



Certified



Corporation®

MÓDULO 2: Fundamentos de Desarrollo Front-End.

Parte 3: CSS y Responsividad.



Aprendizajes Esperados

Construir una página web utilizando estilos CSS para la definición de aspectos visuales de la interfaz web.

Contenidos

- I. Manejando Hojas de Estilo.
 1. Qué es CSS, fundamentos y utilidad.
 2. CSS y HTML.
 3. Estilos en línea, embebidos, archivos externos.
 4. Referencias y selectores, por clase, por id.
 5. El modelo de cajas.
 6. Estilos más utilizados (fuentes, líneas, cajas, etc...).
 7. Buenas prácticas al construir una hoja de estilos.
 8. Manejo de assets e imágenes. Conociendo rutas absolutas y relativas.
 9. Orden jerárquico de aplicación de reglas CSS y el peso asociado a las reglas.
 10. Inspeccionando estilos con las herramientas para desarrolladores en el navegador.

Contenidos

II. Responsividad.

1. El concepto de Responsividad.
2. Tipos de dispositivos y orientaciones.
3. El concepto Mobile First.
4. Utilización de Media Query.
5. Cómo probar los distintos dispositivos.

I. Manejando Hojas de Estilo

1. Qué es CSS, fundamentos y utilidad.

Las Hojas de estilo CSS, es una tecnología que permite aplicar colores, formas, márgenes, entre otros efectos para crear paginas visualmente más atractivas.

Los estilos definidos pueden ser aplicado a una o varias páginas de forma masiva.

CSS son las siglas de Cascading Style Sheets o Hoja de estilo en cascada en español.

La ultima versión disponible es la CSS3.

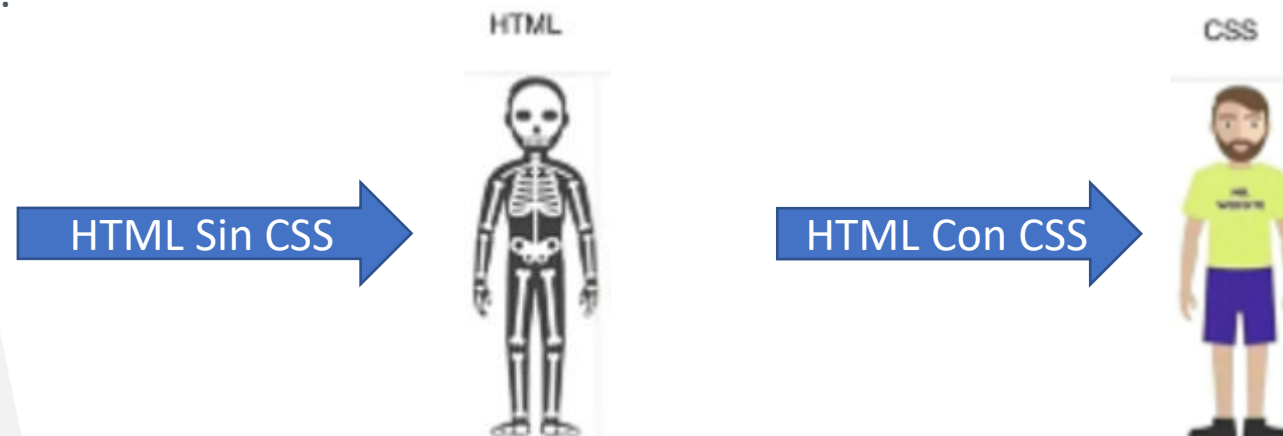


I. Manejando Hojas de Estilo

2. CSS y HTML.

Al comprender el uso de CSS, podemos dar una estética más amigable a nuestro esqueleto HTML.

Principalmente, CSS se utiliza para aplicar distintas propiedades a los elementos HTML para que la pagina web sea atractiva visualmente para los usuarios que la visitan.



I. Manejando Hojas de Estilo

3. Estilos en línea, embebidos, archivos externos.

Existen 3 formas de aplicar estilos con CSS.

En línea.

Es cuando se especifica el estilo **directamente** en el elemento.

Por ejemplo:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Jerarquía</title>
  </head>
  <body>
    <h1 style="color: #0000ff;">Ejemplo de CSS </h1>
  </body>
</html>
```

Ejemplo de CSS

I. Manejando Hojas de Estilo

3. Estilos en línea, embebidos, archivos externos.

Incrustado o interna.

Es cuando se especifica el estilo **dentro** del documento HTML.

Por ejemplo:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Jerarquía</title>
    <style>
      .titulo { color : #ff0000; }
    </style>
  </head>
  <body>
    <h1 class="titulo">Ejemplo de CSS </h1>
  </body>
</html>
```

Ejemplo de CSS

I. Manejando Hojas de Estilo

3. Estilos en línea, embebidos, archivos externos.

Vinculado o Externa.

