id="button-remove"  
    class="button button--red"  
>Eliminar cuenta</a>
```

# Comunicaros con nomenclaturas

- OOCSS
  - Separar la estructura de la superficie
  - Separación del contenedor del contenido
- SMACSS
  - Aumentar el valor semántico de una sección de HTML y contenido
  - Disminuir la expectativa de una estructura HTML específica
- BEM
  - Separar por bloques, elementos y modificadores.

# BEM

- Todas nuestras secciones/componentes se denominan **bloques**.
- El interior de nuestros **bloques** se denominan **elementos**.
- Un **bloque** y un **elemento** se separa por 2 guiones bajos: blog\_\_articulo.
- Un **elemento** y su **modificador** se separan con 2 guiones: boton--rojo.



# BEM



```

<!-- Bloque Cesta -->
<section class="cesta">
  <!-- Bloque Item -->
  <div class="item">
    <!-- Elemento Item Nombre -->
    <div class="item_nombre">
      Libro
    </div>
    <!-- Fin Elemento Item Nombre -->
    <!-- Elemento Item Precio -->
    <div class="item_precio">
      30 €
    </div>
    <!-- Fin Elemento Item Precio -->
  </div>
  <!-- Fin Bloque Item -->
</section>
<!-- Fin Bloque Cesta -->

```

# BEM



```
<button class="boton">Ver ficha</button>  
<button class="boton boton--verde">Comprar</button>
```



# SASS



```
@import "header"
```

```
$tamanyo-bordes: 1.3rem
```

```
$color-rojo: #e54206
```

```
a
```

```
  color: $color-rojo
```

```
  border: $tamanyo-bordes solid $color-rojo
```

```
nav
```

```
  ul
```

```
    margin: 0
```

```
    padding: 0
```

```
    list-style: none
```

```
    li
```

```
      display: inline-block
```

# Los Media queries son sucios



```
a {  
  display: inline-block;  
  color: #e54206;  
  border: 1.3rem solid #e54206;  
}  
  
@media screen and (min-width: 30rem) and (max-width: 50rem) {  
  a {  
    width: 100%;  
  }  
}
```

# Mejoran con SASS



a

```
display: inline-block  
color: #e54206  
border: 1.3rem solid #e54206  
media screen and (min-width: 30rem) and (max-width: 50rem)  
width: 100%
```

# Pero son alucinantes con Split Media

```
● ● ●  
  
<link  
  href="css/mobile.css"  
  rel="stylesheet"  
  media="all and (max-width: 29rem)"  
>  
<link  
  href="css/tablet.css"  
  rel="stylesheet"  
  media="all and (min-width: 30rem) and (max-width: 50rem)"  
>  
<link  
  href="css/desktop.css"  
  rel="stylesheet"  
  media="all and (min-width: 51rem)"  
>
```

# Patrón 7-1



#WCZGZ23

```
sass/
|
|- abstracts/
|   |- _mixins.sass
|   |- _variables.sass
|- base/
|   |- _base.sass
|   |- _fonts.sass
|   |- _helpers.sass
|   |- _typography.sass
|- components/
|   |- _alert.sass
|   |- _button.sass
|   ...
|- layout/
|   |- _footer.sass
|   |- _header.sass
|- pages/
|   |- _home.sass
|   ...
|- themes/
|   |- _dark.sass
|   |- _light.sass
|- vendors/
|   |- _normalize.sass
|   |- _owl-carousel.sass
|   ...
|- main.sass
```

# Patrón 7-1 lite



```
sass/  
|  
|- _base.sass  
|- _header.sass  
|- _footer.sass  
|- _mixins.sass  
|- _typography.sass  
|- _variables.sass  
|- pages/  
|   |- _home.sass  
|   |- _contact.sass  
|   ...  
|- vendors/  
|   |- _normalize.sass  
|   |- _owl-carousel.sass  
|   ...  
|- main.sass
```

# Patrón 7-1 + Split Media



