

# Presentadores de diapositivas HTML5

Libro interactivo



# Presentadores de diapositivas HTML5 INTERACTIVO

Juan Guillermo Rivera Berrío  
Institución Universitaria Pascual Bravo



Fondo Editorial Pascual Bravo  
Medellín  
2020

Título de la obra:  
Presentadores de diapositivas HTML5

Autor:  
Juan Guillermo Rivera Berrío

Código JavaScript para el libro: [Joel Espinosa Longi](#), [IMATE](#), UNAM.  
Recursos interactivos: [DescartesJS](#)  
Fuentes: [Lato](#) y [UbuntuMono](#)  
Núcleo del libro interactivo: septiembre 2023

Fondo Editorial Pascual Bravo  
Calle 73 73A-226  
PBX: (574) 4480520  
Apartado 6564  
Medellín, Colombia  
[www.pascualbravo.edu.co](http://www.pascualbravo.edu.co)

ISBN: [978-958-52584-8-8](#)



[Creative Commons Attribution License 4.0](#) license.

# Tabla de contenido

Prefacio .....	7
<b>1. Algunos elementos del lenguaje HTML .....</b>	<b>9</b>
1.1 Introducción .....	11
1.2 Editores HTML .....	12
1.3 HTML .....	13
1.4 Textos habituales en HTML .....	20
1.5 Cambios de estilo - CSS .....	22
1.5.1 Cambios de estilo en las etiquetas .....	23
1.5.2 Cambios de estilo en una etiqueta div .....	25
1.5.3 Cambios de estilo en la cabecera .....	26
<b>2. Diapositivas con desplazamiento en el plano .....</b>	<b>31</b>
2.2 Presentador Deck.js .....	35
2.3 Presentador DZSlides .....	40
2.4 Presentador Google I/O Slides .....	46
2.5 Presentador Fathom .....	52
2.6 Presentadores con narración .....	56
2.6.1 PowerPoint narrador .....	57
2.6.2 Presentador Deck narrador .....	60
2.6.3 Presentador DZSlides narrador .....	62
2.7 Presentadores de diapositivas CSS3 .....	64

2.7.1 Presentador Fluid CSS3 Slideshow .....	65
2.8 Presentador Flowtime.js .....	68
2.9 Presentador Pow .....	72
2.10 Presenteer.js .....	73
2.11 Shower Presentation Engine .....	74
2.12 Slider.css .....	77
2.13 Webslides .....	79
2.14 Wobbly Slideshow Effect .....	81
2.15 Stack.js .....	82
2.16 Clip-Path Revealing Slider .....	83
2.17 Motion blur effect using SVG filters .....	84
2.18 Preview slider .....	85
<b>3. Diapositivas en 3D .....</b>	<b>87</b>
3.1 Introducción .....	89
3.2 Un breve retorno al HTML .....	91
3.3 Slicebox .....	93
3.4 Efecto tipo cubo .....	97
3.4.1 cubeTransition.js .....	97
3.4.2 3D slider con puro CSS3 .....	99
3.4.3 jQuery Fullscreen Presentation .....	100
3.4.4 Adaptor .....	101
3.5 Reveal .....	103

3.5.1 Reveal 2013 .....	104
3.5.2 Reveal 2020 .....	110
3.6 Split 3D Carousel .....	112
3.7 Flipping Gallery .....	113
3.8 StackSlider .....	115
3.9 Presentador de panel triple .....	117
3.10 Bespoke.js .....	118
<b>4. impress.js*</b> .....	<b>123</b>
4.1 Introducción .....	125
4.2 Ejercicio 1. Presentador de diapositivas en el plano .....	128
4.2.1 Hojas de estilo externas e internas .....	129
4.2.2 Contenido de las diapositivas .....	133
4.3 Ejercicio 2. Diapositivas con escala y rotación .....	145
4.4 Ejercicio 3. Diapositivas circulares .....	151
4.5 Ejercicio 4. Diapositivas en varios planos .....	155
4.6 Ejercicio 5. Diapositivas en un cubo .....	158
4.7 Ejercicio 6. Diapositivas sin marcos .....	163
4.8 Editor de diapositivas - Strut .....	165



# Prefacio

En este libro encontrarás algunas propuestas de presentadores de diapositivas en HTML5. El uso de este tipo de presentadores permite evitar la dependencia con presentadores comerciales como el PowerPoint de Microsoft o el KeyNote de Apple que, además, restringen la incorporación de algunos objetos diseñados en JavaScript, como las escenas interactivas de DescartesJS, GeoGebra, entre otros. Otra ventaja de usar HTML5 es poder compartir nuestras presentaciones vía web.

Presentaremos, inicialmente, algunos elementos mínimos del lenguaje HTML necesarios para el diseño de las presentaciones. Posteriormente, abordaremos las presentaciones de desplazamiento en el plano, haciendo énfasis en el uso de escenas interactivas. En atención a aquellos usuarios que les gusta impactar con efectos especiales, explicaremos algunos presentadores de diapositivas en 3D. Finalmente, describiremos las fortalezas que presenta el Impress.js, como alternativa al Prezzi.

Cada una de las herramientas, utilizadas en este libro, son de licencia creative commons o, algunas, de tipo GNU.



# Capítulo I

## Algunos elementos del lenguaje HTML



# 1.1 Introducción

Antes de iniciar, observa el siguiente vídeo, el cual tiene por objeto motivarte a aprender cómo diseñar nuestras presentaciones sin dependencia alguna y, en especial, que puedan ser compartidas a través de la red.:



Tanto los vídeos como las escenas interactivas, se pueden ampliar en una ventana emergente, haciendo clic en la esquina superior derecha del elemento.

En este capítulo describimos algunos elementos mínimos de HTML, necesarios para comprender cómo se modifican las diapositivas de los diferentes presentadores diseñados en JavaScript. No se trata de un curso intensivo de HTML, es sólo una aproximación al diseño básico de páginas web en formato HTML5, con fundamento en el lenguaje JavaScript.

## 1.2 Editores HTML

Todas nuestras presentaciones deben ser diseñadas usando un editor de textos planos<sup>1</sup>. Recomendamos algunos de estos editores tipo WYSIWYG (acrónimo de *What You See Is What You Get*, que en español significa "lo que ves es lo que obtienes"):

- BlueGriffon: <http://www.bluegriffon.org/> (Windows, Mac y Linux).
- Dreamweaver (comercial)

También son útiles editores simples como:

- SublimeText: <http://www.sublimetext.com/> (Mac, Windows y Linux).
- Notepad++: <http://notepad-plus-plus.org/> (Windows)

o, en algunos casos, un editor como el bloc de notas.

---

<sup>1</sup> Los editores de textos "planos" se distinguen de los procesadores de texto en que se usan para escribir solo texto, sin formato y sin imágenes, es decir sin diagramación (véase [https://es.wikipedia.org/wiki/Editor\\_de\\_texto](https://es.wikipedia.org/wiki/Editor_de_texto)).

## 1.3 HTML



Es un lenguaje de marcas para formatear y estructurar un documento, que puede leerse en cualquier navegador. Está estandarizado por la norma ISO de SGML (*Standard Generalized Markup Language*).

Un documento HTML tiene tres etiquetas que describen la estructura general de un documento y dan una información sencilla sobre él:

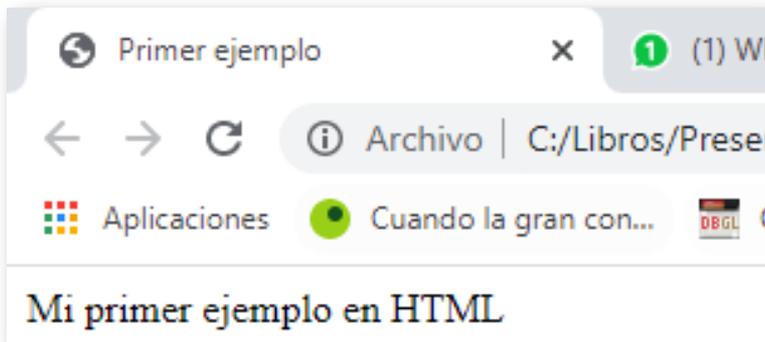
```
<html>, <head> y <body>
```

Las etiquetas pueden escribirse tanto en mayúsculas como en minúsculas, pero se recomienda el uso de minúsculas.

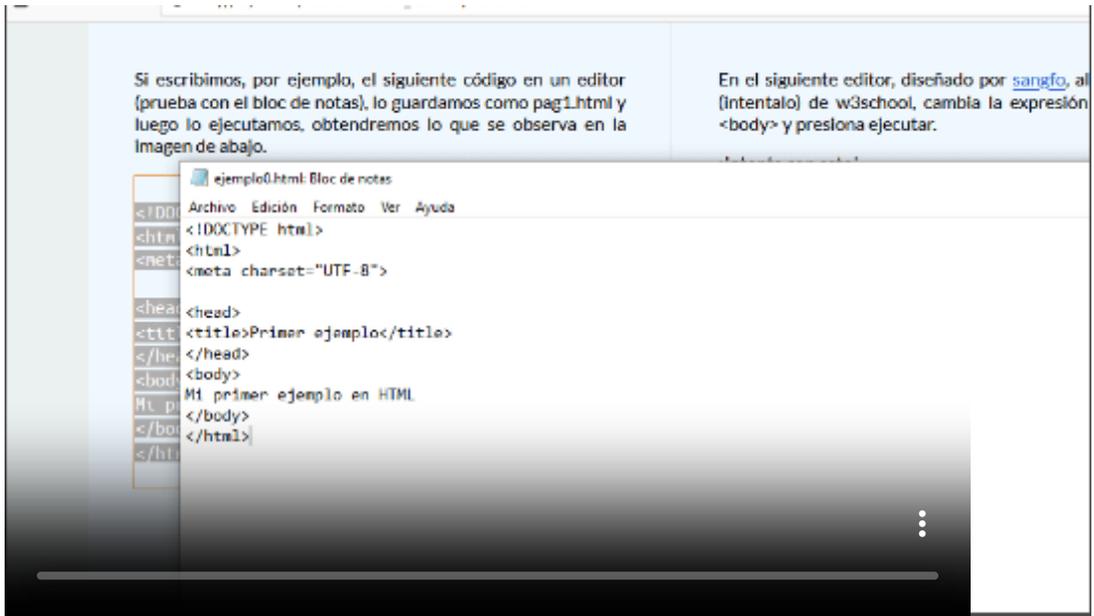
Si escribimos, por ejemplo, el siguiente código en un editor (prueba con el bloc de notas), lo guardamos como pag1.html y luego lo ejecutamos, obtendremos lo que se observa en la imagen de abajo.

```
<!DOCTYPE html>
<html>
<meta charset="UTF-8">

<head>
<title>Primer ejemplo</title>
</head>
<body>
Mi primer ejemplo en HTML
</body>
</html>
```



En el siguiente vídeo, observa cómo lo hicimos con el bloc de notas.



En este capítulo usaremos otros editores que nos permiten verificar, rápidamente, nuestros códigos HTML. Por ejemplo, en los siguientes editores cambia la expresión "Mi primer ejemplo en HTML", que hay en el <body> y presiona ejecutar<sup>2</sup> (puedes ampliar el editor en una ventana emergente, para mayor comodidad).

Obviamente, se trata de probar códigos cortos, pues para los presentadores de diapositivas, debemos usar los editores de textos planos que relacionamos en el apartado 1.2.

---

<sup>2</sup> El primer editor es diseñado por [sangfo](#) y, el segundo, por [slidear](#), al estilo del *Tryit* (inténtalo) de w3school.

¡Intenta con este!



▶ Ejecutar

```
<!DOCTYPE html>
<html>
<meta charset="UTF-8">

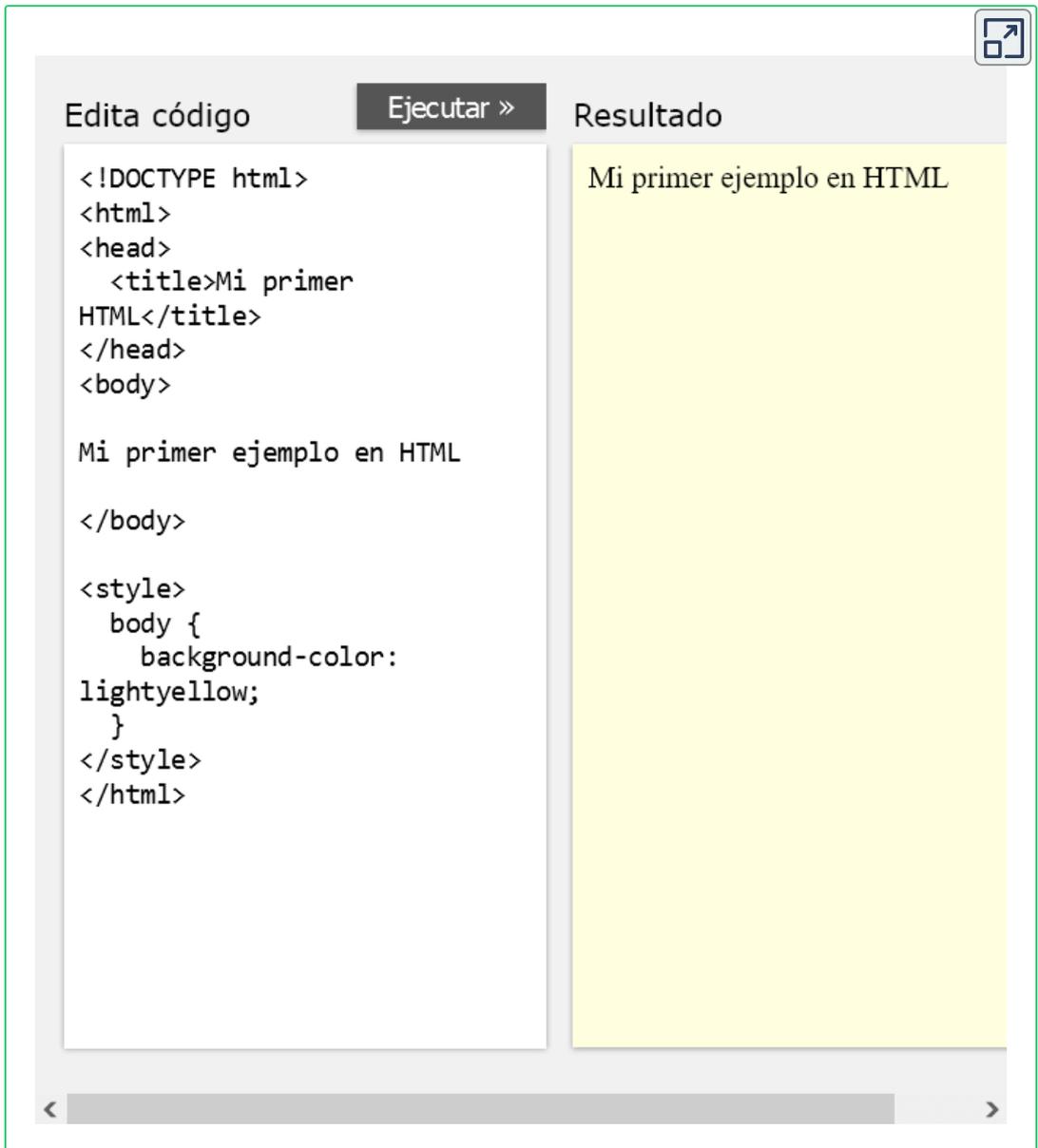
<head>
<title>Primer ejemplo</title>
</head>

<body>
Mi primer ejemplo en HTML
</body>

</html>
```

Mi primer ejemplo en HTML

¡Intenta con este!



The image shows a web editor interface with three main sections: "Edita código", "Ejecutar >>", and "Resultado".

**Edita código:** Contains the following HTML code:

```
<!DOCTYPE html>
<html>
<head>
  <title>Mi primer
HTML</title>
</head>
<body>

Mi primer ejemplo en HTML

</body>

<style>
  body {
    background-color:
lightyellow;
  }
</style>
</html>
```

**Ejecutar >>:** A button to execute the code.

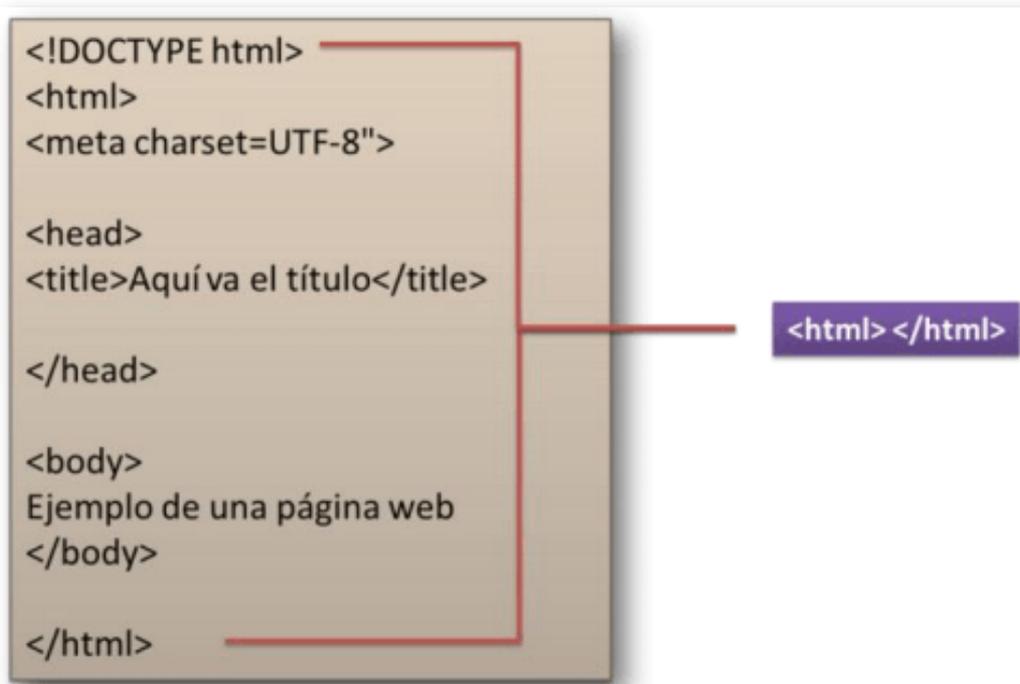
**Resultado:** Displays the rendered output: "Mi primer ejemplo en HTML". The background of this section is light yellow, matching the CSS style defined in the code.

At the bottom of the editor, there is a horizontal scrollbar with left and right arrow buttons.

Ahora, expliquemos el código anterior<sup>3</sup>:

**La etiqueta `<!DOCTYPE html>`** declara un documento HTML5 para que se visualice correctamente.

**La etiqueta `<html>`** declara que está codificado en este lenguaje y limita el principio y el fin del documento.



```
<!DOCTYPE html>
<html>
<meta charset=UTF-8">

<head>
<title>Aquí va el título</title>

</head>

<body>
Ejemplo de una página web
</body>

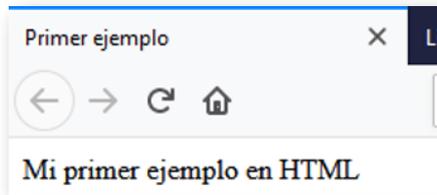
</html>
```

The diagram shows a block of HTML code on the left. A red line starts from the opening `<html>` tag, goes down, then right, then up, and finally right to point to a callout box on the right. The callout box is purple and contains the text `<html> </html>`.

**La etiqueta `<head>`** o cabecera, que contiene información y recursos sobre el propio documento y que no aparece en el documento, en la que se destaca el título `<title>`, que aparecerá en la pestaña del navegador.

---

<sup>3</sup> Para una mejor comprensión de este apartado, recomendamos escribir el código en el editor de textos planos, guardar la página creada y ejecutarla en un navegador (haciendo doble clic sobre el archivo).



La etiqueta `<meta charset='utf-8'>` define varios tipos de metadatos. El atributo **charset** especifica la codificación usada en nuestra página, **charset utf-8** es la *Unicode Transformation Format 8-bit*, que representa el código de caracteres UNICODE compatible con ASCII y permite visualizar los caracteres de todos los idiomas.

La etiqueta `<body>` es el cuerpo del documento, que contiene los elementos visibles en la página: las imágenes, los encabezados, los párrafos de texto, las listas, las tablas, los hipervínculos, escenas de Descartes, etcétera. Los documentos HTML deben tener la extensión html o htm para que puedan ser reconocidos por un navegador.

```
<!DOCTYPE html>
<html>
<meta charset=UTF-8">

<head>
<title>Aquí va el título</title>

</head>

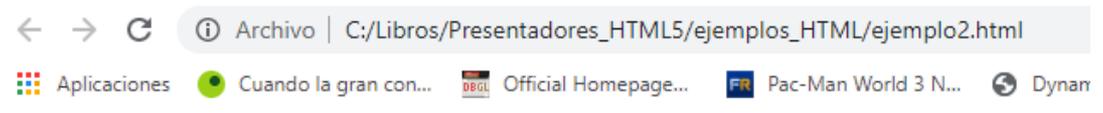
<body>
Ejemplo de una página web
</body>

</html>
```

A diagram illustrating the mapping between HTML code and its rendered output. On the left, a code block shows the HTML structure. A red rectangular box highlights the content between the `<body>` and `</body>` tags, which is the text "Ejemplo de una página web". A red line extends from the right side of this box to a purple rectangular box on the right that contains the text `<body> </body>`, representing the rendered structure of the body element.

# 1.4 Textos habituales en HTML

Un documento HTML inserta en el texto **etiquetas** que contienen los diferentes aspectos de la presentación. Los textos habituales están formados por encabezados `<h1>`, `<h2>`, `<h3>`,..., párrafos `<p>`, listas `<ul><li>`, etcétera. En el código de la siguiente página, hemos incluido varios de estos textos. En el siguiente vídeo puedes observar su edición en Notepad++ y cómo aparece en el navegador Chrome:



## Texto en nivel 1

## Texto en nivel 2

## Texto en nivel 3

Aquí va un párrafo de texto

- Primer elemento de una lista
- Segundo elemento de una lista
- Tercer elemento de una lista

Las **etiquetas o encabezados** `<h1>`, `<h2>` y `<h3>` muestran textos en diferentes tamaños. Nuestras diapositivas utilizarán, especialmente, los dos primeros encabezados, cuyos estilos (tamaños, colores, fuentes, ...) son modificados en cada presentador.