Es cuando se especifica el estilo por medio de la **invocación** de un archivo CSS.

Por ejemplo:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Jerarquía</title>
    <link rel="stylesheet" href="jerarquia.css">
  </head>
  <body>
    <h1 class="titulo">Ejemplo de CSS </h1>
  </body>
</html>
```

Archivo css con nombre: jerarquía.css

```
> # jerarquia.css > .titulo
.titulo { color : #00ff2afb; }
```

Resultado:

Ejemplo de CSS

I. Manejando Hojas de Estilo

3. Estilos en línea, embebidos, archivos externos.

Ejemplo con distinta forma de agregar estilo:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Jerarquía</title>
    <link rel="stylesheet" href="jerarquia.css">
    <style>
      .titulo { color : yellow; }
    </style>
  </head>
  <body>
    <h1 class="titulo" style="color:blue;">Ejemplo de CSS </h1>
  </body>
</html>
```

```
> # jerarquia.css > .titulo
.titulo { color : #00ff2afb; }
```

Ejemplo de CSS

¿Cuál estilo se aplico en el elemento h1?

I. Manejando Hojas de Estilo

3. Estilos en línea, embebidos, archivos externos.

Ejemplo de propiedad repetida con distinto valor:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Jerarquía</title>
  </head>
  <body>
    <h1 class="titulo" style="color:blue; color : grey;">Ejemplo de CSS </h1>
  </body>
</html>
```

¿Cuál estilo se aplica en el elemento h1?

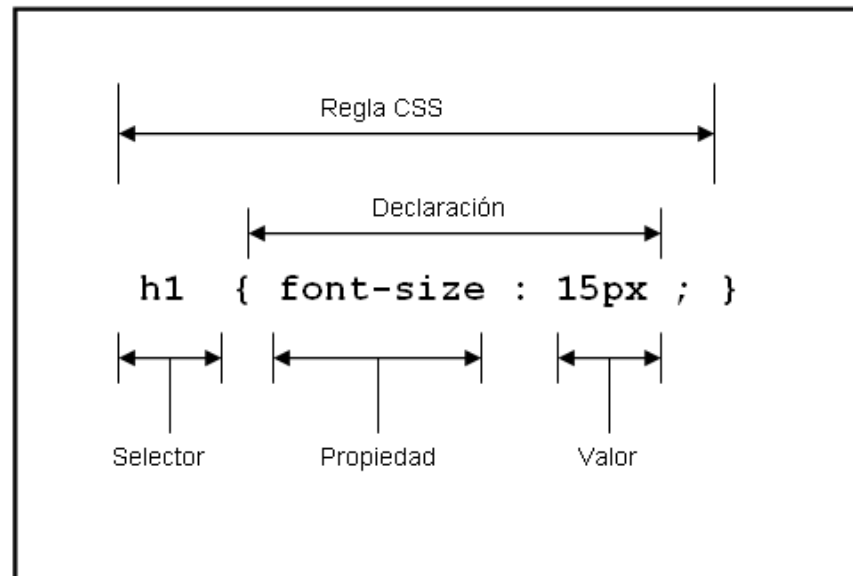


I. Manejando Hojas de Estilo

4. Referencias y selectores, por id, por clase.

La sintaxis básica de CSS3 es la siguiente:

```
selector { propiedad : valor; propiedad : valor; }
```



I. Manejando Hojas de Estilo

4. Referencias y selectores, por id, por clase.

Donde:

- **Selector:** es donde se especifica el elemento donde se quiere aplicar algún estilo.

Para identificar al elemento, se puede utilizar:

- **Por Tipo o Nombre del tag del elemento:**

Esta acción afectará a todos los elementos de la pagina.

Ejemplo: para el tag <h1>, se nombre como **h1**.

I. Manejando Hojas de Estilo

4. Referencias y selectores, por id, por clase.

Para aplicar o referenciar propiedades a un elemento HTML, se utilizan las palabras reservadas:

- **Id:**

El nombre a utilizar debe ser único dentro de la pagina, por lo que no se debe asignar a otro elemento dentro de la pagina.

El símbolo para referenciar el id en css es # (gato).

```
</head>
<body>
  <div id="divImagen">

  </div>
</body>
</html>
```

```
#divImagen { width: 50%; }
```

I. Manejando Hojas de Estilo

4. Referencias y selectores, por id, por clase.

- **Class:**

Permite tener más de un nombre asociado. Lo que implica que se pueden agregar más de un estilo o propiedades.

El nombre de una clase se puede repetir en varios elementos, donde se aplica las mismas propiedades.

El símbolo para referenciar una clase en css es el . (punto).