```
<link
  href="css/mobile.css"
  rel="stylesheet"
  media="all and (max-width: 600px)"
>
<link
  href="css/desktop.css"
  rel="stylesheet"
  media="all and (min-width: 600px)"
>
```



```
sass/
|
|- abstracts/
|   |- _mixins.sass
|   |- _variables.sass
|- base/
|   |- _base.sass
|   |- _fonts.sass
|   |- _helpers.sass
|   |- _typography.sass
|- mobile/
|   |- layout/
|       |- _footer.sass
|       |- _header.sass
|   |- components/
|       |- _alert.sass
|       |- _button.sass
|       |- ...
|   |- pages/
|       |- _home.sass
|       |- _contact.sass
|       |- ...
```

```
|- desktop/
|   |- layout/
|       |- _footer.sass
|       |- _header.sass
|   |- components/
|       |- _alert.sass
|       |- _button.sass
|       |- ...
|   |- pages/
|       |- _home.sass
|       |- _contact.sass
|       |- ...
|- themes/
|   |- _dark.sass
|   |- _light.sass
|- vendors/
|   |- _normalize.sass
|   |- _owl-carousel.sass
|   ...
|- mobile.sass
|- desktop.sass
```

# Flex

- 1D
- Línea.

# Grid

- 2D
- Cuadrícula

# WebGL

- 3D
- Modelado
- Partículas

# Cascade Layers son para equipos

```
● ● ●

@layer base {
  a {
    font-weight: 800;
    color: red; /* Se ignora */
  }

  .pink {
    color: blue; /* Se ignora */
  }
}

@layer typography {
  a {
    color: green;
  }
}

@layer utilities {
  .pink {
    color: pink;
  }
}
```

La arquitectura CSS da fiabilidad y ayuda al mantenimiento,  
el resto no debería ser !important

# FAQ: ¿Debería usar un Framework CSS?

- Limita opciones de implementación.
- Puede ser peligroso actualizar.
- Cargarás en tus visitantes código que no usarás.

FAQ: ¿El subgrid cambiará las reglas del juego?

FAQ: ¿El subgrid cambiará las reglas del juego?

No

# FAQ: ¿Puedo usar CSS sin estandarizar?

Si, usa @supports

```
● ● ●  
  
a {  
    /* Compatible con todos los navegadores */  
    color: rgb(50.24% 14.25% 16.34%);  
}  
  
/* Compatible con Safari */  
@supports (color: lch(30% 45 27)) {  
    a {  
        color: lch(30% 45 27);  
    }  
}
```

# FAQ: ¿En que navegadores debo testear?

## 1. Webkit:

- Midori
- Safari: MacOS y iOS
- GNOME Web: Linux

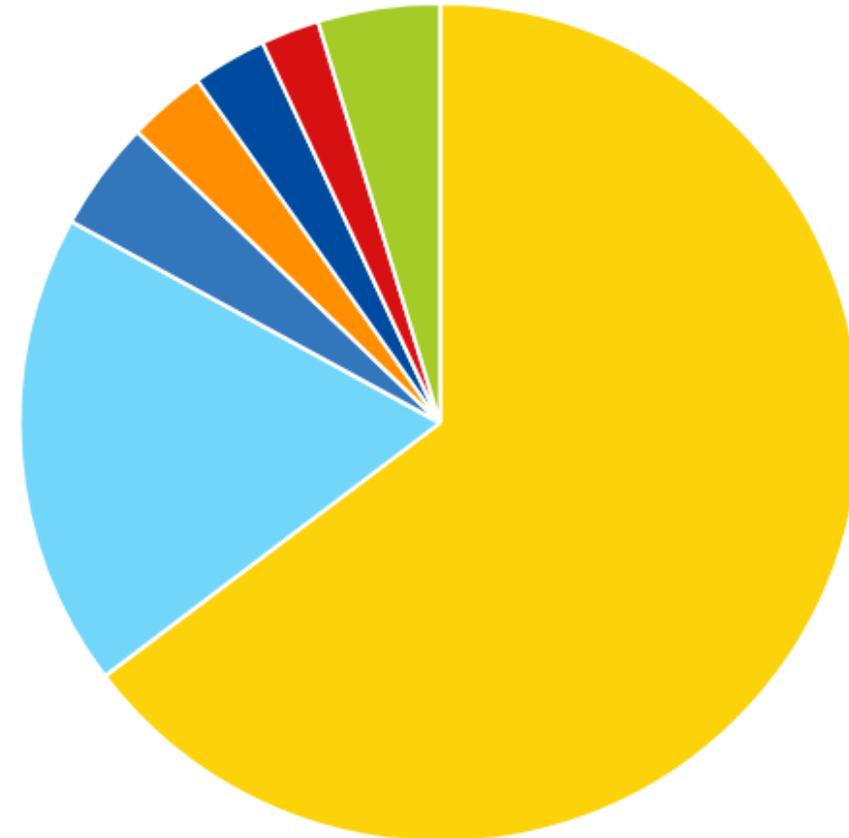
## 2. Blink

- Chrome
- Edge
- Chromium

## 3. Quantum

- Firefox

Chrome Safari Edge Firefox Samsung Internet Opera Otros



# FAQ: ¿React, Vue, Angular...?

# FAQ: ¿React, Vue, Angular...?

Componentes: Lit

<https://lit.dev/>