¡Intenta con este!



Edita código

Ejecutar »

Resultado

```
<!DOCTYPE html>
<html>
<head>
  <title>Mi primer
HTML</title>
</head>
<body>
<h1>Texto en nivel 1</h1>
<h2>Texto en nivel 2</h2>
<h3>Texto en nivel 3</h3>
<p>Aquí va un párrafo de
texto</p>
<ul>
<li>Primer elemento de una
lista</li>
<li>Segundo elemento de una
lista</li>
<li>Tercer elemento de una
lista</li>
</ul>

</body>

</html>
```

# Texto en nivel 1

## Texto en nivel 2

### Texto en nivel 3

Aquí va un párrafo de texto

- Primer elemento de una lista
- Segundo elemento de una lista
- Tercer elemento de una lista



El método de cómo son cambiados los estilos de estas etiquetas, no es objeto de este apartado, puesto que sólo nos interesa cómo funciona la etiqueta; no obstante, basta con saber que se trata de hojas de estilo **CSS**. Si observamos el contenido de los presentadores, seguramente encontraremos archivos con esta extensión.

La etiqueta párrafo `<p>` presenta un texto en el formato más simple del navegador.

Las listas `<ul>` permiten mostrar elementos (tipo viñeta). Cada elemento está encapsulado en una etiqueta `<li>`.

Otras etiquetas usadas para resaltar textos son: `<b>negrita</b>`, `<i>itálica</i>` y `<strong>texto fuerte</strong>`.

## 1.5 Cambios de estilo - CSS

Por defecto, una página HTML tiene asignados valores CSS<sup>4</sup> tanto en el cuerpo del documento, como en las etiquetas antes descritas. En general, variando de un navegador a otro, los valores para el cuerpo `<body>` podrían ser:

```
1 body {
2     font-family: "Times New Roman";
3     font-size: 20px;
4     color: #000;
5     background-color: #fff;
6     margin: 8px;
7 }
```

---

<sup>4</sup> CSS es el acrónimo de *Cascading Style Sheets* (Hojas de Estilo en Cascada)

En nuestros presentadores hay tres formas de intervenir el estilo de las páginas:

- **En las etiquetas.** Es el método más sencillo, añadiendo un atributo **style** en un elemento concreto dentro de la página, que afecta sólo a dicho elemento.
- **En la cabecera de la página <head>.** Este método permite aplicar un estilo a varios elementos de la página.
- **En una hoja de estilo externa.** Lo más recomendable y usado en nuestros presentadores. A través de un elemento <link> en la cabecera, se importan los estilos. En este capítulo no nos ocuparemos de este método.

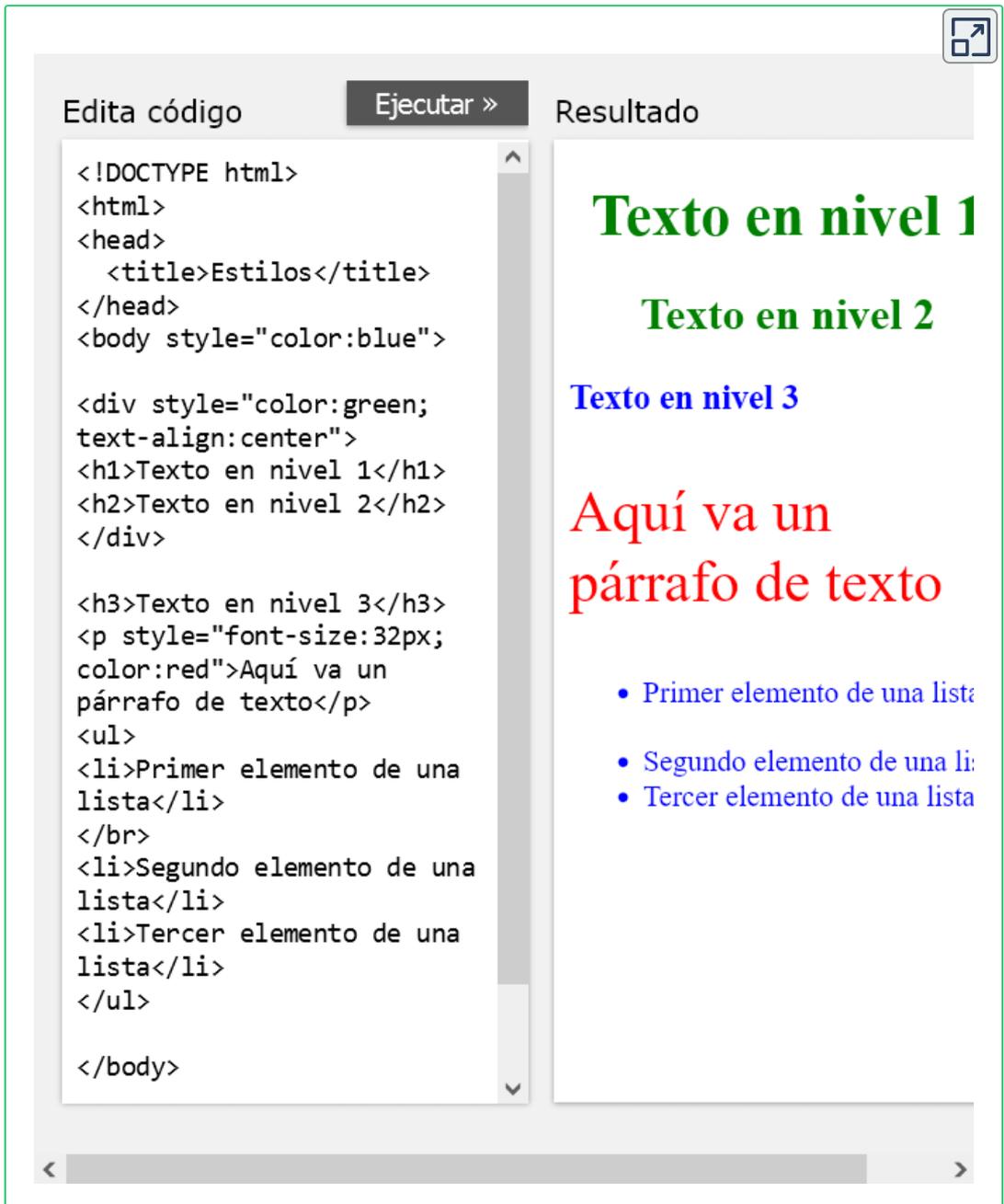
Los cambios de estilo que haremos en los presentadores serán sutiles, en tanto que algunos de ellos presentan varias hojas de estilo que podemos elegir y, seguramente, suplirán nuestras expectativas. Algunos cambios serán en colores de letra y de fondo de las diapositivas o, como en el caso del presentador Bespoke, cambio en el tipo de letra.

### 1.5.1 Cambios de estilo en las etiquetas

Es posible definir algunos estilos dentro del cuerpo del documento HTML. Algunos de ellos son:

**Cambio de estilo en el <body> y en las etiquetas.** Incluimos los siguientes cambios de estilo (style), en el ejemplo anterior:

¡Intenta con este!



The screenshot shows a web editor interface with two main panels: "Edita código" (Edit code) and "Resultado" (Result). The "Edita código" panel contains the following HTML code:

```
<!DOCTYPE html>
<html>
<head>
  <title>Estilos</title>
</head>
<body style="color:blue">

<div style="color:green;
text-align:center">
<h1>Texto en nivel 1</h1>
<h2>Texto en nivel 2</h2>
</div>

<h3>Texto en nivel 3</h3>
<p style="font-size:32px;
color:red">Aquí va un
párrafo de texto</p>
<ul>
<li>Primer elemento de una
lista</li>
</br>
<li>Segundo elemento de una
lista</li>
<li>Tercer elemento de una
lista</li>
</ul>

</body>
```

The "Resultado" panel displays the rendered output of this code. It features a blue background. A green container with centered text contains a green h1, a green h2, and a blue h3. Below this is a large red paragraph and a blue bulleted list with three items.

El color azul afectará a todos los textos encapsulados en el `<body>`, excepto en el texto tipo párrafo que, también, ha cambiado de estilo en color y tamaño de letra. Este tipo de modificaciones las usaremos en los presentadores; sin embargo, en una presentación uniforme, se recomienda el cambio de estilo en la cabecera (segundo método). Las modificaciones anteriores se muestran en el editor [slidear](#), en la página anterior. Practica cambiando el color y el tamaño de letra.

## 1.5.2 Cambios de estilo en una etiqueta `div`

Otro método que usaremos, es encapsular dos o más elementos en un bloque `<div>`<sup>5</sup>, el cual definirá un estilo para todos los elementos que contiene. Observemos, por ejemplo, este cambio en la página anterior, en la que hemos encapsulado las cabeceras `<h1>` y `<h2>`, las cuales se centrarán en la pantalla y tendrán color verde. Incluimos, además, un salto de línea `<br>` después del primer elemento de la lista.

Practica, nuevamente, usando colores en formato hexadecimal (véase <https://htmlcolorcodes.com/es/>); por ejemplo, en lugar de **green** escribe **#750A15** y haces clic en ejecutar.

---

<sup>5</sup> La etiqueta `<div>` define una división. Esta etiqueta permite agrupar varios elementos de bloque (párrafos, encabezados, listas, tablas, divisiones, etc). En principio, los navegadores no muestran nada especial cuando se crea una división, salvo que se dé formato a la división con la hoja de estilo.

## 1.5.3 Cambios de estilo en la cabecera

En nuestro próximo ejemplo, hemos eliminado los estilos en línea o en las etiquetas; ahora, hemos incluido un estilo CSS en un bloque `style` en la cabecera `<head>`. Observemos que se han declarado atributos para ocho elementos:

```
<body>, <h1>,<h2>,<h3>, <p>, <strong>, <em>, <li>
```

Los atributos se definen dentro de llaves.

Los cambios realizados fueron los siguientes:

**body { font-family: Arial, Helvetica, sans-serif; background-color: #FDC;}**. El atributo `font-family` establece la fuente o tipo de letra con la que se muestra el texto, al indicar una serie de fuentes alternativas separadas por comas, los navegadores intentan mostrar el texto con la primera fuente y si no está disponible, intentan utilizar la segunda fuente y así sucesivamente hasta la última fuente indicada. Se incluye, además, un color de fondo para la página `#FFDDCC` (que se puede simplificar como `#FDC`).

Practica cambiando la primera fuente (Arial) por algunas de las siguientes: Script, Courier, Verdana, Calibri o Garamond, preferiblemente en dos navegadores diferentes, pues es importante no arriesgarse a usar una fuente que no funcione en todos los navegadores; por ejemplo, la fuente "Script" va bien en Firefox pero no en Chrome. Nuestro consejo es dejar la fuentes universales como Arial, Verdana, sans-serif, etc.

¡Intenta con este!



Edita código

Ejecutar »

Resultado

```
<!DOCTYPE html>
<html>
<head>
  <title>Estilos en la
cabecera</title>
  <style type="text/css">
    body {
      font-family:Arial,
Helvetica, sans-serif;
background-color: #FFDDCC}
    h1 {color:red;
font-size: 1.6em;
background-color: #CCC}
    h2 {font-size:
1.2em; background-color:
#CCC}
    h3 {font-size:
22px; text-align: center}
    p {background-
color: #E8E8E8}
    strong {color: red;
background-color: #BBB}
    em {color: red;
background-color: #E8E8E8}
    li {color: brown}
  </style>
</head>

<body>
```

**Texto en nivel 1**

**Texto en nivel 2**

**Texto en nivel 3**

Aquí va un párrafo de *texto*

- Primer elemento de una lista
- Segundo elemento de un lista
- Tercer elemento de una lista
- *Cuarto elemento enfatiza*
- **Quinto elemento fuerte**

**h1 {color: red; font-size: 2.8em; background-color: #CCC}**. Los textos en encabezados de primer nivel tendrán color rojo, un fondo para el texto de tonalidad gris (#CCCCCC) y un tamaño de 1.6em. Esta última unidad de medida (em) determina un valor de escala con respecto al tamaño por defecto; es decir, si el tamaño es de 16 píxeles, el tamaño de letra definido por 1.6em será de 25.6 píxeles.

**h2 {font-size: 1.2em; background-color: #CCC}**. Los textos en encabezados de segundo nivel tendrán un tamaño de 1.2em. Se incluye un fondo para el texto de tonalidad gris.

**h3 {font-size: 22px; text-align: center}**. Para este caso, el tamaño de letra se define con unidades en píxeles. Un atributo adicional es el centrado horizontal del texto.

**p {background-color: #E8E8E8}**. Sólo se ha modificado el fondo del texto en la etiqueta párrafo con un color gris claro.

**strong {color: red; background-color: #BBB}**. Letra de color rojo y fondo en tonalidad gris oscura.

**em {color: red; background-color: #E8E8E8}**. Letra de color rojo y fondo en tonalidad gris claro.

**li {color: #4a4acc}**. Los textos de los elementos de una lista irán en color azul.

Se incluyeron etiquetas para negrilla **<b>** e itálicas **<i>** en las líneas de código de la página.

Recomendamos practicar con estos elementos mínimos, antes de emprender la tarea de modificar las presentaciones en los diferentes capítulos de este libro. Sin embargo, en cada presentador, estaremos recordando parte de lo aquí explicado.



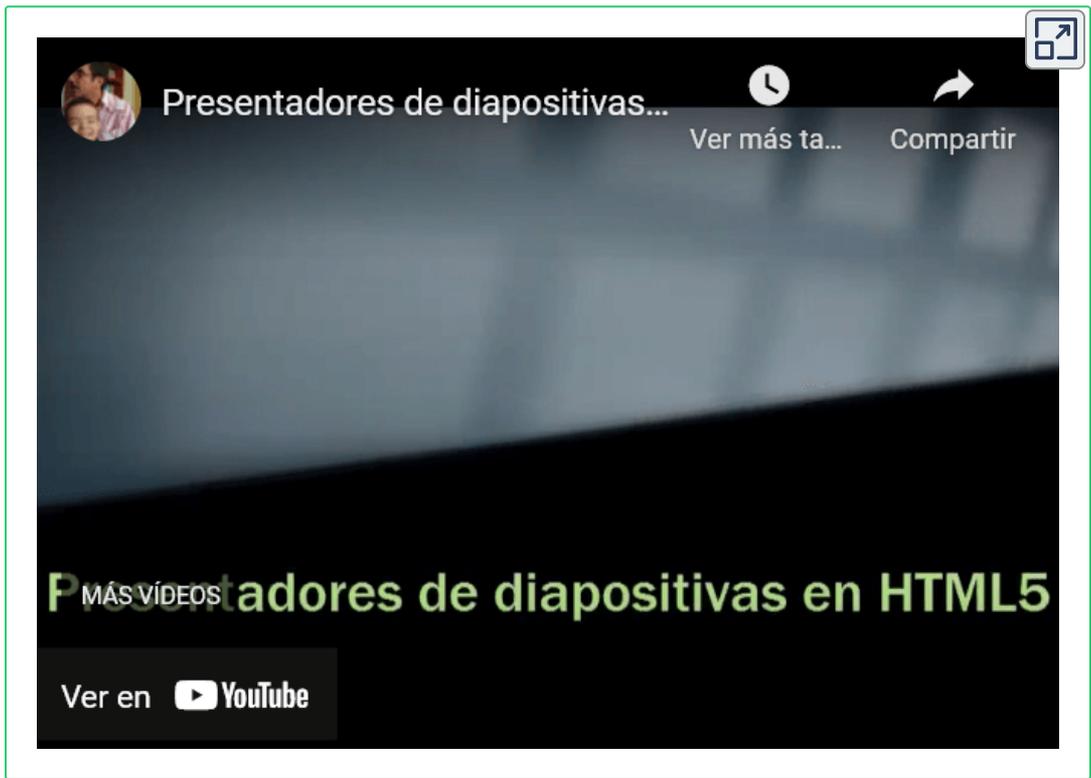
# Capítulo II

**Diapositivas con  
desplazamiento en el plano**



## 2.1 Introducción

Los presentadores que trabajaremos en este capítulo, fueron obtenidos luego de una búsqueda en la red que, antes de iniciar, os comparto un vídeo que da cuenta de cómo los pude hallar. Son trucos de un ratón de biblioteca o, en nuestro contexto, de un ratón de Internet que, seguramente, os ahorrará mucho tiempo para encontrar otros tipos de presentadores de diapositivas; sin embargo, es sólo un referente para búsquedas futuras, ya que debemos concentrarnos, por ahora, en la intervención de cinco presentadores iniciales.



Las bases de este libro surgen en 2014, cuando desarrollamos un taller con miembros de la Red Educativa Digital Descartes. En ese entonces, trabajamos cinco presentadores de diapositivas con desplazamiento en el plano; ahora, seis años después, existe una mayor creatividad, propiciada por los desarrollos en torno al HTML5. De esa gran variedad, hemos seleccionado 15 presentadores que, seguramente, algunos de ellos serán de tu agrado. Los presentadores son los siguientes:

- Deck.js
- Deck narrator
- Google
- DZSlides
- DZSlides narrator
- Fathom
- CSSlider
- flowtime.js
- Fluid CSS3 Slideshow
- Pow
- Presenteer.js
- Shower Presentation Engine
- Slider.css
- Weblides
- Wobbly Slideshow Effect

Cada uno de estos presentadores se suministran con este libro o, si lo prefieres, puedes descargarlo haciendo clic en la imagen zip que aparece con cada presentador. Se han hecho algunas intervenciones para su adaptación a escenas interactivas pero, en general, se conservan las características iniciales.

## 2.2 Presentador Deck.js



Pon el ratón sobre el presentador y observarás las flechas de desplazamiento en el plano. Puedes, también, usar las flechas del teclado para desplazar las diapositivas, luego de haber hecho clic sobre la primera diapositiva.

En el ejemplo, hemos diseñado cinco diapositivas, modelo que seguiremos usando en los demás presentadores con la inclusión. Hemos dejado dos diapositivas de la versión original, para que las estudies y comprendas su funcionamiento.

El presentador Deck.js genera diapositivas con desplazamiento horizontal, vertical o tipo fade (desvanecimiento). El desplazamiento se realiza con la barra espaciadora, las teclas de dirección o haciendo clic en las flechas de los extremos, también funciona con un control remoto de diapositivas.

Deck.js fue diseñado por Caleb Troughton, con Licencia MIT. Copyright (c) 2011-2014, así:

Se concede permiso, de forma gratuita a cualquier persona que obtenga una copia de este software y archivos de documentación asociados (el " Software "), para trabajar con el Software sin restricción, incluyendo sin limitación, los derechos para usar, copiar , modificar, fusionar, publicar , distribuir, sublicenciar y/o vender copias del Software...

## Procedimiento de edición

Para nuestro diseño en el presentador Deck.js, debemos tener en cuenta dos aspectos. El primero es el estilo y las transiciones; el segundo, es el diseño de las diapositivas. Para el primer aspecto, te invitamos a ver el siguiente vídeo.



El segundo aspecto es el diseño las diapositivas. Tal como lo observaste en el vídeo, debes abrir el archivo **index.html** de la carpeta Deck que ya debiste descargar o, si tienes el libro, la puedes encontrar en una carpeta que llamamos "Desplazamiento". El archivo lo abres con un editor de textos planos, como el Notepad++.

En este presentador, cada diapositiva se encuentra en un bloque llamado "section", que inicia con la etiqueta **<section class="slide">** y termina con **</section>**, nuestra edición sólo se hará en el contenido entre estas dos etiquetas. Estas diapositivas se encuentran al inicio del **<body>**:

```
<section class="slide">
  <h2 style="text-align:center">
    Desplazamiento en el plano</h2>
  <h1>Deck.js</h1>
</section>
```

El ejemplo que adjuntamos tiene siete diapositivas. Las tres últimas son tomadas del presentador original de [Troughton](#), las cuatro primeras las explicamos así:

### **Primera diapositiva.** Incluye solo texto

Cada diseño de estilos, determina tipos de fuentes, posición, color y tamaños. En el Deck, la etiqueta `<h1>` escribe un texto de gran tamaño (4.5 em<sup>6</sup>) centrado en la diapositiva. La etiqueta `<h3>`, de unos 20 píxeles de tamaño, presenta el texto en la parte superior izquierda de la diapositiva, a la cual le incluimos un estilo para centrar el texto (`style="text-align:center"`). Es importante tener en cuenta este diseño, para evitar textos sobrepuestos sobre imágenes o vídeos.

Si deseamos incluir otras diapositivas de este tipo, sólo tenemos que copiar este bloque e insertarlo donde queramos que se presente.

**Segunda diapositiva.** Hemos incluido una imagen, que se incorpora a través de una etiqueta `<img>`, la fuente puede ser local, como en este caso, o una dirección de la web.

---

<sup>6</sup> 1.29 em equivale a 18 píxeles

```
<section class="slide">
  <h4>Una imagen</h4>
  </img>
  <pre>Mensaje en un postit</pre>
</section>
```

**Tercera diapositiva.** Se trata de un vídeo de YouTube, que se embebe en un iframe. Notemos que las dimensiones tienen un formato diferente al que usamos en la diapositiva anterior; sin embargo, ambas son válidas en JavaScript.

```
<section class="slide">
  <h2>Un vídeo</h2>
  <iframe src=
    "http://www.youtube.com/embed/ctCGA0VVaak?rel=0"
    allowfullscreen="" frameborder="0" height="800"
    width="1280"></iframe>
</section>
```

**Cuarta diapositiva.** Hemos incluido una escena de Descartes, a través de una etiqueta `<iframe>`<sup>7</sup>. En este caso, se trata de un objeto del proyecto Plantillas. Es importante incluir un estilo que defina las dimensiones del objeto (ancho: `width` y alto: `height`). La fuente `src` (source) puede ser una dirección de la página del proyecto o, si lo deseas, una dirección local.