```
</head>
<body>
  <div class="titulo">

  </div>
</body>
</html>
```

```
</head>
<body>
  <div class="titulo colores formas">

  </div>
</body>
</html>
```

```
> # jerarquia.css > .titulo
.titulo { color : #00ff2afb; }
```

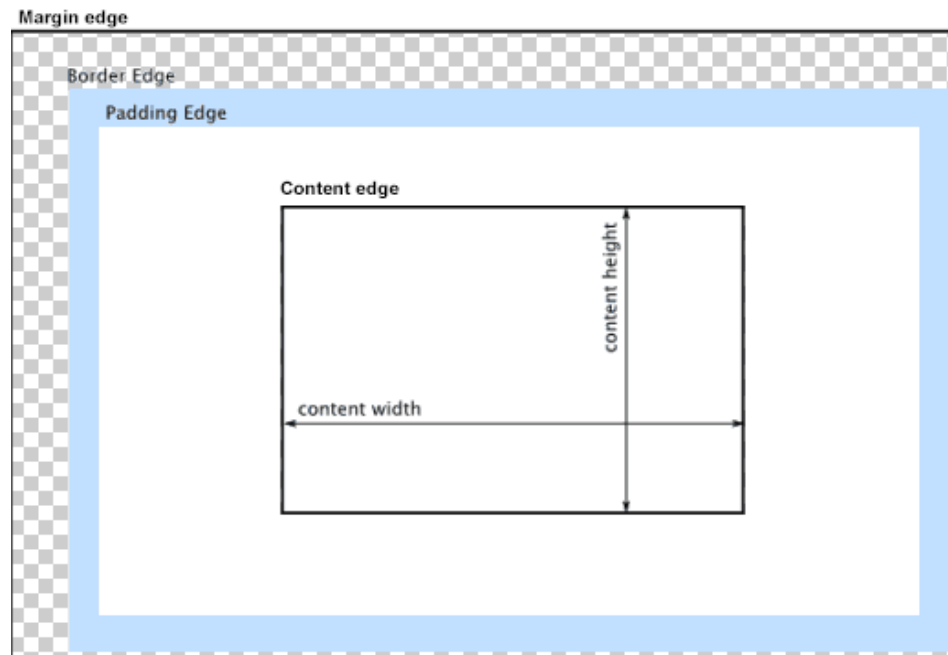
I. Manejando Hojas de Estilo

5. El modelo de cajas.

El modelo de cajas es la representación, con forma de rectángulo, que se da, a cada elemento dibujado por el navegador.

Esta forma tiene asociado las siguientes propiedades **content**, **padding**, **border** y **margin**.

Diagrama



I. Manejando Hojas de Estilo

6. Estilos más utilizados.

Estilos o propiedades más utilizadas son las siguientes:

background	: Define una imagen o color, del fondo de un elemento.
color	: Define el color de la fuente(letra).
font-size	: Define el tamaño de la fuente (letra).
font-family	: Define el tipo de letra a utilizar.
text-align	: Define la alineación del texto.
border	: Define el estilo del borde del elemento.
Height	: Define el alto que tendrá el objeto.
Width	: Define el ancho que tendrá el objeto.

Otras propiedades visitar el siguiente link:

https://developer.mozilla.org/es/docs/Web/CSS/Referencia_CSS

I. Manejando Hojas de Estilo

7. Buenas prácticas al construir una hoja de estilos.

Buenas practicas, hace referencia a la seguir una serie de reglas establecidas por la industria. Estas reglas son opcionales de seguir, pero es bueno tenerlas presente al momento de construir un archivo css.

Algunas buenas practicas son:

- Utilizar tabulación.
- Nombres consistentes con el elemento o zona a afectar
- Organizar los estilos que se aplicaran de arriba hacia abajo
- Combinar elementos que tendrán el mismo estilo.
- En lo posible, utilizar varias clases en los elementos.
- Al finalizar la pagina, comprimir el documento para reducir su tamaño.

I. Manejando Hojas de Estilo

8. Manejo de assets e imágenes. Conociendo rutas absolutas y relativas.

Asserts:

Son carpetas que contienen archivos o recursos adicionales al documento HTML que son utilizados en la pagina para complementar la interfaz grafica de la pagina web.