```
import {LitElement, css, html} from 'lit';
import {customElement, property} from 'lit/decorators.js';

@customElement('saludo-wordcamp')
export class SimpleGreeting extends LitElement {
  // Define scoped styles right with your component, in plain
  CSS
  static styles = css`
    :host {
      color: blue;
    }
  `;

  @property()
  name?: string = 'Mundo';

  render() {
    return html`<p>Hola ${this.name}!</p>`;
  }
}
```

#WCZGZ23

# FAQ: ¿React, Vue, Angular...?

Componentes: Lit

<https://lit.dev/>



```
<saludo-wordcamp></saludo-wordcamp>
```



```
<p>Hola Mundo</p>
```

# FAQ: ¿React, Vue, Angular...?

Eventos: Stimulus

[stimulus.hotwired.dev](https://stimulus.hotwired.dev)



```
<div data-controller="hello">  
  
  <button data-action="click->hello#greet">  
    Aprieta sin miedo  
  </button>  
  
  <span data-hello-target="output">  
  </span>  
  
</div>
```

# FAQ: ¿Testing?

Modo fácil: curl y RegEx  
Modo team: Cypress

The image shows a Cypress test runner interface on the left and a web application on the right. The Cypress runner displays a test suite with 13 steps, including database seeding, login, and transaction creation. The application on the right is a 'Real World App' showing a contact list with a search input field highlighted by a Cypress command: `cy.get('[data-test=user-list-search-input]')`. The application interface includes a navigation bar, a progress indicator, and a list of contacts with their names, user IDs, email addresses, and phone numbers.

Step	Command	Details
1	TASK	db:seed
2	DATABASE	filtering within users data
3	LOGIN BY XSTATE	Authenticating   Austen48
4	WAIT	@loginUser
	(XHR)	POST 200 /login (loginUser)
	(NEW URL)	http://localhost:3000/
	(XHR)	GET 200 /bankAccounts
	(XHR)	GET 200 /transactions/public (publicTransactions)
	(XHR)	GET 200 /notifications (notifications)
1	GET	[data-test*=new-transaction]
2	-CLICK	
	(NEW URL)	http://localhost:3000/transaction/new
	(XHR)	GET 200 /users (allUsers)
3	WAIT	@allUsers
4	GET	[data-test*=user-list-item] (4)
5	-CONTAINS	Delta
6	-CLICK	{force: true}
7	GET	[data-test*=amount-input]
8	-TYPE	95
9	GET	[data-test*=description-input]
10	-TYPE	Fancy Hotel 🏨
11	GET	[data-test*=submit-request]
12	-CLICK	
	(XHR)	POST 200 /transactions (createTransaction)
13	WAIT	@createTransaction

# Gracias

*Soy Andros Fenollosa*

<https://soy.andros.dev/>