---

<sup>7</sup> La etiqueta HTML `<iframe>` (de inline frame) representa un contexto de navegación anidado, el cual permite incrustar o embeber otra página HTML en la página actual (<https://www.w3schools.com>).

```
<section class="slide">
  <h3 style="text-align:center">Una escena
  interactiva</h3>
  <iframe height="800" width="1280" src=
  "https://proyectodescartes.org/plantillas/materiales_
  didacticos/Seleccion_multiple10-JS/index.html"
  ></iframe>
</section>
```

## 2.3 Presentador DZSlides



Haz clic sobre la primera diapositiva y luego usa las teclas de desplazamiento del teclado. Puedes presionar la letra **f** (full), para ver la presentación en pantalla completa.

Excelente presentador con características especiales que describiremos a continuación. Hacemos copia de la licencia (<https://github.com/paulrouget/dzslides>).

Everyone is permitted to copy and distribute verbatim or modified copies of this license document, and changing it is allowed as long as the name is changed (Do what the fuck you want to public license terms and conditions for copying, distribution and modification).

Resumiendo... puedes hacer lo que quieras con la licencia.

Además de las teclas de desplazamiento, el presentador también responde a las teclas "inicio" y "fin".

Similar al presentador Deck.js, una sección `<section>` es una diapositiva. Su tamaño es de 800×600 encapsuladas en un marco... todos los cambios, incluido el resize se hacen en el marco. Describimos solamente algunas diapositivas que tengan algo novedoso (recuerda abrir el archivo index.html con el editor de textos planos).

**Primera diapositiva.** Incluye un pie de página `<footer>`

```
<section>
  <!-- Primera diapositiva -->
  <h1>Presentación</h1>
  <footer>por Juan Guillermo Rivera Berrío</footer>
</section>
```

Observa, en la página anterior, el diseño resultante al usar estas dos etiquetas.

**Segunda diapositiva.** Esta diapositiva sólo usa la etiqueta párrafo `<p>`.

```
<section>
  <p>Tipo de presentador sencillo, con
  desplazamiento horizontal en el mismo marco.
  Funcionan bien las escenas de Descartes
  embebidas. He usado una del Proyecto Canals de
  tamaño 820x590</p>
</section>
```

Es posible que no estés satisfecho con el tamaño de la letra, por lo que puedes intervenir la hoja de estilos; por ejemplo, para la etiqueta párrafo, en la línea 131 del archivo `index.html`, puedes cambiar el tamaño de la fuente que, inicialmente, es de 50px.

```
131   p {
132     margin: 75px;
133     font-size: 50px;
134   }
```

**Tercera diapositiva.** Hemos incluido un objeto interactivo del Proyecto Canals que, pese a su tamaño, el presentador lo ajusta al marco. En este caso hemos usado un objeto que se encuentra en una de las carpetas del libro.

```
<section>
  <iframe style="width: 820px; height: 590px;" src=
  "puzle/index.html"></iframe>
</section>
```

**Diapositivas con imágenes.** Estas diapositivas tienen una característica especial, en tanto que permite usar una etiqueta **<figure>** para escalar las imágenes al tamaño del marco 800×600, lo que obliga a utilizar imágenes cercanas a este valor o de tamaño superior para evitar distorsiones; sin embargo, si tu diseño exige que la imagen se presente en un tamaño menor, sólo debes eliminar esta etiqueta adicional. En nuestro ejemplo, hemos usado dos imágenes, una de formato jpg y otra en png.

```
<section>
  <figure>
    
  </figure>
</section>

<section>
  <figure>
    
  </figure>
</section>
```

**Diapositivas con vídeos.** En esta diapositiva se usa el formato HML5 para vídeos **<video>**. Siempre que se use formatos de vídeo webm o mp4, se contará con las siguientes ventajas: ejecución automática, detención del vídeo cuando se pasa a otra diapositiva, reanudación cuando se regresa a la diapositiva. Se pueden hacer comparaciones con el vídeo YouTube incluido en la presentación y los dos vídeos webm y mp4.

Se podrá notar que el sonido del vídeo YouTube se seguirá escuchando cuando se pasa a otra diapositiva, advirtiendo que aún se está reproduciendo.

```
<section>
  <figure>
    <video src=
      "https://upload.wikimedia.org/wikipedia/commons/
      5/5f/2015-08-13_-_Leica_I&2C_1927_-_lange_Fassun
      g_mit_Musik.webm"></video>
    <figcaption>Ejection automatic</figcaption>
  </figure>
</section>
```

En la siguiente escena interactiva, diseñada con DescartesJS, puedes comprobar la ventaja enunciada. La escena tiene enlaces a vídeos en formato mp4, webm y ogv, este último no es reconocido por DZSlides.



**Vídeos en formatos compatibles con HTML5**

Haz clic en el botón para observar el primer vídeo

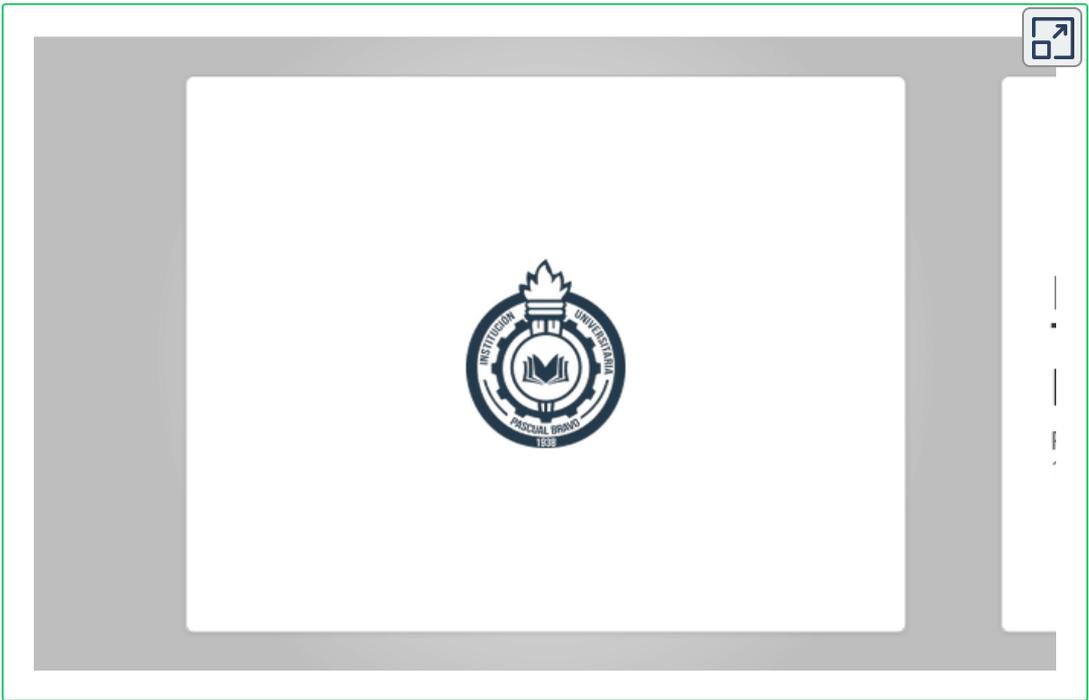
The image shows an interactive scene with a yellow background. On the left, there is text in Spanish: "Haz clic en el botón para observar el primer vídeo". In the center, there is an illustration of a laptop displaying a video player interface with a play button and a progress bar. To the left of the laptop is the HTML5 logo. In the top right corner, there is a small icon of a square with an arrow pointing outwards. In the bottom right corner, there is a circular play button icon.

Otra característica interesante de DZSlides es que tanto el CSS como JavaScript ya están incluidos en el archivo, lo que facilita su manejo para el diseño de nuestras diapositivas. La navegación básica por teclado es compatible con controles táctiles compatibles con dispositivos móviles ([Hipp, 2019](#)).

En el siguiente vídeo puedes observar un resumen de lo explicado en este apartado y el funcionamiento del presentador.



## 2.4 Presentador Google I/O Slides



Haz clic en medio de las diapositivas, o usa las teclas de desplazamiento o la barra espaciadora.

Este presentador, diseñado por Luke Mahé y Marcin Wichary, fue una iniciativa de <https://code.google.com>, que inicialmente se llamó **html5slides**, tuvo una segunda versión en 2014; sin embargo, el proyecto fue abandonado para dar paso a las versiones Google Slides<sup>8</sup>.

---

<sup>8</sup> Google aloja una suite ofimática gratuita en línea como parte de Google Drive. Esto incluye Google Slides, una herramienta de presentación en línea compatible con PowerPoint, lo que significa que los usuarios pueden importar y exportar a Formato de archivo de PowerPoint libremente (ibid).

La licencia del código es [Licencia Apache 2.0](#), por lo que puedes usarlo sin que Google se enoje.

El presentador tiene características similares a las descritas en los presentadores anteriores, en lo que respecta al desplazamiento; sin embargo, en lugar de la etiqueta `<section>` para encapsular las diapositivas, usa la etiqueta `<article>`. Por otra parte, tiene algunos atributos adicionales que describiremos en algunas de las diapositivas que acompañan la carpeta adjunta a este documento.

## Procedimiento de edición

Abrimos el archivo `index.html` de la carpeta Google con un editor de textos planos.

**Modificación del diseño y los temas.** Google presenta dos tipos de diseño: `layoutregular`, `layout-faux-widescreen` y `layout-widescreen`. Además, dos temas diferentes: `template-default` y `template-io2011`. La combinación de diseños y temas están asociados a unas imágenes que Google imprime como pie de página y en el centro de la primera diapositiva. En la carpeta llamada `images`, aparecen cuatro imágenes con los nombres:

- `colorbar.png`
- `googleio-logo.png`
- `google-logo.png`
- `google-logo-small.png`

Inmediatamente después de la etiqueta **<body>**, aparece una etiqueta **<section>**, en el cual están definidos el diseño y el tema, tal como se muestra en la siguiente imagen:

```
<body style='display: none'>
<section class='slides layout-regular
template-default'>
```

Con esta configuración, la presentación es de esta forma:



Si hacemos esta combinación:

```
<section class='slides layout-widescreen
template-io2011'>
```

Obtendríamos tres cambios. El primero es el ancho de las diapositivas, el segundo es la imagen en la primera diapositiva y, el tercero, es la inclusión de la barra de Google, además de cambiar la imagen del pie de página.



Excepto por el cambio en el ancho, los cambios en el diseño del presentador no son significativos, por lo que es mejor dedicar el tiempo en el diseño de las diapositivas. Como lo dijimos antes, en este presentador, cada diapositiva se encuentra en un bloque llamado “article”, que inicia con la etiqueta `<article>` y termina con `</article>`. Nuestras modificaciones sólo se harán en el contenido entre estas dos etiquetas. Las modificaciones son similares a los presentadores anteriores, así que nos concentraremos en sólo algunas diapositivas especiales.

**Primera diapositiva.** Es la que incluye el logo o imagen centrada en la pantalla, se genera a través de este código:

```
<!-- Primera diapositiva -->
<article class='biglogo'></article>
```

Si no se desea incluir, simplemente se elimina.

**Diapositiva con textos.** Recomendamos explorar cómo funcionan las etiquetas de texto; sin embargo, adelantamos que, para el caso de `<h1>`, el texto es centrado verticalmente en la pantalla, el siguiente código:

```
<!-- Segunda diapositiva -->
<article>
  <h1>Presentación<br>Tipo desplazamiento
  horizontal</h1>
  <p>Red Educativa Digital Descartes<br>16 de
  abril de 2014</p>
</article>
```

Corresponde a la segunda diapositiva, que incluye: un texto con etiqueta `<h1>`, dos saltos de página, y un texto con etiqueta de párrafo `<p>`. La diapositiva generada es la siguiente:



**Uso de viñetas.** A través de la etiqueta para listas no ordenadas `<ul>` y `<li>` para los elementos de lista, se crea una diapositiva cuyos elementos irán apareciendo con los desplazamientos realizados.

Para listas ordenadas, se usaría `<ol class="build">`.

**Diapositivas de fondo.** El presentador Google tiene una opción para que las imágenes y los `<iframe>` ocupen el fondo de la diapositiva. Lo hemos hecho para tres de las diapositivas del presentador que acompaña este libro. Para ello usamos la clase `fill`.

```
<article class='fill'>
  <p><img src='imagenes/i4.jpg'></p>
  <div>
    <span class='yellow'>Imagen de fondo (con
      título opcional)</span>
  </div>
  <div class='source white'>
    Fuente: la red
  </div>
</article>
```

**Páginas web embebidas.** En los presentadores anteriores también se pueden incluir páginas web, usando una etiqueta `<iframe>`:

```
<article class='nobackground'>
  <h3>Diapositiva con iframe</h3>
  <iframe src='http://proyectodescartes.org/'
  ></iframe>
</article>
```

## 2.5 Presentador Fathom



Haz clic en la primera diapositiva y luego usa las teclas de desplazamiento, también puedes arrastrar las cinta de diapositivas con la barra scroll horizontal, que aparece en la parte inferior.

El presentador [Fathom](#) es muy similar al Google I/O Slides pero, como se observa arriba, las diapositivas de los extremos presentan un nivel de opacidad, que permite destacar la diapositiva central. Este presentador fue diseñado por Marcos Dalglish en 2011, con la siguiente licencia:

Se concede permiso, de forma gratuita, a cualquier persona que obtenga una copia de este software y archivos de documentación asociados (el "Software").

Para trabajar con el Software sin restricción, incluyendo sin limitación, los derechos para usar, copiar, modificar, fusionar, publicar, distribuir, sublicenciar y / o vender copias del Software.

Una primera modificación, que podemos realizar, es cambiar el estilo de presentación, que por defecto es "multi", tal como aparece al inicio del archivo index.html:

```
<script type="text/javascript">
$(document).ready(function() {
    $("#presentation").fathom({
        displayMode: 'multi'
    });
});
</script>
```

Si lo cambiamos por "single", el presentador tendrá similitudes al Deck.js y DZSlides.



El diseño de las diapositivas es similar a como lo hemos hecho en los tres presentadores anteriores, por lo que no nos detendremos a explicarlo. La diferencia se encuentra en su contenedor que, para este caso, se usa la etiqueta de bloque `<div>`. Observa, por ejemplo, la primera diapositiva:

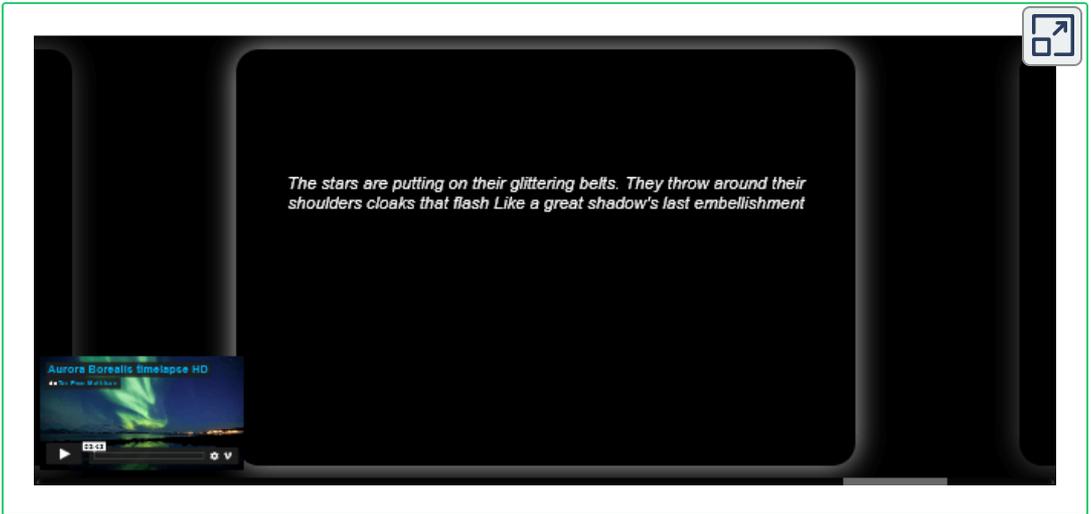
```
<div class="slide" align="center">
  <h1 style="font-size:45px; color:
  red;">Proyecto UN_100</h1>
  <br></br>
  
</div>
```

En nuestro ejemplo, sólo por practicar, hemos puesto tres escena interactivas del Proyecto Un\_100 y algunas referidas a las nuevas etiquetas del HTML5.

Una característica que hace especial este presentador, es la posibilidad de sincronizar la presentación con un vídeo de **Vimeo**, a través de una línea de tiempo:

```
$ ( '# presentación' ). fathom ({
  línea de tiempo : [ 0 , 5 , 20 , 30 , 40 , 50 , 60 , 75 , 90 , 120 , 155 ] .
```

Según la imagen anterior, la segunda diapositiva se presentará, automáticamente, a los cinco segundos, la tercera a los 20, la cuarta a los 40, etc. A continuación, presentamos el ejemplo que trae el presentador original. Te invitamos a que lo explores, pues además de la línea de tiempo, se debe incluir la fuente del vídeo y su código.



Haz clic sobre la primera diapositiva y debe aparecer, en la parte inferior izquierda, el vídeo, reproducéelo... y observa... y ¡escucha!

Este ejemplo lo encuentras en la carpeta **test** con una línea de tiempo en diferentes formato, pero puedes cambiarlos con números únicamente:

```
$( function() {  
    $( "#presentation" ).fathom( {  
        timeline: [ 0, "5", "0:20", { time:  
30 }, { time: "40", slide: 4 }, {  
time: 50, slide: "#slide5" }, "1:00",  
"1:15", 90, "2:00", { time: "2:35",  
slide: 0 }  
    ],  
    video: {  
        source: "vimeo",  
        id: "16917950",  
        parent: "#vimeo",  
        autoplay: true  
    }  
}
```

## 2.6 Presentadores con narración

La educación en 2020 tuvo un giro extraordinario, pues la presencia de la Covid 19 obligó a la comunidad académica a volcarse a la mal llamada "educación virtual" que, en muchos países, la llamamos "educación remota con ayuda de las TIC". Para sorpresa de muchos, tal educación resultó más real que virtual, pues evidenció la realidad de docentes y discentes carentes de competencias digitales pero, lo más importante, es la desaparición del distanciamiento social en el "aula digital", en tanto que como docentes conocimos, cercanamente, la realidad, a veces cruel, de nuestros estudiantes; nos acercamos a sus angustias y agobios, a su desesperanza, a ese desespero generado por la falta de recursos, no sólo tecnológicos, que los llevaba al borde del colapso... de la deserción.

Pese a este escenario de caos, surge la resiliencia que da respuesta a la nueva normalidad, tanto nosotros los amantes de lo digital como sus detractores más acérrimos nos adaptamos a la "educación digital", encontrando una grandiosa solidaridad de colegas, productores de herramientas digitales y, para no extender la lista, de toda la sociedad que con gran esperanza quiso establecer un nuevo escenario, donde la educación tuviera a su disposición las mejores condiciones que la llevaran a continuar su labor **esencial**, para el desarrollo de las regiones y de los países.

Como es común en la educación presencial, muchos docentes continuaron exponiendo su conocimiento con una herramienta tradicional... "el presentador de diapositivas". Pero, en la nueva normalidad, la conectividad no es una de sus fortalezas, situación que ha obligado a los docentes a compartir sus presentaciones asincrónicamente, con un gran ausente: "su discurso" o, dicho de otro modo **"su narración"**. En este apartado, presentamos algunas alternativas, que permitan compartir presentaciones de diapositivas que incluyan la narración del docente.

## 2.6.1 PowerPoint narrador

Las presentaciones pueden tener un gran impacto en el mundo. El 5 de febrero de 2013, el Secretario de Estado de EE. UU. Colin Powell acudió al Consejo de Seguridad de las Naciones Unidas para hacer una presentación sobre el desarrollo de armas de destrucción masiva en Iraq. Sus 45 diapositivas de PowerPoint fueron bien organizadas y convincentes, y se consideran "una de las presentaciones de PowerPoint más famosas de todos los tiempos" ([Hipp, 2019](#)).

Sin demeritar el Keynote, lanzado para MacOS en 2003, como alternativa a PowerPoint, no podemos desconocer el cambio radical que el PowerPoint hizo en las otras diapositivas en acetatos que se proyectaban en el vetusto proyector. Pero, para sorpresa de muchos, el PowerPoint había nacido para Macintosh:

A mediados de 1980 dos productores de software en una pequeña oficina en California llamada Forethought, estaban desarrollando el primer software de presentaciones gráficas para la computadora personal. Los primeros esfuerzos de estos dos productores de software fueron poco recompensados. Una serie de compañías inversionistas declinaron invertir en dicho programa, que era conocido como Presenter y estaba dirigido a la plataforma Mac. Pero Bob Gaskins, dueño de la compañía Forethought, Inc.,<sup>2</sup> y el coproductor del programa, no perdieron las esperanzas. Después de dos años de negociaciones (1985-1987), vendieron por fin la compañía y su software a la gigante Microsoft de Bill Gates (<https://es.wikipedia.org>).