Estos recursos son agrupados, por lo general, según el tipo u objetivo del recurso.

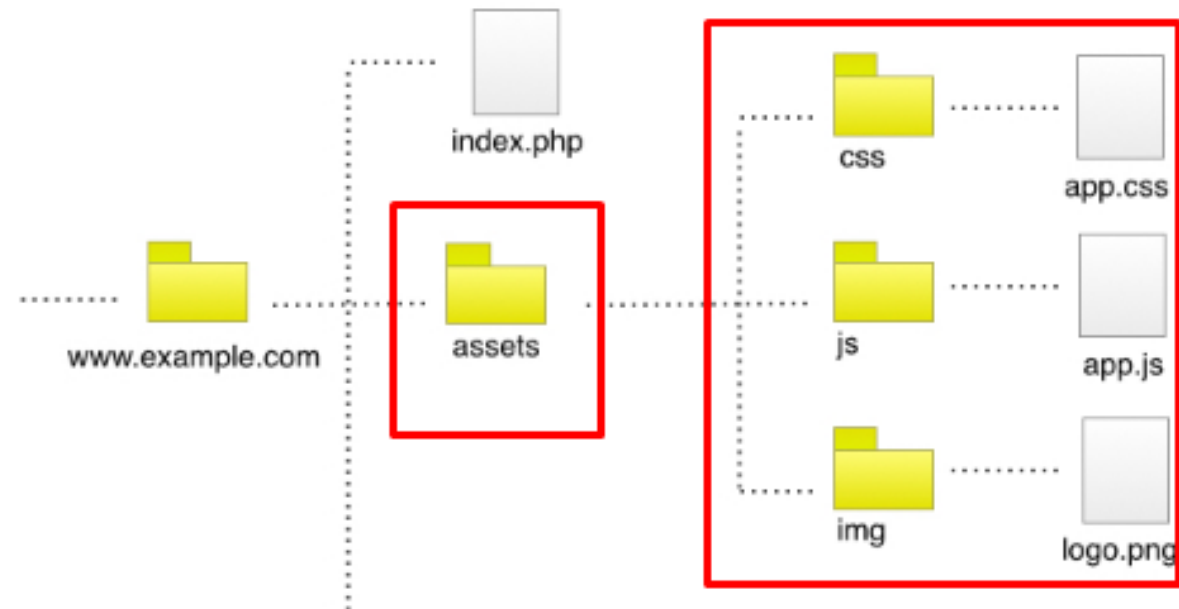
Algunos recursos utilizados son:

- Hoja de estilos en cascada o CSS
- Imágenes en distintos formatos. Por ejemplo: PNG, JPG, GIF, etc.
- JavaScript
- Videos en formato MP4, AVI, etc.

I. Manejando Hojas de Estilo

8. Manejo de assets e imágenes. Conociendo rutas absolutas y relativas.

Ejemplo de las carpetas Asset



I. Manejando Hojas de Estilo

8. Manejo de assets e imágenes. Conociendo rutas absolutas y relativas.

Rutas

Existen 2 tipos de rutas que se utilizan en las paginas web:

Rutas absolutas

Usada principalmente para acceder a recursos fuera de nuestro dominio.

Ejemplo: `https://www.dominio.cl/assert/css/`

Rutas relativas

Usado principalmente para adjuntar recursos disponible localmente a nuestro sitio.

Ejemplo: `/assert/images/`

I. Manejando Hojas de Estilo

9. Orden jerárquico de aplicación de reglas CSS y el peso asociado a las reglas.

En CSS, existe la jerarquía de aplicación, el que consiste en que solo un estilo se aplica, dependiendo de la precedencia que tenga la forma, en la siguiente muestra desde la primera prioridad hasta la de menor prioridad.

1. **En línea.**
2. **Incrustada o interna.**
3. **Vinculado o externa.**

Si por algún motivo se repite una propiedad para un mismo elemento, el que se aplica o perdura, será, la **última** propiedad señalada.

I. Manejando Hojas de Estilo

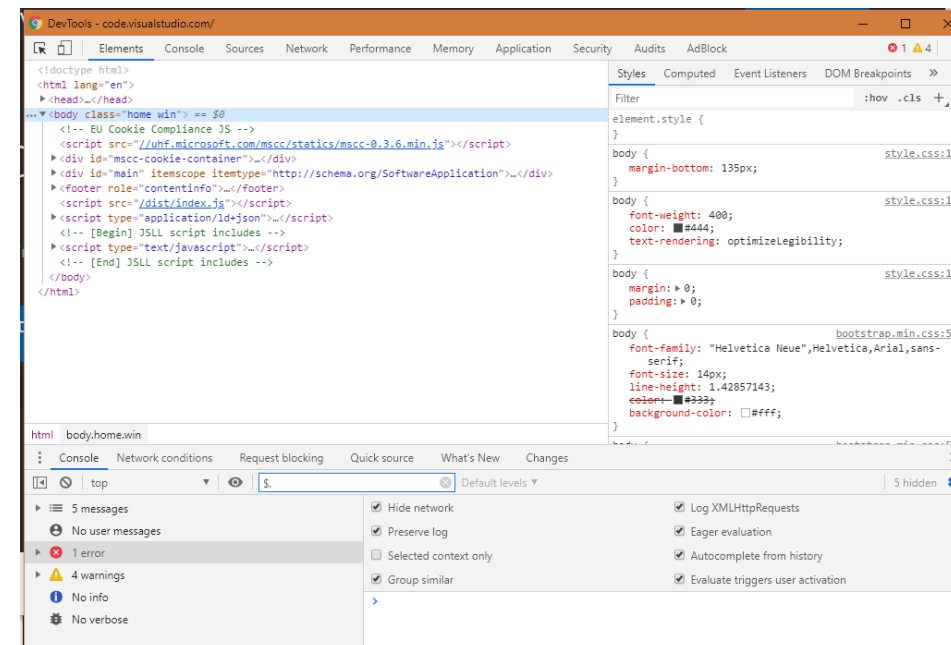
10. Inspeccionando estilos con las herramientas para desarrolladores en el navegador.

Al utilizar Chrome como navegador el inspector, aparece cuando se presiona F12 o con el botón derecho del mouse, la opción inspeccionar.

Esta herramienta tiene el nombre de Dev Tools

En ella se puede ver:

- El código HTML
- La consola (js)
- Los archivos descargados
- Almacenamiento local
- Errores
- Entre otras informaciones acerca de la pagina.



I. Manejando Hojas de Estilo

10. Inspeccionando estilos con las herramientas para desarrolladores en el navegador.