La primera versión de PowerPoint (1987) presentaba diapositivas a blanco y negro:

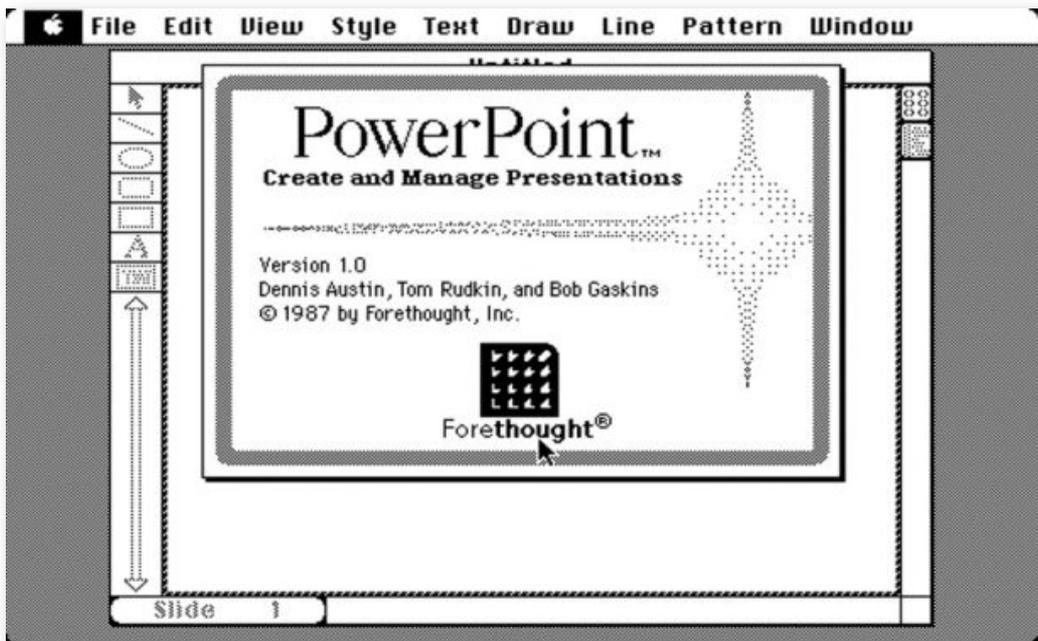
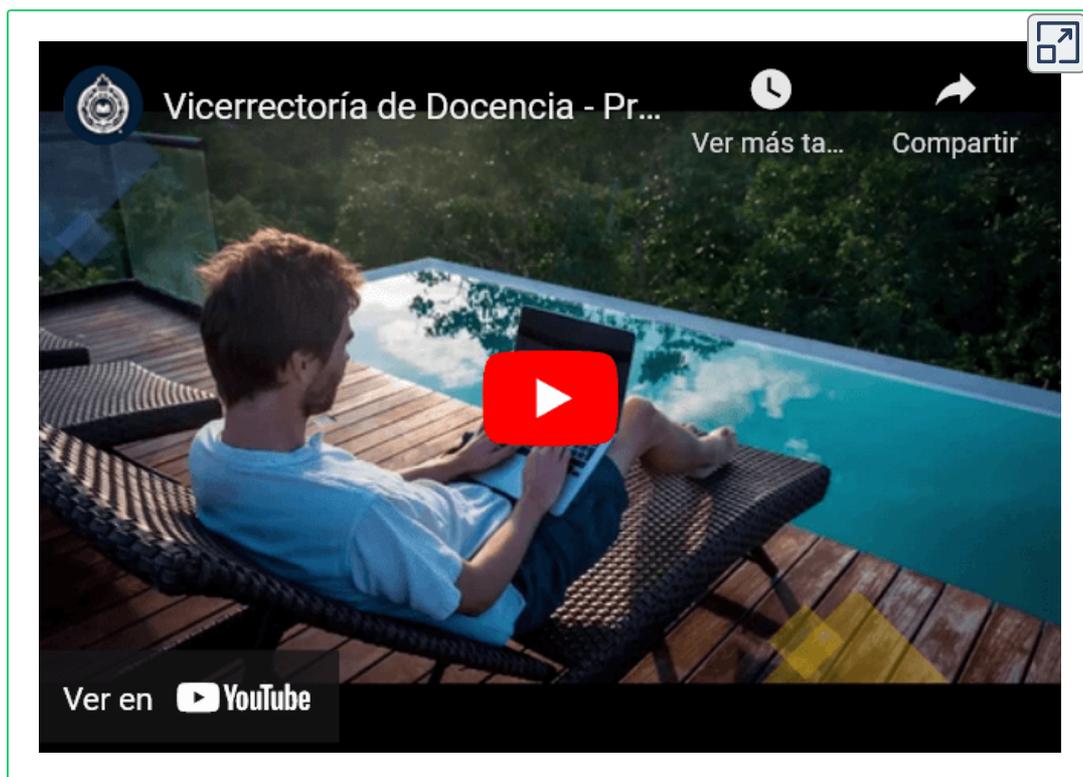


Imagen tomada de <https://blogthinkbig.com/>

Ahora, en 2020, el presentador de diapositivas PowerPoint ofrece nuevas alternativas, una de ellas la posibilidad de grabar audio por diapositiva, utilidad que permite compartir presentaciones de mayor calidad con nuestros estudiantes. La presentación obtenida se puede guardar con los audios o, si se prefiere, en formato vídeo, tal como se explica a continuación:



Incluimos, en este libro, el presentador PowerPoint, en tanto que al poder obtener un vídeo de la presentación con narración, éste se puede subir a un canal de YouTube y compartir en la web; no obstante, aún sigue impidiendo el encapsulamiento de escenas interactivas HTML5.

## 2.6.2 Presentador Deck narrador



Reproduce el audio en la barra de sonido y notarás que las diapositivas se presentan cada vez que termine la narración.

Esta versión del presentador Deck.js es una adaptación de [Kevin Lamping](#), en la que se incluyen archivos de audio para generar la narración de la presentación. El diseño de las diapositivas ya las explicamos en el apartado 2.2, así que sólo nos ocuparemos del archivo de audio.

A diferencia del PowerPoint narrador, que usa un archivo de audio por cada diapositiva, el Deck.js modificado, emplea un sólo archivo de audio, guardado en la carpeta **audio**.

El nombre del archivo es audio.mp3; es decir, debes crear un sólo archivo de sonido en el que vas narrando cada diapositiva. Ahora, si quieres cambiar el nombre de tu archivo de audio, también debes hacerlo en la etiqueta `<audio>` del archivo index.html:

```
86 <audio controls class="deck-narrator-audio" id=
    "narrator-audio">
87   <source src="audio/audio.mp3" type="audio/mp4" />
88   <source src="audio.ogg" type="audio/ogg" />
89   <track kind="caption" src="transcript.vtt" srclang=
    "en" label="English" />
90 </audio>
```

Finalmente, en cada diapositiva debemos escribir la duración del audio (el valor de cuánto tiempo debe reproducirse el audio en esa diapositiva), que corresponde a una porción del archivo audio.mp3; por ejemplo, la primera diapositiva dura 14 segundos, la segunda 34, la tercera 34, la cuarta 25, etc. Debes, entonces, calcular estas duraciones; por ejemplo, la tercera diapositiva inicia a los 48 segundos (14 + 34) y termina a los 82 o 1:22 (verifícalo en la presentación de la página anterior).

```
<section class="slide" data-narrator-duration="14"
  data-duration="14000">
  <h1>El efecto túnel</h1>
  <footer>Haz clic en el Play de la barra de
    sonido, para iniciar la presentación</footer>
</section>
```

El número indicado en `data-duration` está dado en milésimas de segundo.

## 2.6.3 Presentador DZSlides narrador



Haz clic en sobre la diapositiva y luego sobre la flecha de desplazamiento a la derecha, escucharás el audio a partir de la segunda diapositiva (en el zip, el audio inicia con la primera diapositiva). Para este caso, los audios se deben generar por cada diapositiva y guardarlos en la carpeta **sonidos**.

- sonido1.mp3
- sonido2.mp3
- sonido3.mp3
- sonido4.mp3
- sonido5.mp3
- sonido6.mp3

El funcionamiento del presentador lo explicamos en el apartado 2.3, por lo que sólo debemos indicar cómo incluir el audio en la diapositiva; para ello, observa las dos primeras:

```
10 <section>
11   <figure style="background-color:blue;" <!--
      Figure muestra imágenes and videos en el
      marco completo -->
12     
13     <audio src="sonidos/sonido1.mp3"></audio>
14   </figure>
15 </section>
16
17 <section>
18   <p>Segunda diapositiva con sonido. Continúa
      con la siguiente diapositiva o regresa a la
      anterior. Luego vuelve a esta diapositiva<br/>
      ¿Qué ocurre con el sonido?</p>
19   <audio src="sonidos/sonido2.mp3"></audio>
20 </section>
```

Obviamente, tienes la opción de incluir o no el audio; por ejemplo, la quinta diapositiva es un vídeo, por lo que el audio no se requiere.

Observa que los audios son cargados en la etiqueta `<audio>`, donde la fuente (`src`) u origen del audio se encuentra en la carpeta `sonidos`. Para la primera imagen, se carga el audio `sonido1.mp3`

```
<audio src="sonidos/sonido1.mp3"></audio>
```

## 2.7 Presentadores de diapositivas CSS3

Los presentadores anteriores fueron diseñados siete u ocho años atrás, ahora en 2020 los desarrollos de CSS3 ofrecen otras posibilidades como las de añadir animaciones y transiciones, que permiten cambiar la apariencia y el comportamiento de un elemento cada vez que se da un cambio de estado (por ejemplo, cuando el cursor se posa sobre dicho elemento). Por otro lado, las animaciones permiten que tanto la apariencia como el comportamiento de un elemento se altere en base a fotogramas (<https://blog.aulaformativa.com>).

CSS3 no sólo te permite modificar el color de los elementos que conforman tu página, sino que también te permite crear gradientes. Anteriormente, esto no era posible y tenías que emplear imágenes, pero ahora que puedes hacerlo desde el mismo archivo CSS. De esta manera, mejorar el tiempo de respuesta de tu sitio. La propiedad "Transform" de CSS3 no sólo te permite rotar elementos, sino también cambiar su tamaño y moverlos (Ibid).

Para nuestros propósitos, la ventaja más atractiva del CSS3 es la posibilidad de diseñar presentadores de diapositivas sin necesidad de recurrir al JavaScript; es decir, usando "puro CSS3". A continuación, presentamos dos ejemplos.

El primer ejemplo es un demo, que hemos incluido por el efecto de "paralaje" que tiene incorporado. El segundo es similar al primero pero, al ser libre, permite una fácil edición.

## 2.7.1 Presentador Fluid CSS3 Slideshow



Haz clic en las flechas o en los botones inferiores.

Este presentador fue diseñado por [Justin Nicholes](#). Incluye un efecto de paralaje que se evidencia en los movimientos del fondo, cada que desplazamos las diapositivas.

Dado que es un demo para un máximo de cinco diapositivas, no nos detendremos a explicar su edición; sin embargo, si sabes de CSS3, puedes descargar el presentador e intervenirlo.

Un presentador de diapositivas CSS3 con un diseño similar, sin el efecto paralaje, lo presentamos a continuación, incluyendo la explicación de cómo se puede intervenir, para crear nuestras presentaciones.

## 2.7.2 Presentador CSSslider



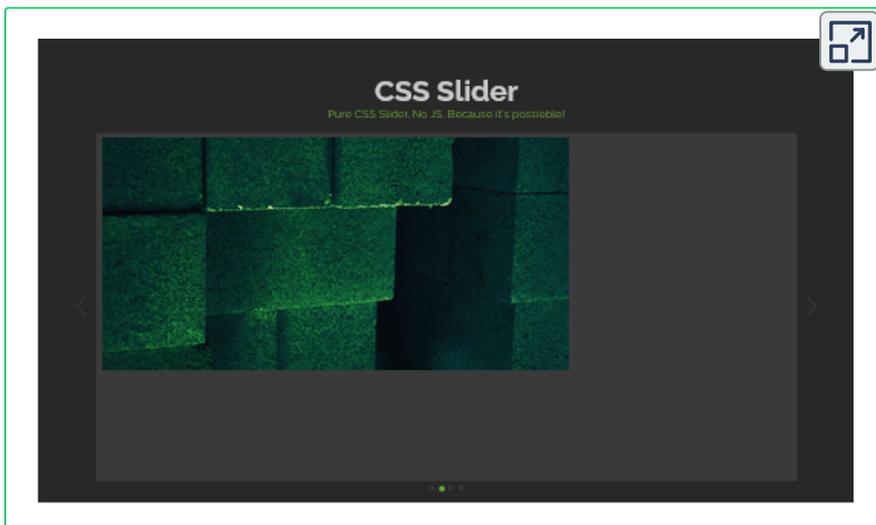
Similar al anterior, puedes pasar las diapositivas haciendo clic en las flechas de los extremos o en los radio botones.

Este presentador fue diseñado por [Damian Drygiel](#)

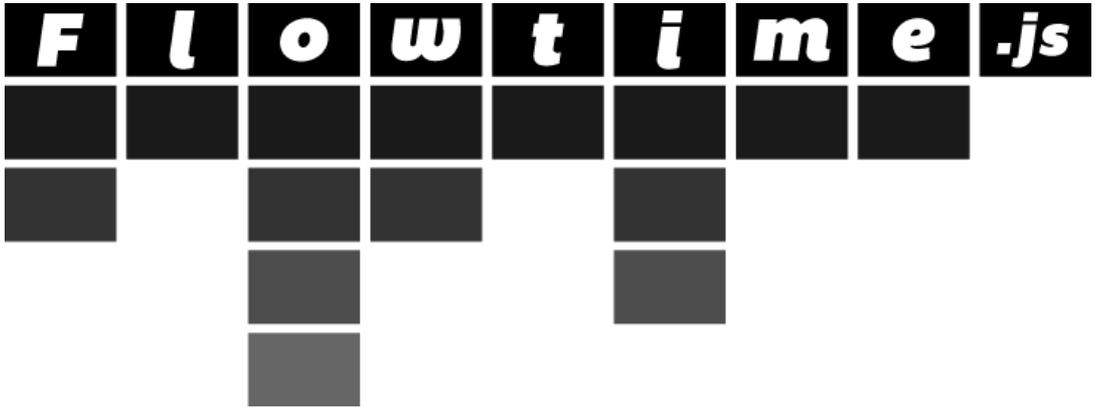
La edición del presentador se realiza en el archivo index.html. Las diapositivas se encuentran encapsuladas en etiquetas `<li>` y su edición es muy similar a lo que hemos hecho en los presentadores anteriores; sin embargo, te invitamos a que observes en el siguiente vídeo los aspectos más relevantes del proceso de edición.



Este es el presentador con estilo **dark**:



## 2.8 Presentador Flowtime.js



Flowtime.js permite crear presentaciones con un flujo en el plano, que puede ser horizontal, vertical o diagonal. Tiene algunos añadidos como imágenes superpuestas:



o soporte Parallax nativo. También soporta fragmentos para ir avanzando poco a poco en la misma diapositiva o resaltado de código (<https://www.emezeta.com/>)

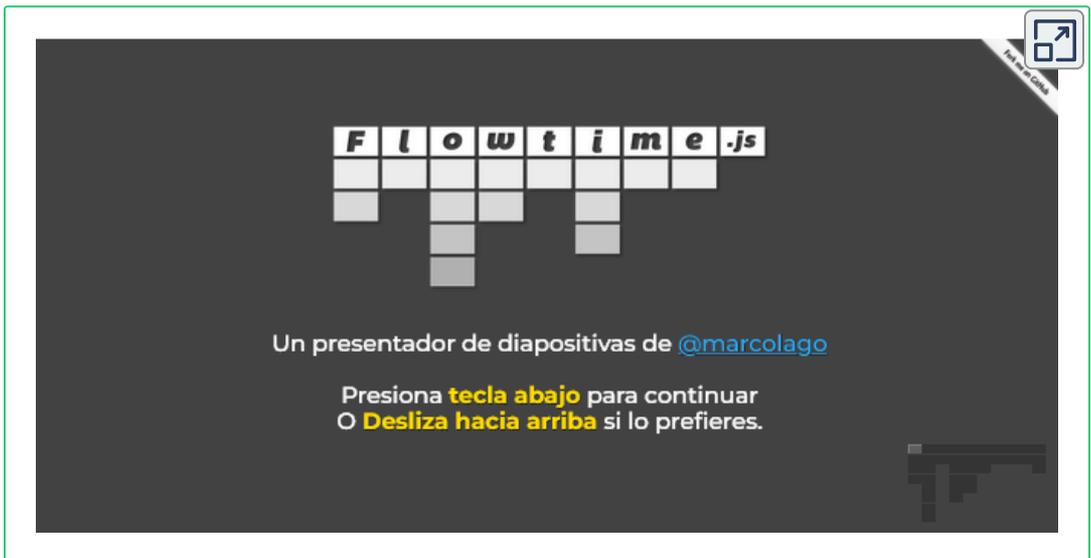
Flowtime.js fue diseñado por [Marco Lago](#), con última actualización en 2014 y la siguiente licencia:

Copyright (c) 2012 - Marco Lago. Por la presente, se otorga permiso, sin cargo, a cualquier persona que obtenga una copia de este software y los archivos de documentación asociados (el "Software"), para negociar el Software sin restricciones, incluidos, entre otros, los derechos de uso, copia, modificación, fusión, publicar, distribuir, sublicenciar y/o vender copias del Software, y permitir que las personas a quienes se les proporcione el Software lo hagan, sujeto a la siguiente condición: El aviso de copyright anterior y este aviso de permiso se incluirán en todas las copias o partes sustanciales del Software.

**Controles de navegación** . Puedes navegar por las páginas a través de enlaces, teclado y mouse. La navegación del teclado se basa en las teclas de flecha con la tecla Mayúsculas como modificador para saltar sobre fragmentos o secciones. Esta es la lista de comandos:

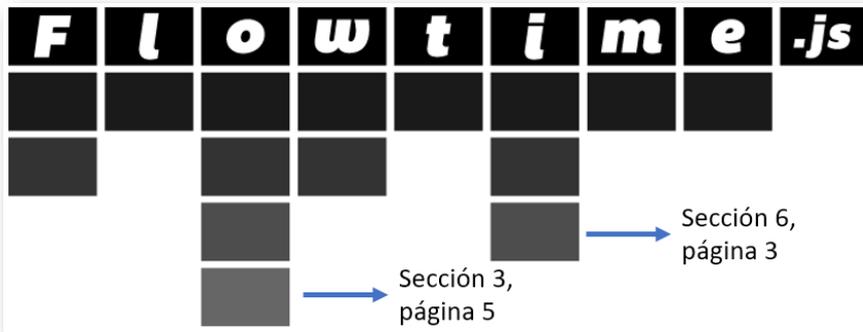
- Flechas hacia abajo o hacia arriba : navega a la página siguiente o anterior. Esta es la acción principal de navegación; todo el contenido es navegable usando solo estas teclas. Si hay fragmentos en la página, cada entrada muestra u oculta un fragmento.
- Shift + Flechas arriba o abajo : navega a la página siguiente o anterior saltando todos los fragmentos.
- Flechas izquierda o derecha : navega a la sección anterior o siguiente.

- Shift + Flechas izquierda o derecha : navega a la primera página de la sección anterior o siguiente.
- Re Pág : navega a la primera página de la sección actual.
- Av Pág : navega a la última página de la sección actual.
- Inicio : navega a la primera página de la presentación.
- Fin : navega a la última página de la presentación.
- ESC : modo de vista general (muestra todas las diapositivas).



El ejemplo es igual al original de Marco Lago, sólo hemos hecho tres cambios. El primero, es la traducción al español de las dos primeras diapositivas (sección 1), el segundo es la incorporación de un objeto interactivo (sección 5, página 5) y, el tercero, es la inclusión de la sección 10 con tres páginas, las dos últimas con vídeos de YouTube y Vimeo.

Este último cambio nos ayudará para comprender el diseño de Flowtime.js. Cada columna es una sección que, generalmente, trata sobre un mismo tema; por ejemplo, la sección 10 (columna 10) trata el tema de vídeos, a través de tres páginas. Observa el siguiente ejemplo:

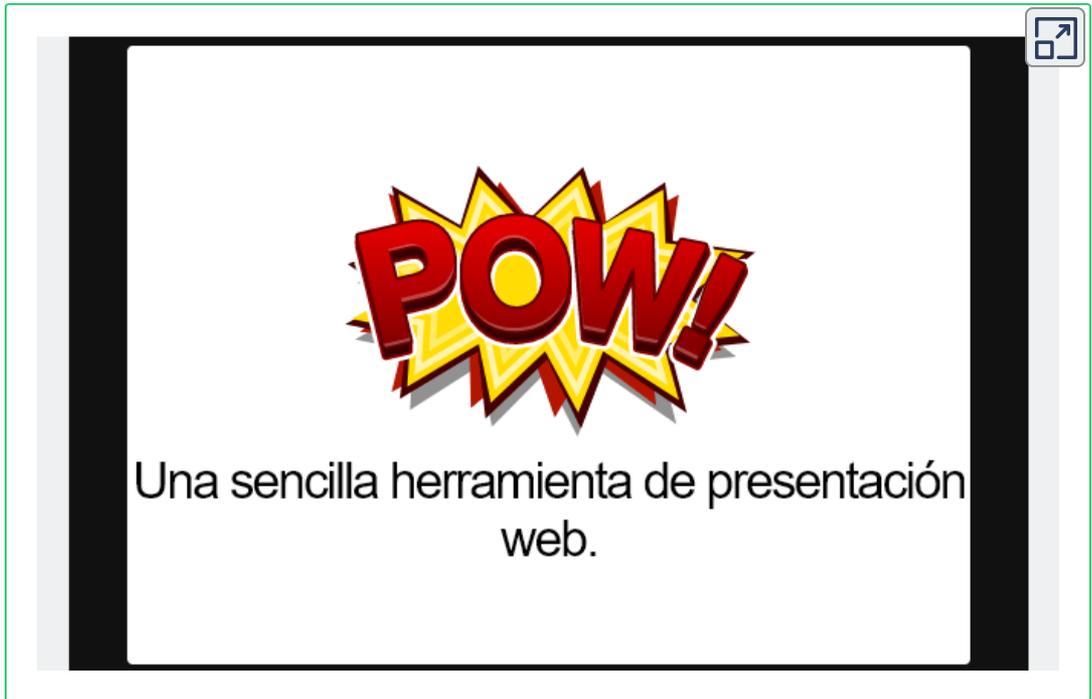


En el archivo index.html puedes observar como se encapsulan las diapositivas (páginas), las cuales deben ir en una caja (div), identificada con el número de sección (class="ft-section" data-id="section-10"), la página va en otra caja (div) identificada por el número de página:

```
<!-- Sección 10 (3 páginas) -->
<div class="ft-section" data-id="section-10">
  <div id="/section-10/page-1" class="ft-page"
  data-id="page-1">
    <h1>Demo de incrustación de video</h1>
    <p>Esta presentación es una prueba para
    incrustar videos de You Tube y Vimeo.</p>
  </div>
```

El diseño de la diapositiva es similar a las de los presentadores anteriores.

## 2.9 Presentador Pow

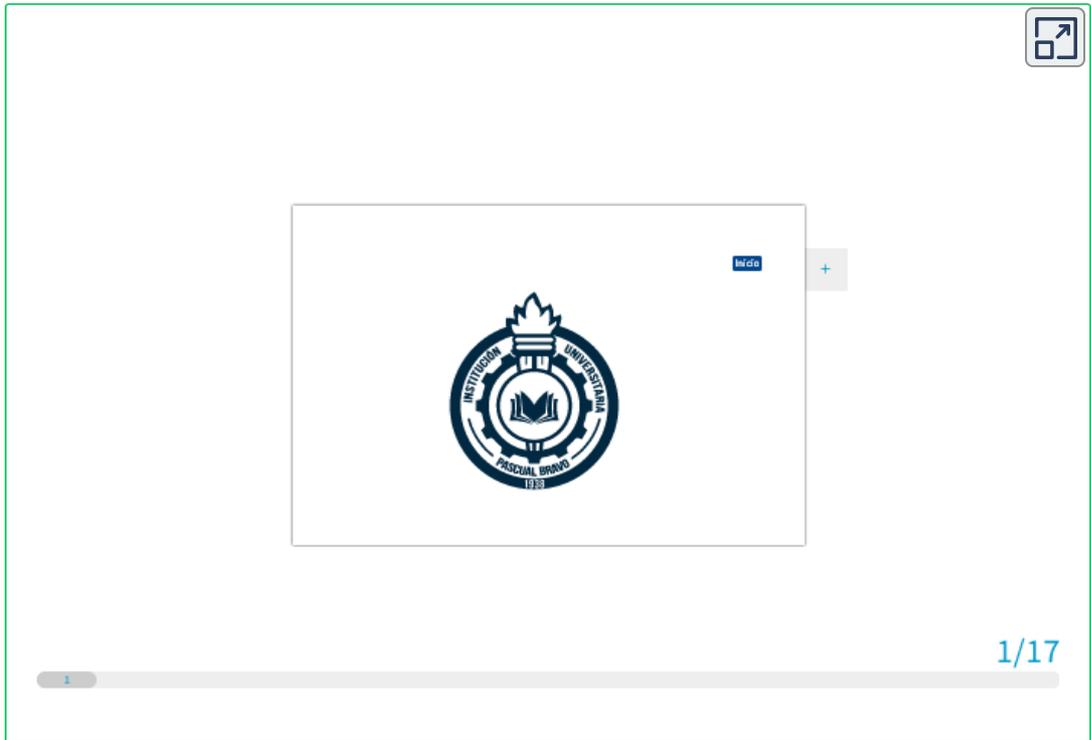


Haz clic en un lado de la diapositiva, o usa las teclas de desplazamiento, la barra espaciadora o, si lo prefieres en los botones de navegación que aparecen en la parte inferior.

Este presentador, diseñado por [Max Goodhart](#), en 2011, es muy similar al Google I/O Slides, incluyendo un código de animación en la primera diapositiva.

Las diapositivas están encapsuladas en una etiqueta `<div>` y su diseño es similar a lo que hemos explicado en las presentaciones anteriores.

## 2.10 Presenteer.js



Haz clic en un lado en el botón "+" y luego en Inicio, usa las teclas de desplazamiento.

Este presentador, diseñado por [Willem Mulder](#), con última actualización en 2018, utiliza transformaciones y transiciones CSS3. Está Licenciado bajo [Creative Commons BY-SA 3.0](#).

Las diapositivas están encapsuladas en una etiqueta `<div>`, su diseño es sencilla. Hemos agregado un vídeo y un objeto interactivo a la presentación original y la traducción de las diapositivas del holandés al español.

## 2.11 Shower Presentation Engine



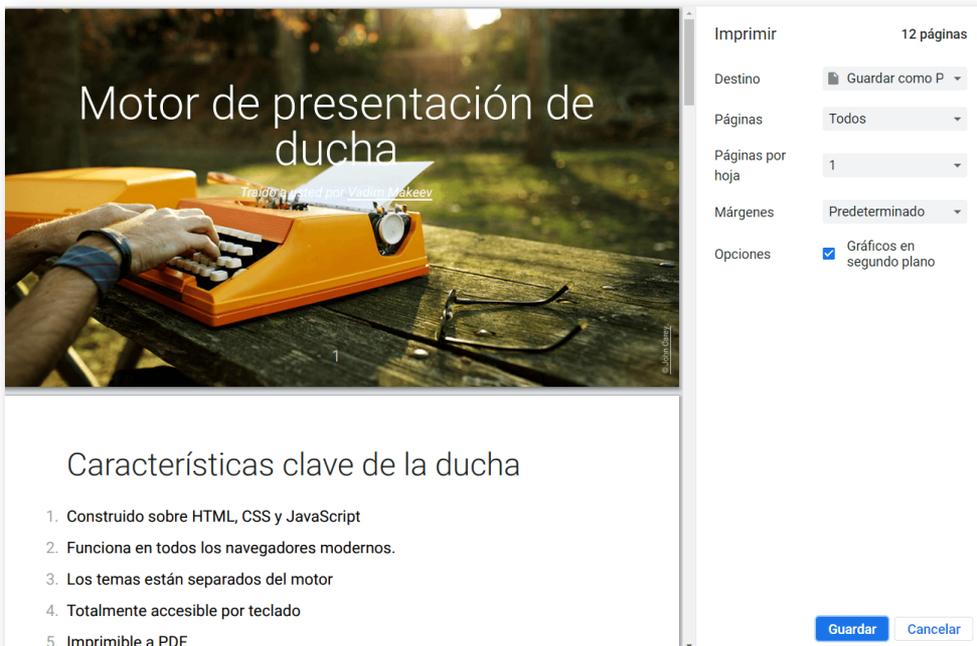
Shower (ducha) es un motor de presentación HTML5 creado por [Vadim Makeev](#) (Opera Software) y con actualización a 2020.

Parecido a la apariencia de PowerPoint, Shower intenta recrear presentaciones clásicas para imprimir. Shower también proporciona un medio para que los creadores de diapositivas implementen y alojen sus presentaciones usando Netlify. Esto activa automáticamente una reconstrucción cuando se actualiza Shower y no es necesario instalar herramientas adicionales para organizar la presentación.

La herramienta viene incluida con un "archivo template" (plantilla), que contiene los dos temas principales, Ribbon (Cinta) y Material, y los archivos principales ([Hipp, 2019](#)).

Una de las características interesantes es la posibilidad de exportar la presentación a PDF para cargarla en un servicio como SlideShare o Notist.

Lo hemos probado en el navegador Edge con el estilo Ribbon y en Chrome con estilo "material", tal como se aprecia en las siguientes imágenes.



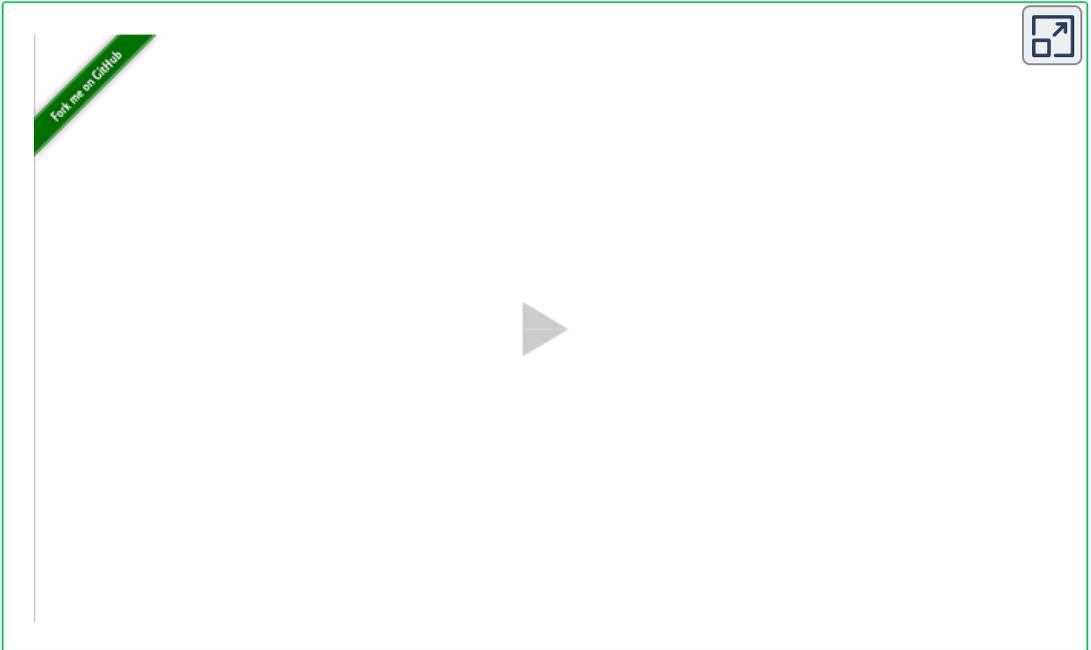
He aquí el presentador, en el que hemos incluido un objeto interactivo (diapositiva 12) y un vídeo (diapositiva 13). Las diapositivas están encapsuladas en etiquetas `<section>` y su edición es sencilla. En algunas diapositivas (ver la primera) se cambian los estilos para un mejor diseño. La navegación se hace a través de las flechas del teclado, barra espaciadora y teclas inicio, repág, avpág y fin. Se puede regresar al mapa de diapositivas con la tecla "esc".

Haz clic sobre la imagen, para abrir el presentador en una ventana emergente.



Si no deseas que aparezca el número de la diapositiva, incluye "clear" así: `<section clear>`, o para un fondo negro: `<section black>`; obviamente, debes incluir un estilo para cambiar los colores del texto. Otras características las puedes consultar en <https://github.com/shower/>.

## 2.12 Slider.css



Haz clic en el centro y luego navega con comodidad.

Slider.css, diseñado por [MZhou](#), con licencia MIT<sup>9</sup>, es otro presentador que no incluye JavaScript.

---

<sup>9</sup> Además del Copyright (c), se indica: "Por la presente se otorga permiso, sin cargo, a cualquier persona que obtenga una copia de este software y los archivos de documentación asociados (el "Software"), para tratar el Software sin restricciones, incluidas sin limitación los derechos de uso, copia, modificación, fusión, publicación, distribuir, sublicenciar y / o vender copias del Software, y para permitir a las personas a quienes se les proporciona el Software que lo hagan, sujeto a las siguientes condiciones: El aviso de copyright anterior y este aviso de permiso serán incluidos en todas las copias o partes sustanciales del Software.

El diseño es similar a las diapositivas anteriores, con algunas características especiales que se pueden incluir, como el zoom o la rotación (observa la presentación). Las diapositivas están encapsuladas en una etiqueta `<section>`. Algo diferente, asociado a los botones de avance y retroceso, es la necesidad de indicar al final de cada diapositiva, el número de la diapositiva anterior y/o siguiente; por ejemplo, para la diapositiva 10:

```

<!-- Diapositiva 10 -->
<section id="slider-10" default>
  <div class="center">
    <h1 style="font-size:55px">Una imagen
    </br></br>
    
  </div>
</section>

<div>
  <a href="#slider-9"></a>
  <a href="#slider-11"></a>
  <span></span>
</div>

```

La declaración de las diapositivas, para los botones de avance o retroceso, se hace en una etiqueta `<div>`.

Abre el archivo `index.html` y diseña tu propia presentación.

## 2.13 Weblides



Ahora, entramos en lo que podríamos llamar... ¡las liga de mayores!

El presentador lo puedes ver, admirar y disfrutar en pantalla ampliada, así que haz clic en la esquina superior izquierda y ¡sorpréndete"

De la wiki de [José Luis Antúnez, Antonio Laguna y Luis Sacristán](#), creadores de Weblides, sacamos algunos comentarios:

Con WebSlides se trata de contar la historia y compartirla de una manera hermosa. Hipertexto, código limpio y belleza como elementos narrativos. Esto se trata de buen karma y productividad.

Puedes crear presentaciones y artículos largos al instante. Solo se requiere de un conocimiento básico de HTML y CSS. Los diseñadores, desarrolladores, comercializadores y periodistas ahora pueden centrarse en el contenido.

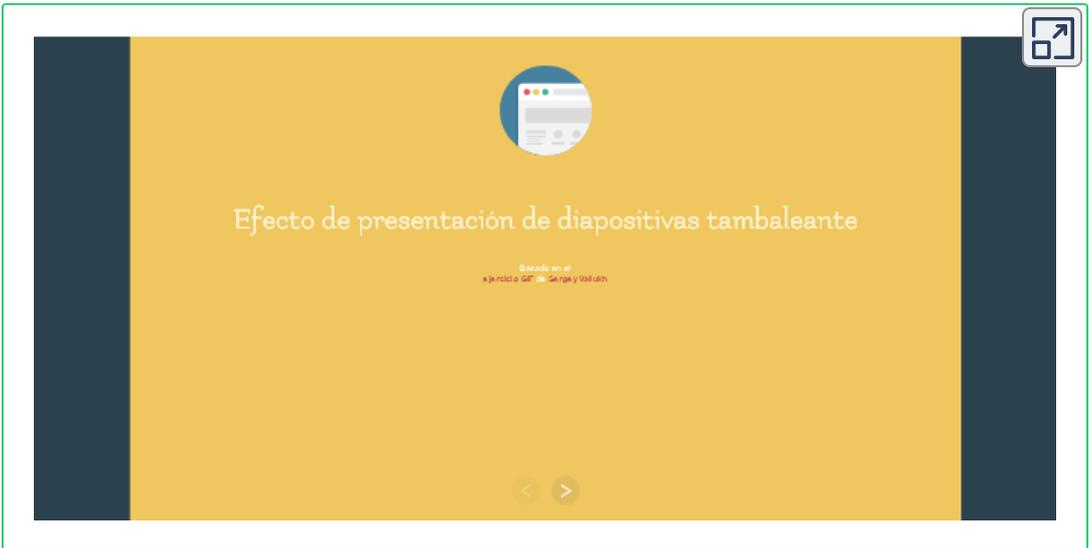
Todos estamos cansados de los pesados frameworks CSS. WebSlides es un punto de partida que proporciona componentes estructurales básicos y una arquitectura CSS escalable.

WebSlides se ofrece bajo licencia MIT, advirtiéndote que es una licencia permisiva corta y simple con condiciones que solo requieren la preservación de los derechos de autor y avisos de licencia.

La navegación se puede hacer con las teclas de desplazamiento o haciendo clic en las flechas que aparecen en la parte inferior del presentador.

Hemos dejado los demos para que explores todas las posibilidades de diseño que ofrece WebSlides; sin embargo, en la presentación que hemos intervenido, seleccionamos 14 diapositivas y agregamos una simulación de Física (Phet). Podrás observar las impresionantes diapositivas, que incluyen vídeos de pantalla completa, atenuados para resaltar los textos, la combinación de imágenes y textos y una gran cantidad de posibilidades que ofrece el presentador. Abre y explora el archivo `index.html` y comprenderás su diseño, pues con lo visto en el capítulo I y en las presentaciones anteriores, estarás en capacidad de hacerlo.

## 2.14 Wobbly Slideshow Effect



Este presentador, diseñado por [Mary Lou](#) con licencia MIT, se destaca por su efecto tambaleante. Sólo hemos dejado cuatro diapositivas, para probar el funcionamiento con vídeos y objetos interactivos. Su edición es sencilla, como podrás observar al abrir el archivo `index.html`.

Habrás observado que algunas presentaciones, en el libro, se inician con el editor DescartesJS. Esto obedece a que no son adaptables, por lo que tuvimos que ayudarles con DescartesJS; no obstante, todas se ven bien en pantalla completa.

A continuación, compartimos algunos presentadores, sin mayor explicación, pero que seguramente podrás intervenirlos con lo aprendido hasta aquí.

## 2.15 Stack.js

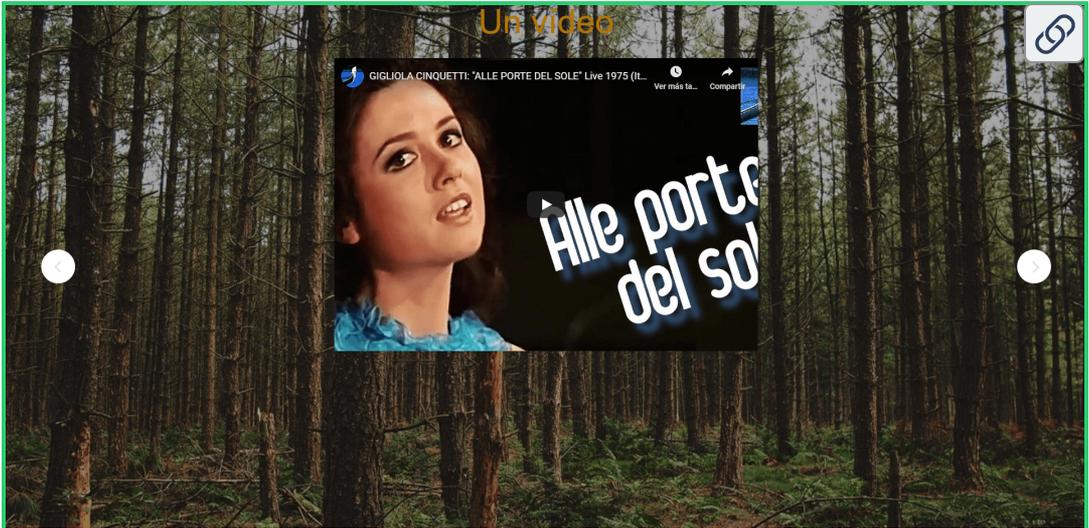


Stack, diseñado por Michael Bostock en 2012, adopta un enfoque único para las presentaciones de diapositivas al apilarlas de arriba a abajo, en lugar de las transiciones usuales de izquierda a derecha. Hemos incluido el vídeo "Balada triste de trompeta" de Rafael" y un objeto interactivo del Proyecto Canals del área de Estadística.

**Haz clic sobre la imagen para ver la presentación en pantalla completa.**

Es importante aclarar que las presentaciones son más útiles en ordenadores de mesa o portátiles, pues nuestro propósito es **presentar** a nuestro público algún tema, portafolio o conferencia y, obviamente, desde un ordenador es más funcional; si bien son compatibles con dispositivos móviles, algunas de las presentaciones, como lo dijimos antes, no son adaptables.

## 2.16 Clip-Path Revealing Slider

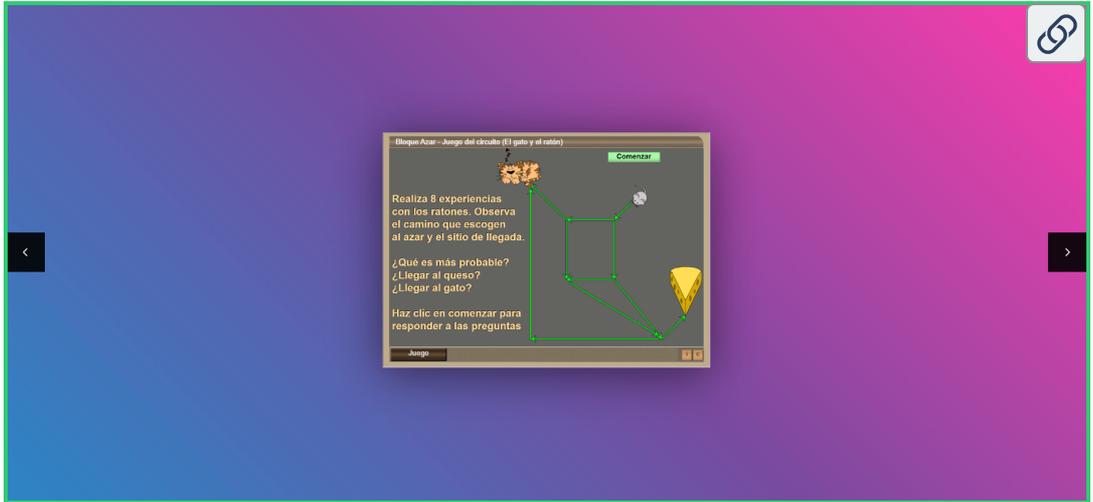


Control deslizante revelador de Clip-Path con HTML, CSS y JavaScript, diseñado por Nikolay Talanov en 2016. A diferencia de los demás, no hay desplazamiento de la diapositiva, como su nombre lo indica hay un efecto de animación que "revela" la diapositiva siguiente con un aparente desplazamiento (yo diría ocultamiento) de la diapositiva actual. Hemos incluido un vídeo de Gigliola Cinquetti y un objeto interactivo del Proyecto Canals del área de Estadística.

**Haz clic sobre la imagen para ver la presentación en pantalla completa.**

La navegación se realiza haciendo clic en las flechas de los extremos.