Para ver los estilos aplicados a una página, se debe seleccionar la opción “Styles”.



II. Responsividad.

1. El concepto de Responsividad.

El concepto de responsividad está relacionado al diseño de interfaces gráfica y tiene como objetivo, que la visibilidad de una pagina web, se adapte a cualquier tipo de dispositivo que el usuario use para visitarla.

Por lo tanto, la pagina debería estar disponible, sin errores gráficos, en todos los dispositivos disponibles en el mercado, como por ejemplo: móviles, Tablet, PC, notebook, o cualquier otro con conexión al sitio.

En la actualidad existen varios “Framework” que ayudan al desarrollo de paginas web responsivas, entre las cuales están:

- Bootstrap (Ampliamente usado y recomendado)
- Foundation
- Materialize
- JQuery Mobile
- PureCSS



Pure



II. Responsividad.

2. Tipos de dispositivos y orientaciones.

Los tipos de dispositivos a considerar para el desarrollo de una pagina web son:

- Teléfonos móviles
- Tablet
- PC y/o Notebooks.
- Televisores



Las orientaciones se consideran las siguientes:

- Horizontal o Landscape
- Retrato o Portrait.



II. Responsividad.

3. El concepto Mobile First.

El concepto **Mobile First** hace referencia a los “Framework” que ayudan a crear pagina web responsivas y que por defecto, el diseño inicial es para un **dispositivo móvil**, sin perjuicio de que posteriormente, se adapte a pantallas más grandes si es requerido.

Mobile First Web Design



II. Responsividad.

4. Utilización de Media Query.

Las “**media query**” , permiten aplicar un estilo en función del tipo de dispositivo o según la resolución o ancho de la pantalla en que se ve la página.

Se utiliza la instrucción @medio para definir la resolución y estilo por aplicar.

Ejemplo:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>@media</title>
    <style>
      .titulo { color: grey; }
      @media (max-width: 800px) {
        .titulo { color:blue; }
      }
    </style>
  </head>
  <body>
    <h1 class="titulo">Ejemplo de @media </h1>
  </body>
</html>
```