## 2.17 Motion blur effect using SVG filters

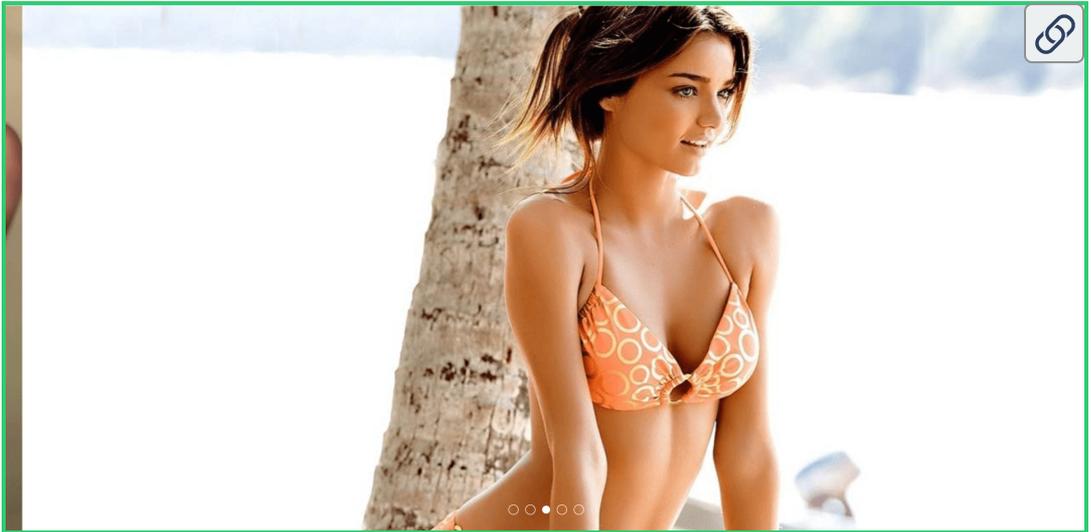


El Efecto de desenfoque de movimiento con filtros SVG es un experimento que simula un efecto de desenfoque de movimiento cada vez que se cambia una diapositiva. Aprovecha el filtro SVG Gaussian Blur y algunas animaciones de fotogramas clave CSS. La presentación fue diseñada por Damián Muti en 2017. Hemos incluido un vídeo de humor y un objeto interactivo del Proyecto Canals del área de Estadística.

**Haz clic sobre la imagen para ver la presentación en pantalla completa.**

La navegación se realiza haciendo clic en las flechas de los extremos. Observa que el efecto también se produce en la animación y en el interactivo (haz varias pasadas para verificarlo).

## 2.18 Preview slider



Control deslizante GSAP + Slick con vista previa de diapositivas anteriores/siguientes. Fue diseñado por Karlo Videk en 2016. Hemos incluido un vídeo de humor y un objeto interactivo del Proyecto Canals del área de Estadística.

**Haz clic sobre la imagen para ver la presentación en pantalla completa.**

El demo, que hemos descargado, está dedicado a presentar fotos de mujeres hermosas, las cuales no hemos eliminado, no por respeto al autor, sino porque sería un sacrilegio.

Bueno, hemos terminado este capítulo con 19 presentadores de diapositivas, seguro alguno será de tu gusto ¿no?... entonces, prepárate para las presentaciones en 3D.



# Capítulo III

## Diapositivas en 3D



## 3.1 Introducción

Iniciamos este capítulo con un hilo de comentarios acerca del uso de presentaciones de diapositivas alternativas, publicadas en <https://www.emezeta.com>:

Ahora, vamos a hacer una presentación de 10 laminas, algo básico, en PowerPoint vs cualquier herramienta que escojas... a ver cual se hace mas rápida y cuál puede usar el usuario común... ¿Cuántos usuarios comunes que usan Excel como un calculadora grande o PowerPoint como un Block de dibujo, pueden escribir código?... (XDuende)

Tiene razón XDuende, usar PowerPoint es mucho más cómodo y sencillo, pero exige que los amigos o estudiantes a los que les compartes la presentación hayan comprado, también, el Office u otra suite que lea el archivo. Aquí nos hemos ocupado de buscar alternativas libres, no sólo de pago, sino también de dependencias y restricciones.

No es una opción para cualquier tipo de usuario, pero los resultados de cada una tampoco son equivalentes. Muchas veces, el PowerPoint es suficiente, otras veces, no. :-). Como casi siempre, lo bueno es tener alternativas (Manz).

Conocía las maravillas en cuestión de efectos que se pueden hacer combinando CSS y JavaScript, pero nunca me había puesto a buscar librerías que permitiesen trabajar presentaciones. Me ha gustado mucho la primera, impress.js, con la que seguramente juegue un poco en los próximos días (Iván).

Plenamente de acuerdo con Manz y, como lo veremos en el cuarto capítulo, con las apreciaciones de Iván con respecto a Impress.js, simplemente... maravilloso.

Creo que no debemos entender este tipo de opciones como para el usuario común o para el experto. Si es cierto que normalmente la gente utiliza el MS PowerPoint, también es cierto que la gente suele utilizarlo de manera muy tosca y sin tener conocimiento de que hay un elenco de alternativas que se pueden utilizar para lo mismo (tales como LibreOffice y otras suites ofimáticas) o incluso para darle un toque de originalidad con respecto a lo tradicional en las presentaciones (como Prezi) o hacerlas aún más potentes trabajando con estas librerías (Iván).

Nuestro propósito es brindar alternativas sin costo y de fácil acceso para cualquier usuario; obviamente, se requiere de un mínimo de esfuerzo para comprender el funcionamiento de los presentadores de diapositivas que hemos compartido y las que, a continuación, presentaremos. Otra dependencia que procuramos evitar es la conectividad, como es el caso de Prezi que, como lo enunciamos en el prefacio, tiene como alternativa Impress.js.

Como en toda propuesta de cambio, hay detractores y defensores pero, si has llegado hasta acá, te consideramos un defensor de esta propuesta o, al menos, un interesado en conocer lo que la comunidad de desarrolladores y diseñadores nos ofrecen como alternativas al software propietario.

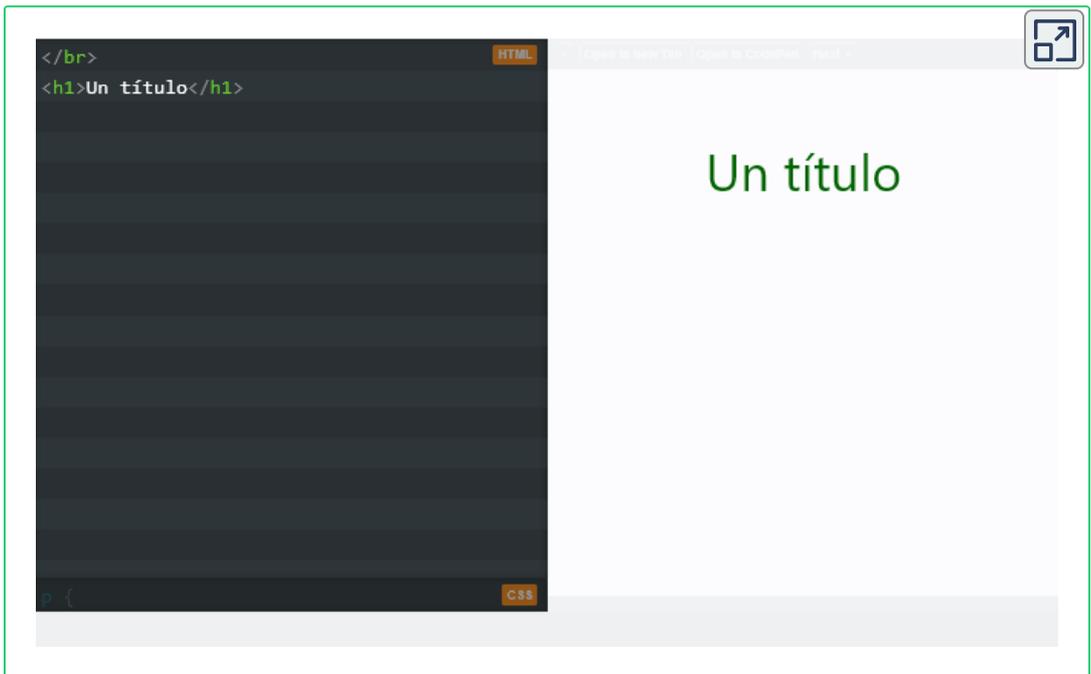
## 3.2 Un breve retorno al HTML

Seguramente, ya habrás practicado con algunos de los presentadores del capítulo anterior y, obviamente, habrás cambiado algunos estilos, tal como lo explicamos en el capítulo I. Sin embargo, es bueno que repases, nuevamente, cómo intervenir estos estilos. Para ello, observa el siguiente vídeo y, luego, practica con el interactivo de la siguiente página, tanto con el código HTML como con el de CSS.



He aquí el aplicativo en el cual puedes practicar con los estilos. Este aplicativo hace parte del presentador de diapositivas **Inspire.js**, diseñado por [Lea Verou](#). Este presentador creado en 2010 con el nombre CSSS, tuvo tanta aceptación que luego se relanza como Inspire.js en 2018.

Este presentador no lo hemos incluido en el capítulo anterior; sin embargo, puedes descargarlo en [GitHub](#) y explorarlo.



Escribe un texto que incluya etiquetas tipo título, párrafo y viñetas. Luego, con el botón CSS, cambia los estilos de estas etiquetas (colores, tamaños de letra, posición, etc.).

Iniciemos, entonces, con otras alternativas, ahora como diapositivas tipo 3D.

## 3.3 Slicebox



Slicebox es un complemento de control deslizante de imágenes en 3D jQuery que utiliza CSS 3D Transforms y proporciona una alternativa elegante para navegadores más antiguos que no admiten las nuevas propiedades. Fue diseñado por [Pedro Botelho](#) en 2011 con licencia Creative Commons.

Como lo hicimos con algunas presentaciones en 2D, aprovecharemos el editor DescartesJS para escalar las diapositivas.

## Apoyo en DescartesJS

El uso del editor DescartesJS obedece a que presentadores, como Slicebox, son adaptables únicamente para lo que fue diseñado, en este caso para imágenes. Como nuestro propósito es incluir vídeos y objetos interactivos, se hace necesario garantizar que la presentación sea adaptable, también, para estos elementos; por ello, encontrarás en la carpeta de Slicebox, un archivo llamado `descartes.html`, el cual no debes intervenir. Para crear tu presentación, sólo debes intervenir el archivo `index.html`, tal como lo hemos venido haciendo y, para presentarla, ejecutas el archivo `descartes.html`.

## Diseño en Slicebox

Slicebox viene con varias opciones para producir diferente efectos, de las cuales sólo nos ocuparemos de la orientación (ver línea 88 en el archivo `index.html`).

```
orientation : 'r',  
cuboidsRandom : true
```

La orientación puede ser **vertical**, **horizontal** o tipo **random** (aleatoria). Por defecto, viene con el tipo **random**.

Ahora, vamos al cuerpo del presentador (`<body>`), inicialmente con el título que aparece en la parte superior, recuerda que puedes modificar el color con un estilo en línea o en la etiqueta (en este caso `<h1>`).

```
<body>
  <div class="container">
    <h1>Medellín - Antioquia - Colombia</h1>
```



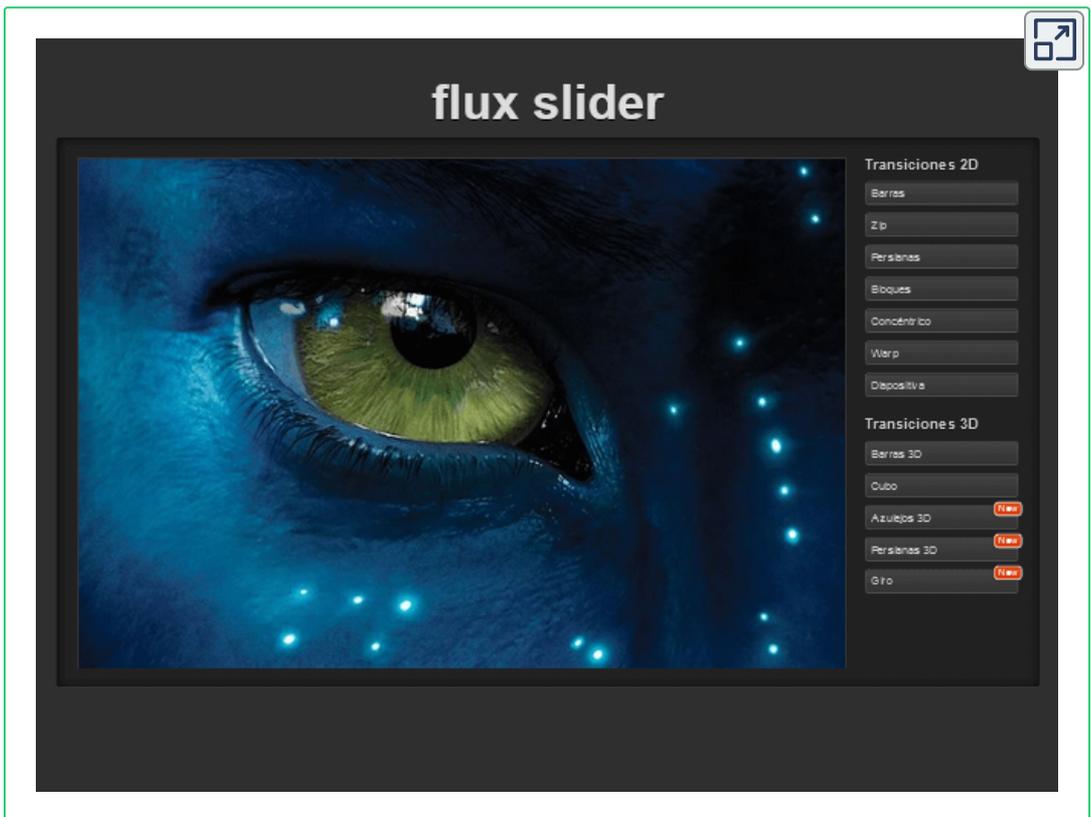
Las diapositivas están encapsuladas en etiquetas `<li>`. Observa que para la segunda hemos incluido un título, modificando su estilo; además, de un vídeo de YouTube. Si bien se muestra el vídeo y los objetos interactivos, los efectos especiales de partición vertical u horizontal no son aplicables, pues la presentación fue diseñada para imágenes.

```
<li>
  <a></a>
</li>

<li>
  <h1 style="color:red;
font-size:20px">Un vídeo de
YouTube</h1>
  <a><iframe width="730" height=
    "450" src=
    "http://www.youtube.com/embed/eq-B
XwVerFs?rel=0" frameborder="0"
    allowfullscreen></iframe></a>
</li>
```

En la quinta y novena diapositiva incluimos dos objetos interactivos. El primero es tomado del Proyecto Un\_100. El segundo es un rompecabezas circular. Al igual que en el vídeo de YouTube, las diapositivas no presentan efectos especiales como las persianas tipo 3D.

Es posible que encuentres otros presentadores con este tipo de animaciones; sin embargo, la mayoría tienen un diseño propio para imágenes, un ejemplo de ello es el Flux Slider, diseñado por [Joe Lambert](#):

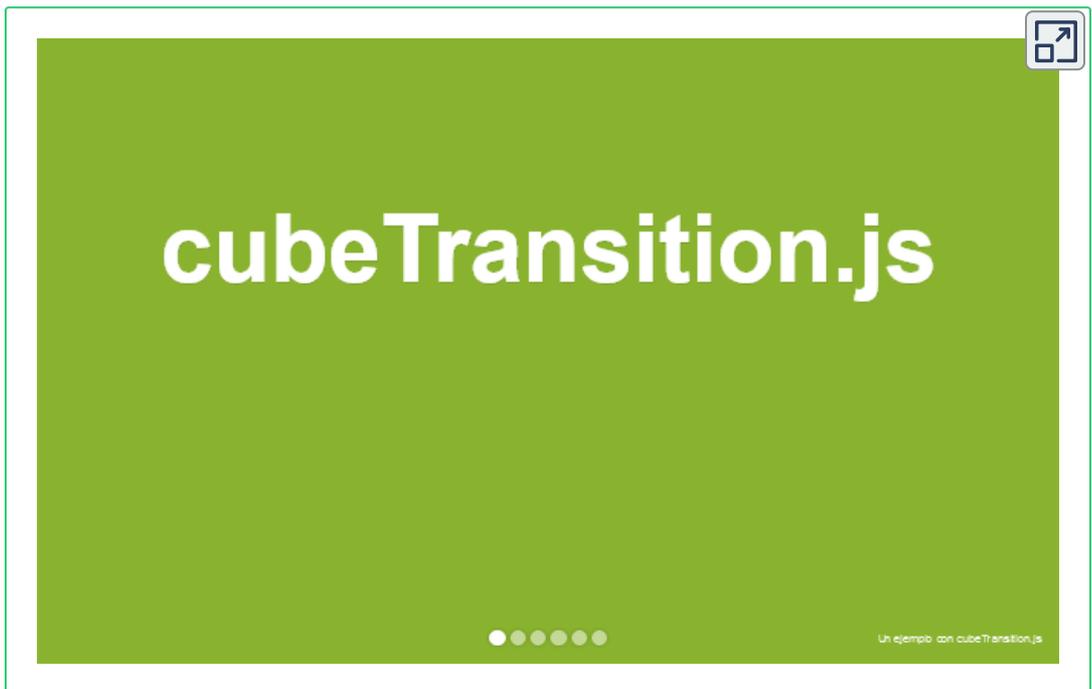


Haz clic en cualquiera de las transiciones del menú (2D o 3D) y observas los efectos producidos.

## 3.4 Efecto tipo cubo

Una transición muy usada es el efecto tipo cubo, el cual observamos en los dos ejemplos anteriores. En este apartado, describimos algunos presentadores de diapositivas que ofrecen esta transición.

### 3.4.1 cubeTransition.js



Haz clic en los círculos inferiores para pasar diapositivas o usa el ratón.

Hemos incluido en las diapositivas un vídeo y un objeto interactivo.

Este presentador fue diseñado por [2byte \(nickname en GitHub\)](#). El diseño básico encapsula las diapositivas en etiquetas <div>:

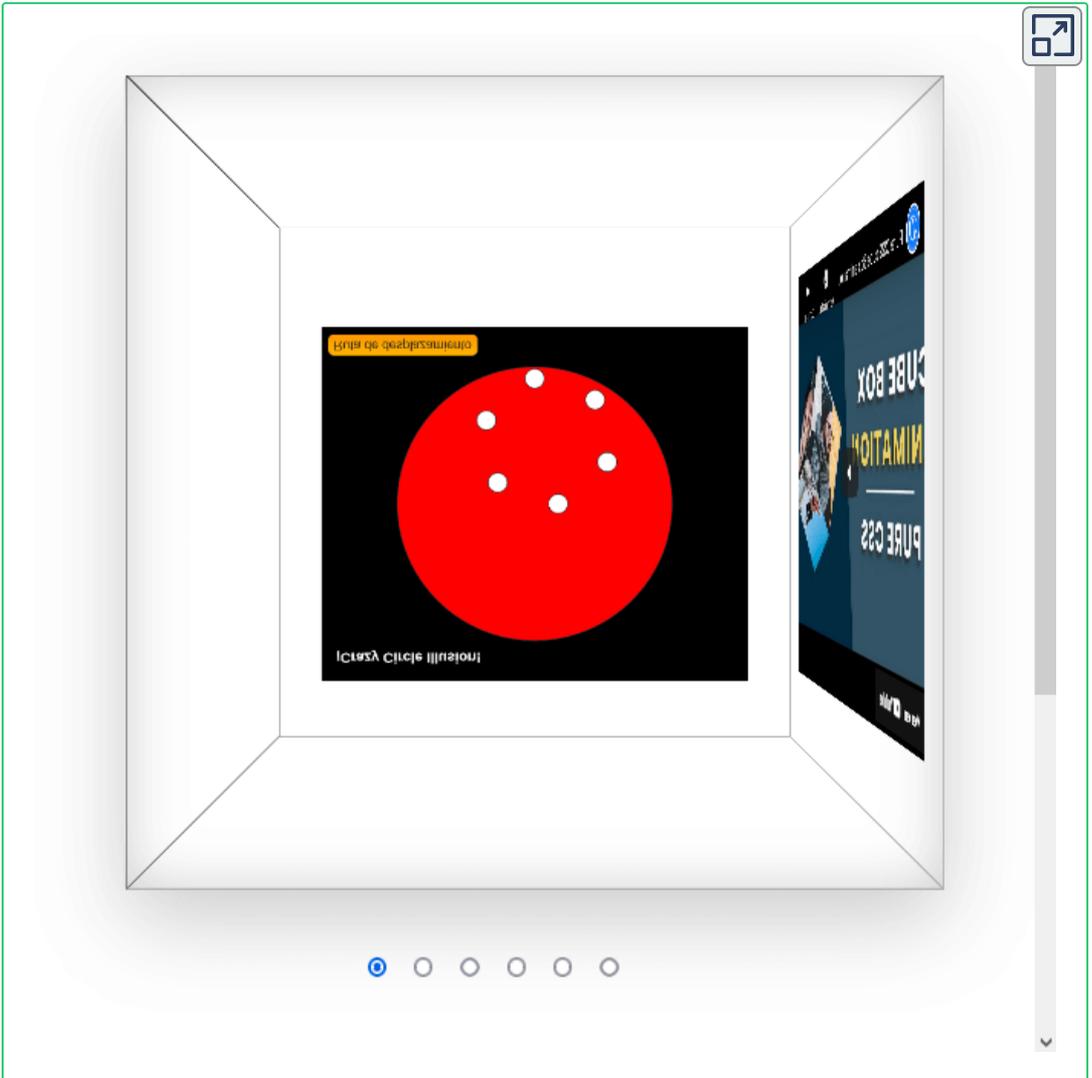
```
<body>
<div id="cubeTransition">
<div class="page1"><h2>cubeTransition.js</h2></div>
<div class="page2"><h2>Elegant,</h2></div>
<div class="page3"><h2>Exciting!</h2></div>
<div class="page4"><h2>Simple.</h2></div>
</div>
```

Para incluir diapositivas adicionales, es necesario intervenir, también, el archivo style.css que se encuentra en la carpeta **CSS**, por defecto son sólo cuatro diapositivas, para dos más...

```
60  □ .page4 {
61      background: green;
62  }
63
64  □ .page5 {
65      background: purple;
66  }
67
68  □ .page6 {
69      background: blue;
```

Observa el vídeo en la quinta diapositiva, el cual es un mini tutorial de este presentador tipo cubo.

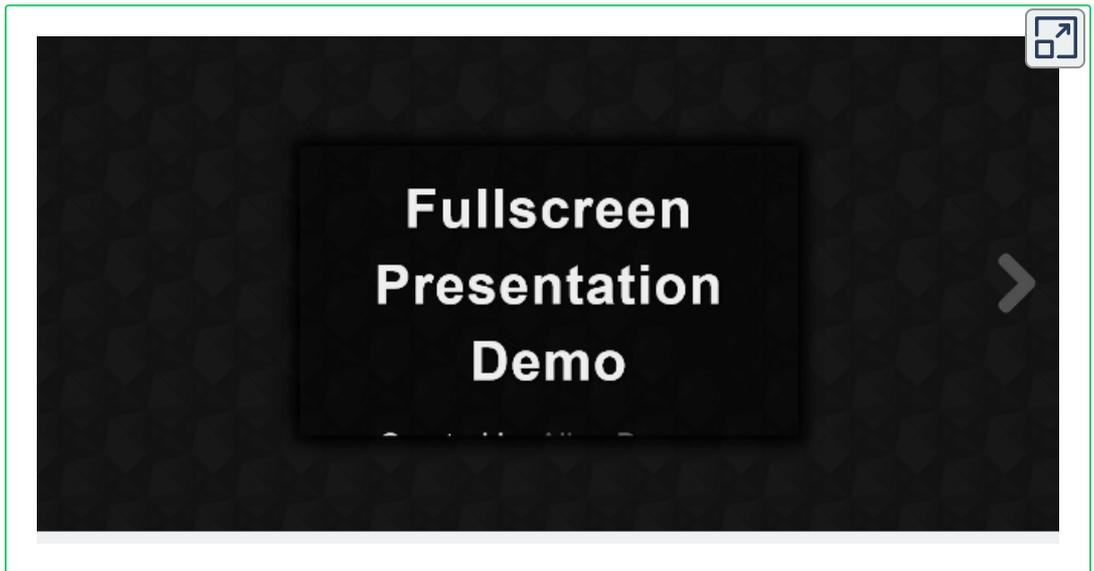
## 3.4.2 3D slider con puro CSS3



Este presentador, diseñado por [Forny Hucker](#) en 2020, es un cubo en sentido estricto, en tanto que sólo permite seis diapositivas, de acuerdo al número de caras del cubo.

Al igual que en las demás presentaciones, incluimos un vídeo y un objeto interactivo. El desplazamiento del cubo se puede hacer haciendo clic en los círculos inferiores o con las teclas de flecha derecha e izquierda. Las diapositivas están encapsuladas en la etiqueta `<figure>`.

### 3.4.3 jQuery Fullscreen Presentation



Incluimos esta presentador de [Alina Basova](#), como introducción al Reveal.js, que veremos más adelante. Alina incluye en un sólo archivo los códigos HTML, CSS y JavaScript.

En la última diapositiva puedes ver un vídeo sobre Reveal.js, pero... es mejor que esperes a que llegemos a él... que será pronto.

## 3.4.4 Adaptor

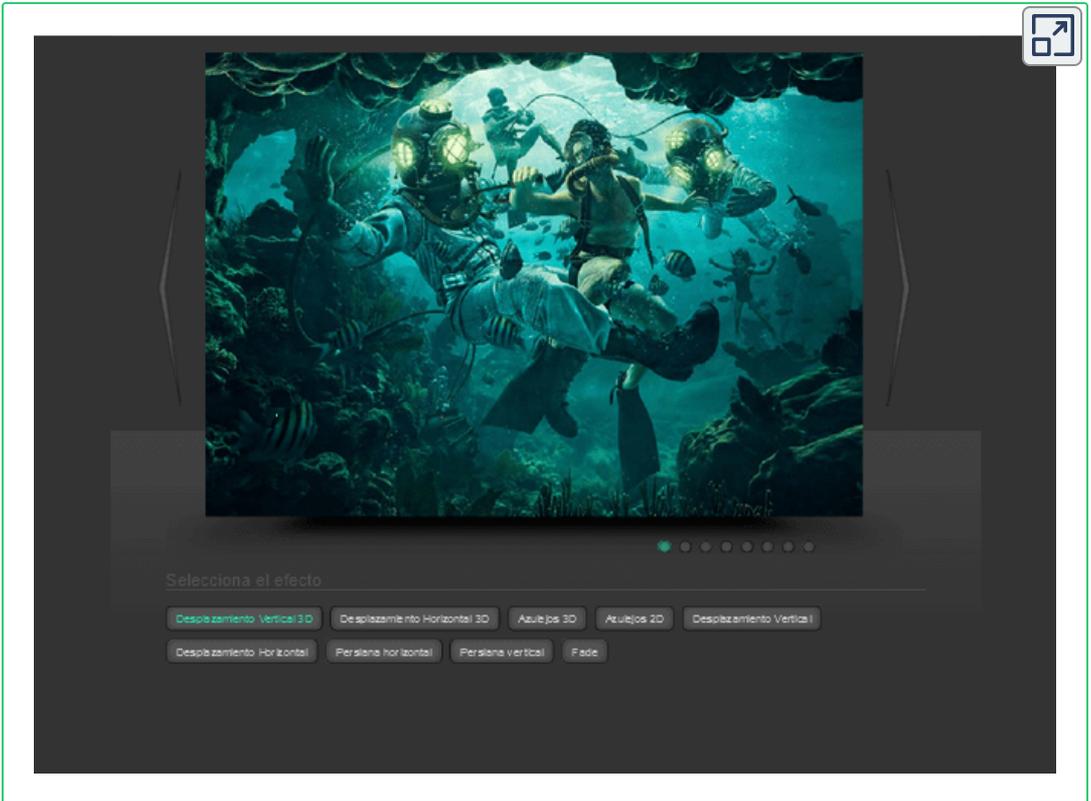
Adaptor tiene como objetivo crear geniales transiciones de animación de diapositivas en 2D o 3D, fue diseñado por [Phil Parsons](#) en 2012.

Las diapositivas, encapsuladas en una etiqueta <figure>, fueron diseñadas para albergar imágenes de un tamaño de 830×590 píxeles, tamaños diferentes pueden producir efectos indeseados. Hemos incluido una imagen de menor tamaño:

```
<figure class="slide">  
    
</figure>
```

Los efectos que se pueden generar, son los siguientes:

- Desplazamiento vertical 3D (tipo cubo)
- Desplazamiento horizontal 3D (tipo cubo)
- Azulejos 3D (losas o piezas de la imagen)
- Azulejos 2D
- Desplazamiento vertical 2D
- Desplazamiento horizontal 2D
- Persiana horizontal
- Persiana vertical
- Desvanecimiento (fade)



El desplazamiento de las diapositivas se puede hacer haciendo clic en las flechas laterales o los botones circulares inferiores. Por cada diapositiva adicional (el vídeo y el objeto interactivo que adicionamos, por ejemplo), hay agregar botones adicionales en la etiqueta <footer>:

```
<li><a class="goto-slide current" href="#" data-slideindex="0"></a></li>
<li><a class="goto-slide" href="#" data-slideindex="1"></a></li>
<li><a class="goto-slide" href="#" data-slideindex="2"></a></li>
<li><a class="goto-slide" href="#" data-slideindex="3"></a></li>
<li><a class="goto-slide" href="#" data-slideindex="4"></a></li>
<li><a class="goto-slide" href="#" data-slideindex="5"></a></li>
<li><a class="goto-slide" href="#" data-slideindex="6"></a></li>
<li><a class="goto-slide" href="#" data-slideindex="7"></a></li>
```

## 3.5 Reveal



Este presentador, diseñado por [Hakim El Hattab](#), produce diapositivas con efectos 3D. Presentaremos dos versiones, la primera diseñada en 2013 y la segunda en 2020.

El desplazamiento, en ambas versiones, se realiza con la barra espaciadora, las teclas de dirección o haciendo clic en las flechas que aparecen en la esquina inferior derecha, también funciona con un control remoto de diapositivas. Es importante tener en cuenta que cuando estamos interactuando con una escena, una página web o estamos reproduciendo un vídeo, debemos hacer clic fuera de la diapositiva para que los controles de desplazamiento se vuelvan a activar. La ventaja adicional de este tipo de presentación son sus transiciones, algunas tipo 3D.

Iniciamos nuestra explicación con el Reveal 2013; posteriormente, lo que diseñemos lo trasladamos a Reveal2020 y aclaramos los cambios significativos en la nueva versión.

## 3.5.1 Reveal 2013



### Diseño en Reveal

**Paso 1.** Abrimos el archivo index.html de la carpeta Reveal con un editor de textos sin formato.

**Paso 2. Modificación del diseño y la transición.** Reveal presenta ocho tipos de diseño (temas): beige, default, moon, night, serif, simple, sky y solarized. Cada diseño cambia colores de textos, botones y fondos. Por otra parte, hay ocho tipos de transiciones: default, cube, page, concave, zoom, linear, fade y none. La transición linear convierte el presentador al estilo de desplazamiento horizontal.

Tanto los estilos como las transiciones, las podemos modificar en las líneas de código que se encuentran al final que, inicialmente, presenta un tema por defecto (default) y una transición tipo cubo (cube):

```
theme: Reveal.getQueryHash().theme ||  
'default', // temas disponibles en la  
carpeta /css/theme  
transition: Reveal.getQueryHash().  
transition || 'cube', //  
default/cube/page/concave/zoom/linear/fade  
/none
```

Estos son imágenes que corresponden a algunos temas (sugerimos practicar con varias opciones, hasta encontrar aquella que se ajuste a tu gusto):



**Paso 3. Modificación o inclusión de diapositivas.** En este presentador, cada diapositiva se encuentra encapsulada en una etiqueta <section>. Estas diapositivas se encuentran al inicio del <body>, el ejemplo que presentamos tiene 12 diapositivas, que explicamos así:

## Primera diapositiva. Incluye solo texto

```
<section>
  <h1>Mi Medellín</h1>
  <h3>Una pequeña muestra</h3>
  <p>y otras diapositivas</p>
</section>
```

Cada diseño determina tipos de fuentes, posición, color y tamaños. En el diseño que hemos escogido (default), la etiqueta `<h1>` escribe un texto grande centrado en la parte superior de la diapositiva. La etiqueta `<h3>`, de una tercera parte de `<h1>`, y la etiqueta párrafo `<p>`, mucho más pequeño. Si deseamos incluir otras diapositivas de este tipo, sólo tenemos que copiar este bloque e insertarlo donde queramos que se presente.

**Diapositivas con imágenes.** Hemos incluido seis imágenes, combinando tipos de letra y colores de fondo.

Imagen con fondo dado por el diseño. Observemos que es necesario definir el tamaño de la imagen:

```
<section>
  <h2>Feria de las flores</h2>
  <h3>Se llena de color Medellín</h3>
  <img width="593" height="445" src=
    "imagenes/foto2.jpg"/>
</section>
```

Imagen con fondo blanco (white, #FFF o #FFFFFF), se usa el comando data-background:

```
<section data-background="#FFF">
  <h1>El metrocable</h1>
  <img width="640" height="427" src
    ="imagenes/foto1.jpg">
</section>
```

Imagen con fondo blanco y texto rojo:

```
<section data-background="#FFF">
  <h2><font color="red">Biblioteca
  España</font></h2>
  <img width="640" height="451" src
    ="imagenes/foto3.jpg">
</section>
```

Así lo diseñamos hace seis años, aunque lo correcto sería:

```
<section data-background="#FFF">
  <h2 style="color:red">Biblioteca
  España</h2>
  <img width="640" height="451" src
    ="imagenes/foto3.jpg">
</section>
```

Imagen con fuente externa. No se incluyó el tamaño, lo captura de la página.

La quinta y sexta diapositiva se encapsulan, a su vez, en otra etiqueta <section>, lo que genera un desplazamiento vertical, algo así como las secciones de flowtime.js:

```
<section>
  <section data-background="#0077FF">
    <h2><font color="yellow">Nuestro Botero
    </font></h2>
    <img src=
    "https://i.pinimg.com/originals/63/e9/a2/63
    e9a2b5f68de3bafbb2cfe362d46631.jpg">
  </section>
  <section data-background="#0077FF">
    <img src=
    "https://static.arteinformado.com/docum
    entos/eventos/06/61044/Obra_de_Fernando
    _Botero.jpg">
  </section>
</section>
```

La séptima diapositiva, otra imagen, combina un fondo verde con un texto amarillo (te invitamos a corregir el estilo para el color del texto).

```
<section>
<h2><font color="yellow">Una escena Descartes
</font></h2>
  <iframe src=
  "https://proyectodescartes.org/Telesecundar
  ia/materiales_didacticos/2f_b01_t01_s01-JS/
  index.html" width= "790" height= "565"
  ></iframe>
</section>
```

**Un objeto interactivo.** En este caso, se trata de un objeto del Proyecto Telesecundaria llamado "Escuchando el movimiento". Es importante ensayar el tamaño de la escena con varios valores hasta ajustarla.

```
<section>
<h2 style="color: yellow">Una escena de
Descartes</h2>
  <iframe src=
    "https://proyectodescartes.org/Telesecundaria/materiales_didacticos/2f_b01_t01_s01-JS/index.html" width= "790" height= "565"
  ></iframe>
</section>
```

**Un vídeo de YouTube.** Tal como lo hemos hecho en todos nuestros presentadores, la novena diapositiva es un vídeo. Observa que le incluimos el parámetro **vq=hd1080**, para forzarlo a que lo muestre en alta calidad de 1080 (HD).

```
<section>
<h2 style="color: green">Un vídeo</h2>
  <iframe src=
    "https://www.youtube.com/embed/awmg14u3Px0?vq=hd1080" width= "960" height= "480"
  ></iframe>
</section>
```

Las últimas tres diapositivas se presentan en sentido vertical (sección) y corresponden a las originales del autor.

## 3.5.2 Reveal 2020



En esta nueva versión de Reveal.js, Hakim El Hattab nos ofrece un cambio significativo que, frente al PowerPoint, lo convierte en una de las mejores alternativas. En la versión anterior, teníamos la posibilidad de cambiar el tema de las transiciones; sin embargo, el tema elegido era único para todas las diapositivas. Si elegíamos "cubo" (cube), el desplazamiento en toda la presentación tenía el efecto de la cara de un cubo pero, para presentaciones extensas, el atractivo inicial desaparece.

En la nueva versión, el estilo de la transición se define para cada diapositiva, como lo veremos a continuación.

Son seis **los estilos de transición** que trae la nueva versión: fade, zoom, ninguno, deslizar, convexo y cóncavo, donde el zoom es una de la novedades, tal como lo observas en la segunda diapositiva:

```
<section data-background="imagenes/foto2.jpg">
  <h1>El metrocable</h1>
  <img width="640" height="427" src=
    "imagenes/foto1.jpg">
</section>
```

Para hacer más impactante la transición entre la primera y la segunda diapositiva, en esta última hemos puesto como fondo (data-background), la imagen de la primera. También podrías usar gifs animados o vídeos como fondo.

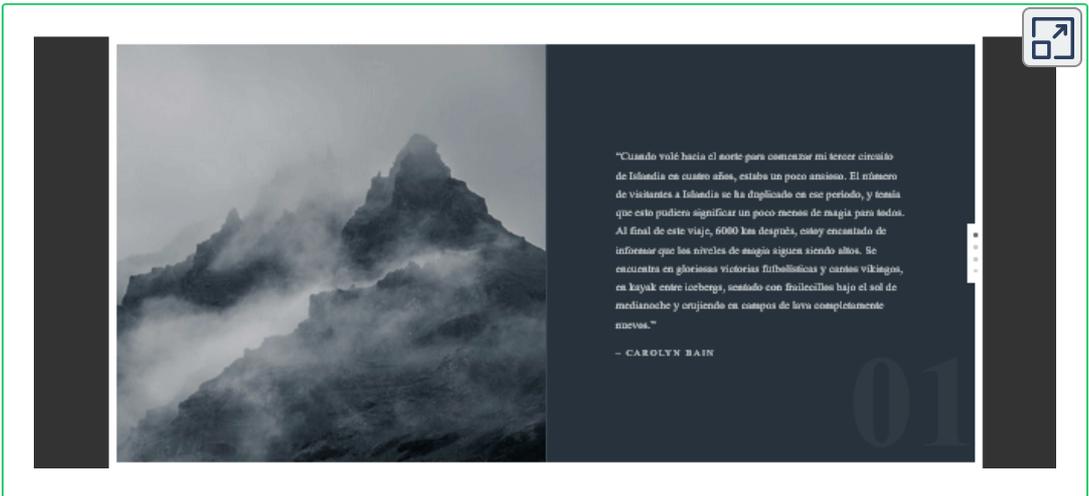
Las presentaciones se pueden **exportar a PDF**, para lo cual sólo tienes que abrir el archivo index.html en tu navegador y, una vez abierto, incluyes al final de la dirección: **?print-pdf**:



Esta acción te genera una cinta de diapositivas vertical. Luego, simplemente vas a la opción **imprimir** de tu navegador, cambiando la configuración de Destino a Guardar como PDF.

Bueno, solo falta que explores más el presentador y, finalmente, que eches a volar tu imaginación.

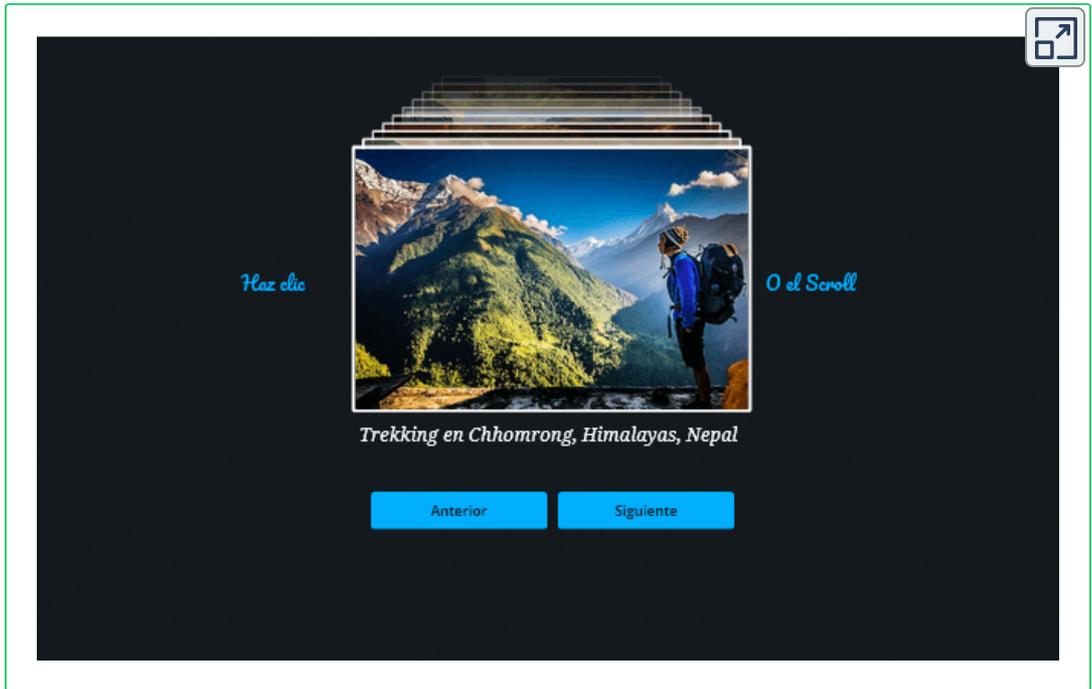
## 3.6 Split 3D Carousel



Un interesante presentador, diseñado por [Paul Noble](#), que realiza una transición de panel dividido en tres dimensiones. El movimiento de los paneles es vertical y en sentido contrario, realizado con la botonera lateral o con las flechas arriba y abajo del teclado. Te dejamos como tarea explorarlo. Inicialmente, observa la estructura de la primera diapositiva:

```
<div class="spinner__face js-active" data-bg="
"#27323c">
  <div class="content" data-type="iceland">
    <div class="content__left">
      </div>
    <div class="content__right">
      <div class="content__main">
        <p>"Cuando volé hacia el norte
        para comenzar..." </p>
        <p>- Carolyn Bain</p>
      </div>
      <h3 class="content__index">01</h3>
    </div>
  </div>
</div>
```

## 3.7 Flipping Gallery



Un complemento jQuery fácil y elegante que utiliza transiciones y transformaciones CSS3 para crear una galería 3D con una increíble animación de volteo y posibilidad de voltear las imágenes con clic o la rueda del ratón (<https://www.jqueryscript.net/>). El presentador fue creado por [Pete R.](#), fundador de BucketListly

Lo que quería que este plugin hiciera es simplemente tomar todas las etiquetas `<img>` en un contenedor y el plugin debería apilarlas, crear dimensiones y permitir que los usuarios interactúen de inmediato sin agregar más marcas. Logré crear ese complemento ideal, pero no fue fácil.