II. Responsividad.

4. Utilización de Media Query.

Para abarcar todas las pantallas de los dispositivos mencionados, se puede utilizar la siguiente tabla de resoluciones:

Dispositivo	Resolución
Móviles	< 576px con orientación portrait.
Móviles	mínimo 576px con orientación horizontal.
Tablet	mínimo 768px o más.
NoteBooks y Equipos de escritorio	mínimo 992px o más.
Pantallas Grandes (tv, etc)	mínimo 1200px o superior.

Referencia de resolución:

<https://getbootstrap.com/docs/4.3/layout/overview/>

II. Responsividad.

5. Cómo probar los distintos dispositivos.

Al diseñar una pagina responsiva, se debe disponer de la herramienta adecuada para probar los diferentes formatos de dispositivo de forma virtual y así, evitar disponer del equipo físico para hacer las pruebas.

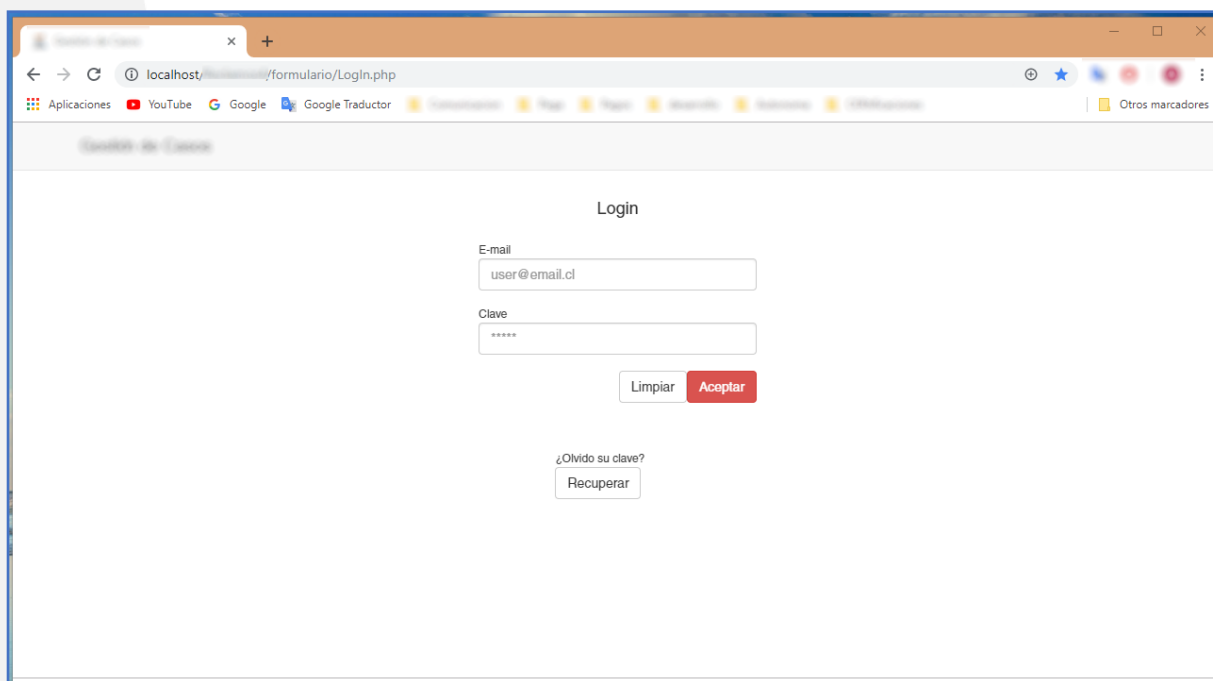
Existen dos formas de probar los formatos con el navegador (chrome). La primera, es cambiando el tamaño del navegador para que aumente o reduzca la resolución y utilizar el DevTools.

DevTools, viene integrado por defecto varios dispositivos móviles, también, esta la opción de agregar otros dispositivos que se requieran para probar la página.

II. Responsividad.

5. Cómo probar los distintos dispositivos.

Reduciendo el tamaño del navegador.



localhost/formulario/Login.php

Aplicaciones YouTube Google Google Traductor

Inicio de Sesión

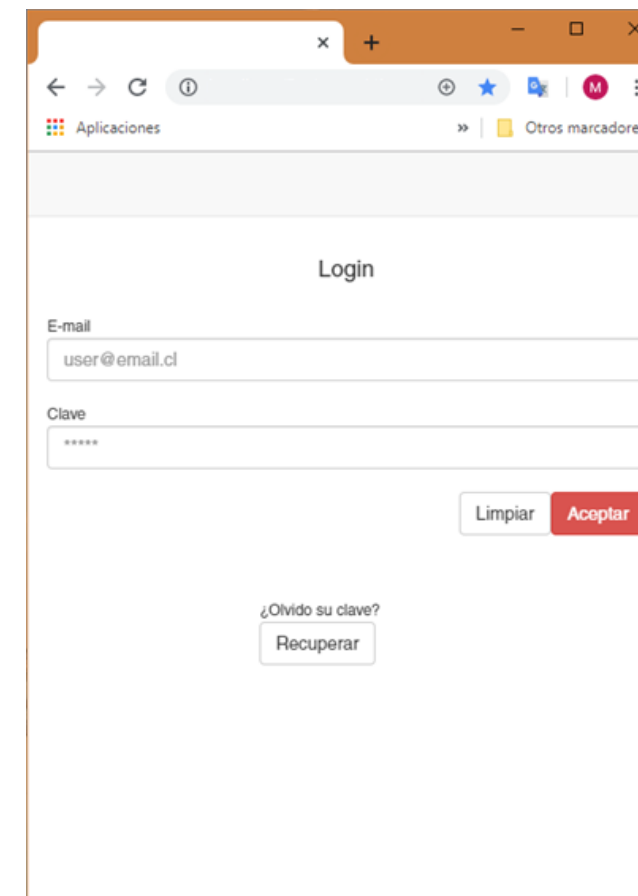
Login

E-mail
user@email.cl

Clave

Limpiar Aceptar

¿Olvido su clave?
Recuperar



Aplicaciones Otros marcadores

Login

E-mail
user@email.cl

Clave

Limpiar Aceptar

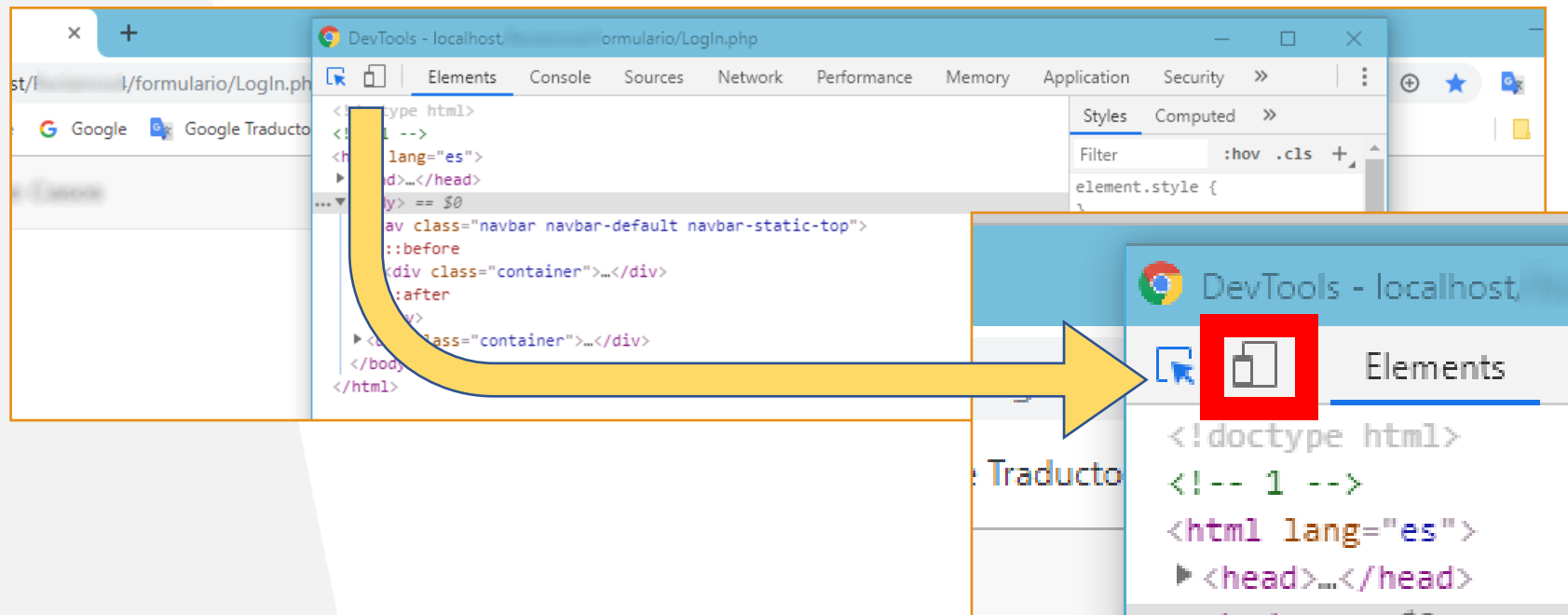
¿Olvido su clave?
Recuperar

II. Responsividad.

5. Cómo probar los distintos dispositivos.

DevTools.

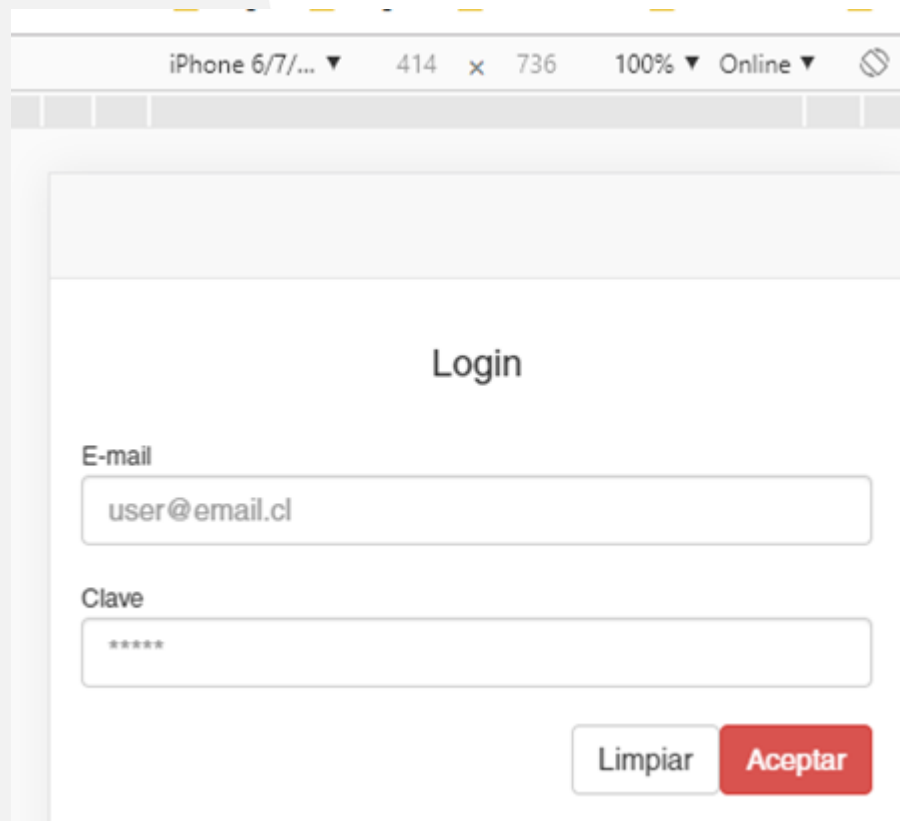
Para acceder a **DevTools**, se debe presionar botón derecho y escoger la opción inspeccionar o también presionando ctr + shift + i.



II. Responsividad.

5. Cómo probar los distintos dispositivos.

Ejemplo del navegador con vista de un iPhone, tanto en formato *portrait* como *landscape*.



iPhone 6/7/... 414 x 736 100% Online

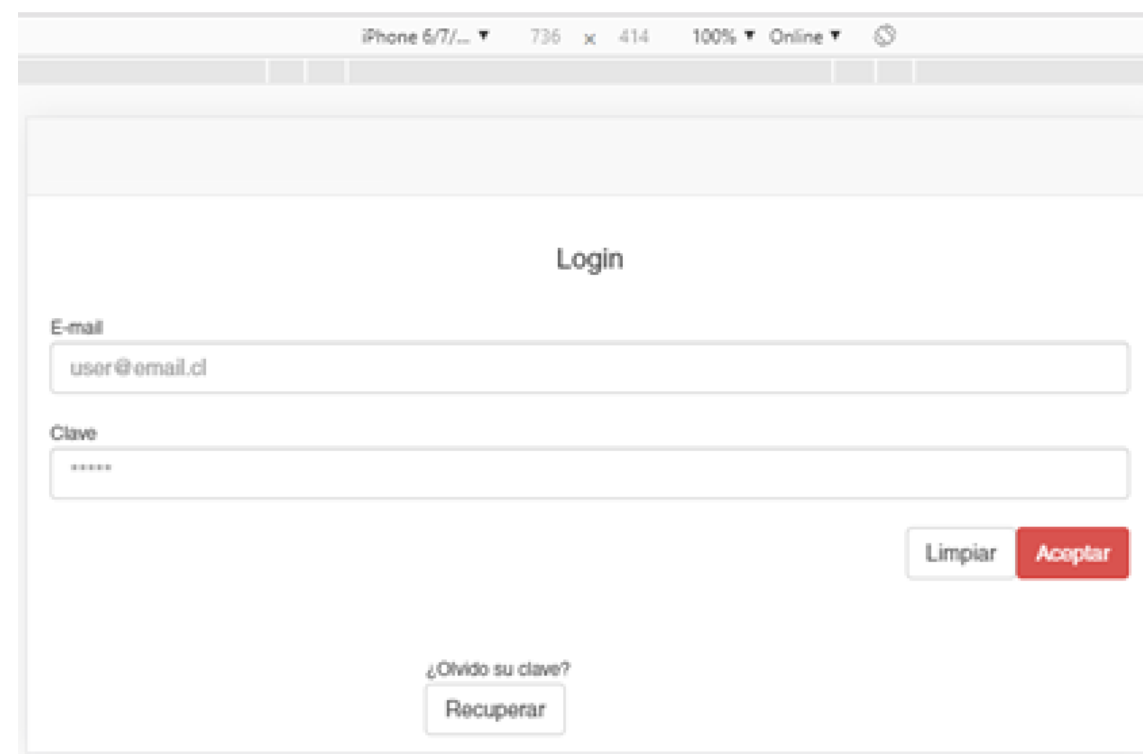
Login

E-mail

user@email.cl

Clave

Limpiar Aceptar



iPhone 6/7/... 736 x 414 100% Online

Login

E-mail

user@email.cl

Clave

Limpiar Aceptar

¿Olvido su clave?

Recuperar

Actividad