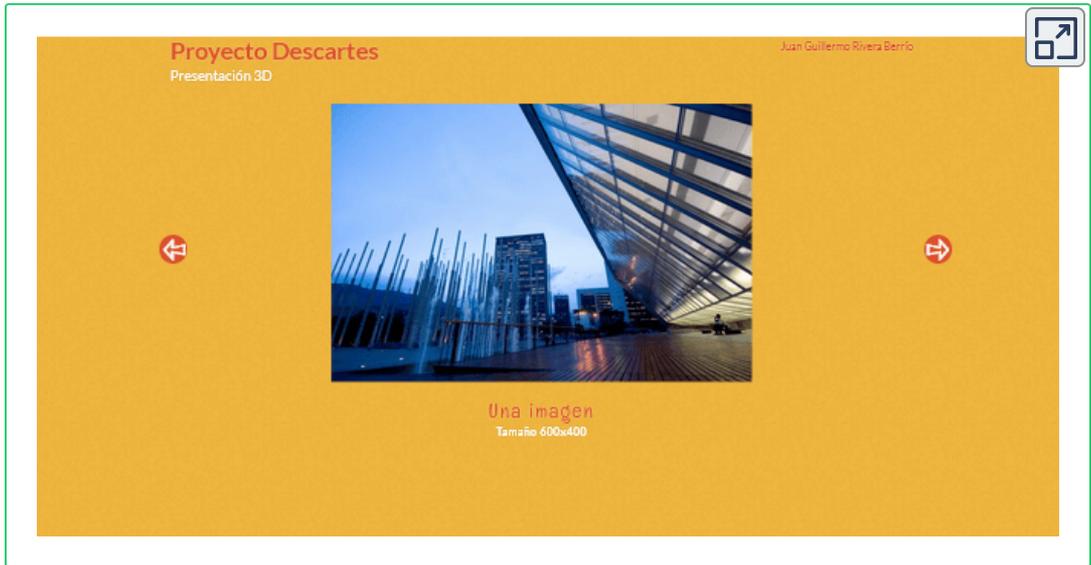
El primer problema que encontré fue que necesitaba una forma de agregar una dimensión a la galería mientras las apilaba en el orden correcto. Lo que tuve que hacer fue recorrer todas las fotos, determinar cuál estaba en la parte superior y luego calcular el índice z correcto, la opacidad y aplicarlas en cada foto individual. Luego los escalé con CSS3 transform para que creara una dimensión de fotos apiladas cada vez más en el sitio web ([Pete R.](#)).

Pese a que la galería fue diseñada para imágenes, la incorporación de vídeos y objetos interactivos se hizo sin mayor problema, siempre que los ajustemos al tamaño de la diapositiva, que es 500×333 pixeles y, cuando se muestre, pasarlos con los botones "siguiente" o "atrás". La intervención es sencilla, al final del archivo index, pones el texto (data-caption) y luego la imagen o el objeto:

```
259 <div class="gallery">
260   <a href="#" data-caption="Trekking en Chhomrong,
    Himalayas, Nepal"></a>
261   <a href="#" data-caption="Ghandruk, obteniendo una buena
    sensación de lo local"></a>
262   <a href="#" data-caption="¡Este es un vídeo de Nepal!"
    ><iframe width="500" height="333" src=
    "http://www.youtube.com/embed/sseAV4yOMbo" frameborder="0"
    allowfullscreen></iframe></a>
```

Observa que las diapositivas están encapsuladas en una etiqueta `<a>`, tanto el texto descriptivo como la imagen o el `<iframe>` que contiene el vídeo en la tercera diapositiva.

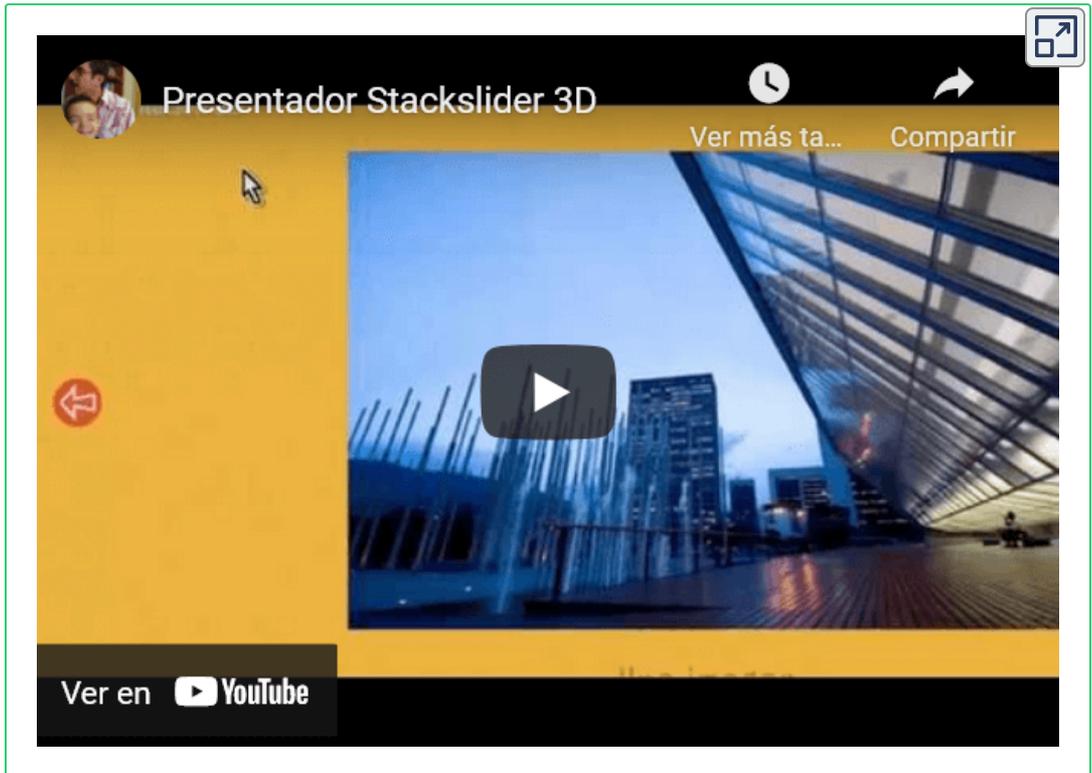
## 3.8 StackSlider



StackSlider es un complemento de control deslizante de imágenes jQuery altamente experimental que explora un concepto de visualización diferente y divertido para las miniaturas, utilizando transformaciones y perspectivas CSS 3D. La idea es navegar a través de dos pilas de imágenes donde la transición se asemeja a un elemento que se levanta de la parte superior de la pila y se mueve/gira al centro del contenedor para ser visto. La imagen vista anteriormente cae en la pila opuesta ([Pedro Botelho](#), 2012).

Aquí tenemos otra idea similar a la de Pete R., pues se trata de diapositivas apiladas, que se van desplazando con un efecto 3D. Para ver un flujo continuo, mantén presionado el puntero del ratón sobre las flechas de navegación.

En el siguiente vídeo puedes observar cómo se hace la edición en este presentador; no obstante, hemos incluido la ayuda con descartes.html, pues al incorporar vídeos y objetos interactivos, el presentador no se adapta adecuadamente en pantallas más pequeñas.



El vídeo fue realizado en 2014, año en el que era posible usar los servicios del Drive de Google y el Dropbox, para alojar nuestras páginas web. Actualmente, estos servicios se han eliminado; por ello, la parte final del vídeo, simplemente... ignórala.

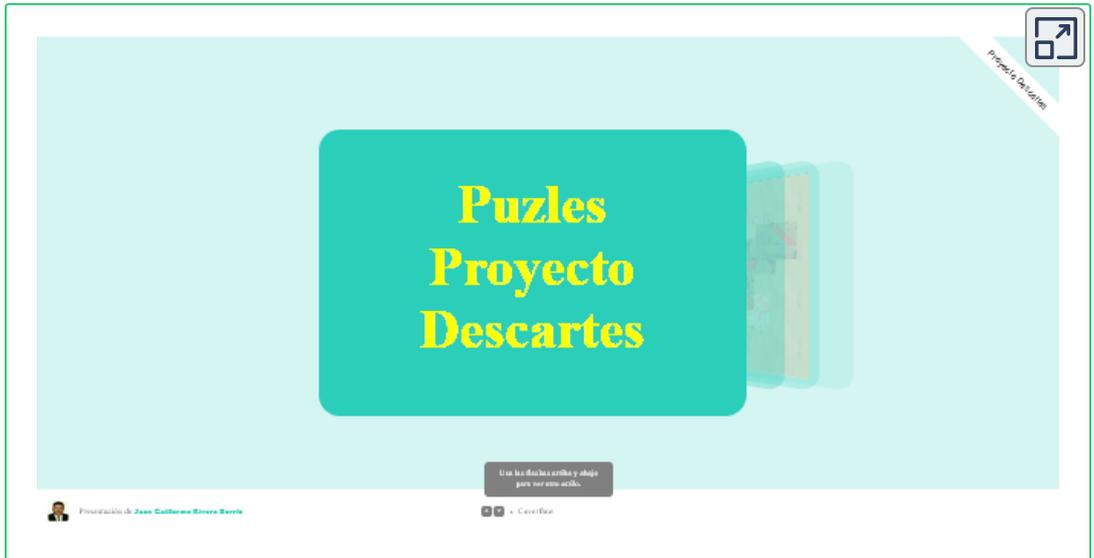
## 3.9 Presentador de panel triple



Un ejemplo más, antes de pasar a nuestro último presentador de diapositivas. Se trata de un presentador que muestra tres paneles, pero que nos permite que incluyamos todas las diapositivas que queramos. Fue diseñado por ([Mary Lou](#) en 2012). Incluir vídeos u objetos interactivos, es bastante sencillo:

```
<figure>
  
  <h1 style="color:red">Un vídeo de YouTube</h1>
  <h2><iframe width="480" height="460" src=
    "https://www.youtube.com/embed/8xdRvWSvFpQ" frameborder=
    "0" allow="accelerometer; autoplay; encrypted-media;
    gyroscope; picture-in-picture" allowfullscreen
    ></iframe></h2>
</figure>
```

## 3.10 Bespoke.js



Para interactuar con la presentación, haz clic sobre la diapositiva y luego presiona la barra espaciadora o las teclas de dirección, también funciona con un control remoto de diapositivas. Es importante tener en cuenta que cuando estamos interactuando con una escena, una página web o estamos reproduciendo un vídeo, debemos hacer clic fuera de la diapositiva para que los controles de desplazamiento se vuelvan a activar.

El presentador fue diseñado por [Dan Allen](#) en 2014, pero ha continuado con actualizaciones; por ejemplo, nuevas transiciones como el Nebula y Voltaire.



## Procedimiento de diseño

Hasta aquí, seguramente, no necesitarás mayores explicaciones de cómo diseñar tu presentación; no obstante, destacamos algunos aspectos a continuación.

Como siempre, debes abrir el archivo index.html para intervenirlo:

- **Pestaña forkme.** Aparece en la esquina superior derecha de la presentación. Al final, antes del pie de página (footer), aparece el bloque que podemos modificar. La referencia es opcional, en nuestro ejemplo, el usuario accede a la página de la Red Educativa Digital Descartes.

```
<div id="forkme"><a href="http://proyectodescartes.org/">Proyecto Descartes</a></div>
```

Puedes, si lo deseas, eliminar este bloque.

- **Pie de página.** Luego del bloque anterior, aparece el bloque de pie de página, que podemos modificar cambiando: la imagen del autor de la presentación (profile.jpg), el nombre del autor y el enlace a tu twitter, LinkedIn, etc., en nuestro ejemplo, hemos dejado el enlace a Dan Allen.

```
<div id="forkme"><a href="http://proyectodescartes.org/">Proyecto Descartes</a></div>
```

Para el diseño de las diapositivas, observa el siguiente vídeo:



Al igual que la última versión de Reveal.js, podemos eliminar la monotonía de una presentación, cambiando las transiciones en el momento que estemos presentando nuestra propuesta.

Aquí terminamos este capítulo, el que viene es mucho más emocionante.



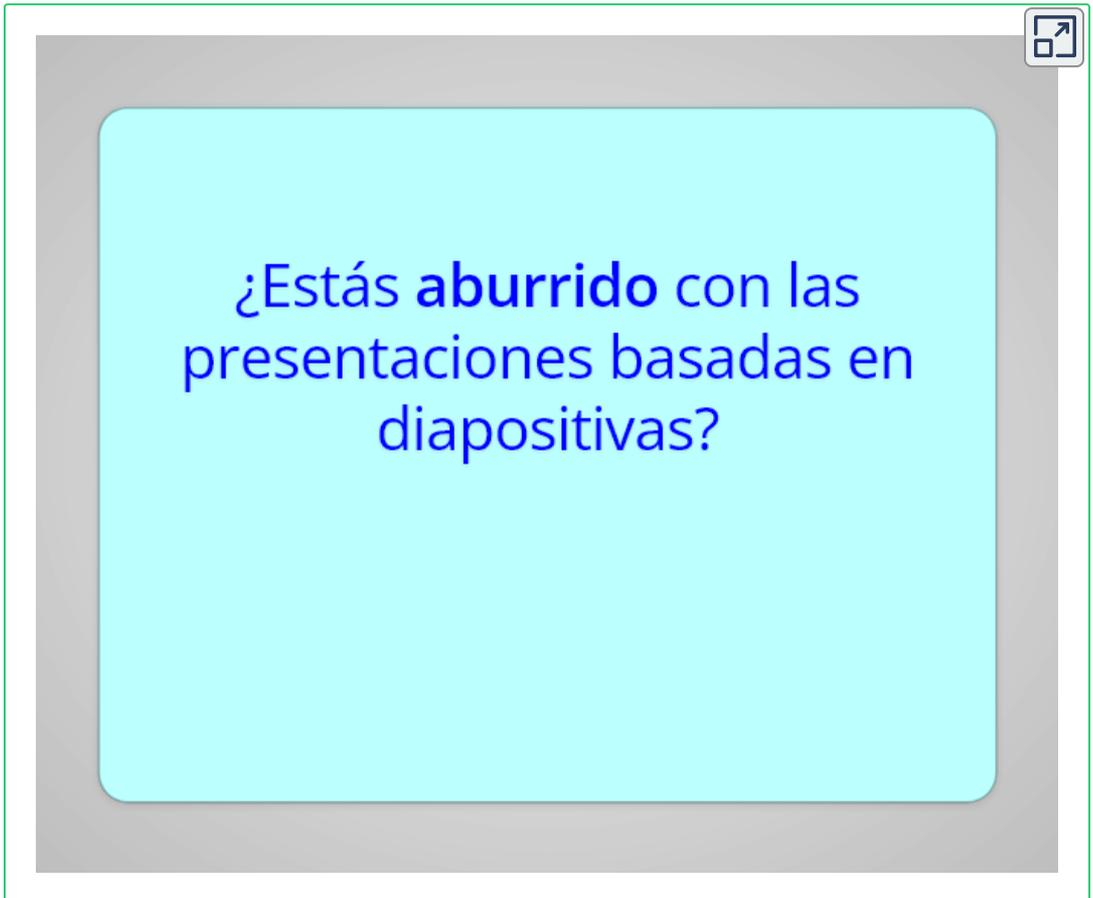
# Capítulo IV

**impress.js\***



## 4.1 Introducción

Impress.js\*, en adelante **impress**, está inspirado en la idea de Prezi. Es una herramienta para crear presentaciones con el poder de CSS3, tales como:



**¡Haz clic sobre la primera diapositiva y luego usa la barra espaciadora o flechas del teclado!**

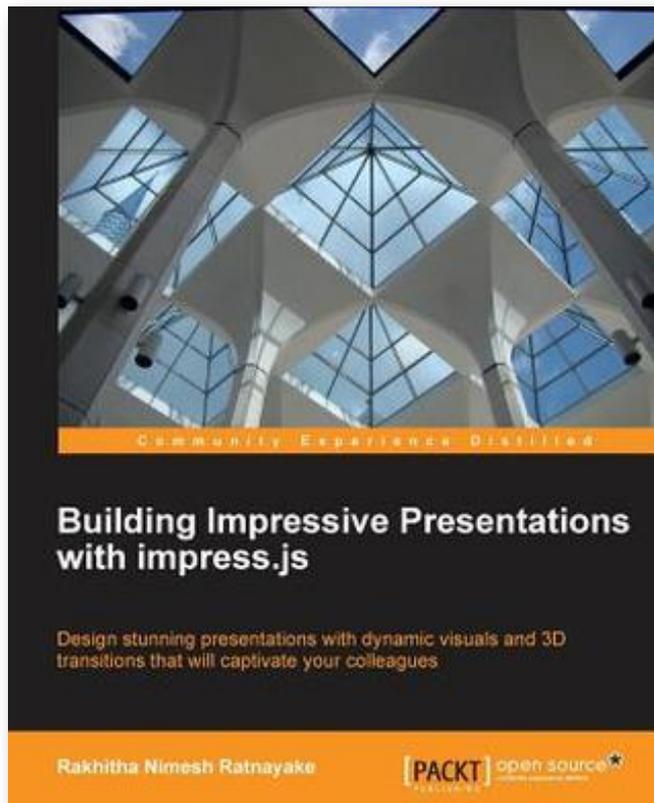
Impress.js fue creado en 2011-2012 por [Bartek Szopka](#), con actualizaciones realizadas, hasta la fecha, por Henrik Ingo (puedes consultar [aquí](#) las últimas versiones). En este capítulo usaremos la versión 1.1.0 de 2020, que incluye, entre otros, los siguientes complementos:

- Reproducción automática
- Controles para pausa y reproducción
- 5 nuevas presentaciones de demostración en la carpeta ejemplos
- Elimina el código que impedía que impress.js se ejecutara en dispositivos móviles
- 19 nuevos complementos
- Cambio a licencia MIT (Copyright 2011-2016 Bartek Szopka, Copyright 2015-2020 Henrik Ingo)

Con el proyecto, estaba aprendiendo y explorando las posibilidades abiertas por las nuevas transformaciones CSS 3D. Si bien requería que los usuarios crearan presentaciones codificando manualmente HTML y CSS, rápidamente se convirtió en un marco popular para aquellos que desean un poco más de sorpresa en sus presentaciones, ¡después de 20 años de PowerPoints! O, como en mi caso, entre aquellos que prefieren usar software de código abierto también para sus necesidades de escritorio. ¡Incluso después de un paréntesis relativo de 5 años, con 33k favoritos, todavía está sólidamente en el Top 100 de los proyectos favoritos de GitHub! (Henrik Ingo).

El ejemplo de la introducción corresponde a la primera versión de Szopka.

Si bien, se requiere tener conocimiento de HTML y CSS para obtener cambios significativos en el diseño, con lo aprendido en el apartado HTML y en las partes 2 y 3 de este taller, es suficiente para obtener excelentes presentaciones. Existen varios tutoriales que te permiten desarrollar tus presentaciones, pero para aprender Impress el mejor camino es realizando los ejercicios que plantearé en este capítulo, algunos de ellos, adaptados del libro [Building Impressive Presentations with impress.js](#), cuyo autor es Rakhitha Nimesh Ratnayake.



## 4.2 Ejercicio 1. Presentador de diapositivas en el plano



*Presentador de desplazamiento en el plano*

*Las diapositivas se desplazan horizontal o verticalmente en el plano, se incluyen textos, imágenes, vídeos y objetos interactivos en este ejemplo.*

Pasa el puntero del ratón sobre la primera diapositiva y observarás los controles de reproducción en la esquina inferior derecha, igual puedes usar el teclado. En este primer ejercicio, lo primero que debes hacer es descargar el ejemplo y abrir los archivos `index.html` y `ejemplo1.css` (carpeta `css`) con un editor de textos sin formatos.

Una vez comprendas como editar y crear tu presentación, con este modelo, te darás cuenta que puedes obtener resultados similares a las presentaciones del capítulo 3, tales como Deck.js, DZslides, Google I/O Slides, Fathom, CSS Slider , Flowtime.js, Presenteer y Slider.css. Así que, a prestar atención a este primer tutorial.

## 4.2.1 Hojas de estilo externas e internas

No pretendemos explorar todo el código css3 que utiliza impress.js, pero sí abordar algunos elementos que puedes intervenir para ajustar la presentación según tus gustos.

Inicialmente, en el archivo **ejemplo1.css**, podemos realizar los siguientes cambios:

**En el cuerpo (<body>).** En este bloque, puedes cambiar el color de fondo de la presentación que, para el ejemplo, es AACCB. Igualmente, se define el color de los textos; no obstante, en el archivo index.html lo vamos a cambiar.

```
29 body {
30     font-family: 'PT Sans', sans-serif;
31     min-height: 740px;
32     background: #aacbb;
33     color: #6644ff;
34 }
```

En la clase paso (<step>). Esta es un clase obligada para cada diapositiva, la cual puedes intervenir fácilmente. No obstante, tienes la opción de aceptar el diseño tal como está.

```
36  .step {
37      position: relative;
38      width: 900px;
39      height: 700px;
40      padding: 40px 60px;
41      margin: 20px auto;
42      box-sizing: border-box;
43      line-height: 1.5;
44      background-color: white;
45      border-radius: 10px;
46      box-shadow: 0 2px 6px rgba(0, 0, 0, .1);
47      text-shadow: 0 2px 2px rgba(0, 0, 0, .1);
48      font-family: 'Open Sans', Arial, sans-serif;
49      font-size: 24px;
50      letter-spacing: -1px;
51  }
```

Veamos algunas intervenciones posibles:

**padding: 40px 60px;**

Este elemento es nuevo en nuestro libro, pero sencillo de entender: "La propiedad CSS padding establece el espacio de relleno requerido por todos los lados de un elemento". Es decir, las márgenes asignadas a las diapositivas (no confundir con *margin*); por ejemplo, en la primera diapositiva, la margen superior es de 40px y las laterales de 60px. Puedes probar cambiando los valores y observando el resultado.

**background-color: white;**

Aquí podemos cambiar el color de fondo de las diapositivas (diferente al color de fondo de la presentación).

**border-radius: 10px;**

Define la curva en los extremos de las diapositivas, prueba con 40px y observa el resultado.

**box-shadow: 0 2px 6px rgba(0, 0, 0, .1);**

En este elemento podemos modificar la sombra de la diapositiva; por ejemplo, ampliemos el tamaño de la sombra y aumentemos la opacidad del color, así: `box-shadow: 0 12px 16px rgba(0, 0, 0, .9);`, y observarás una sombra menos sutil a la actual.



Ahora, en el archivo **index.html**, encontrarás estilos en la cabecera, de los cuales te recomendamos modificar los correspondientes a la clase **slide**:

```
.slide{
    color: #395B91;
    font-size: 50px;
    font-weight:bold;
    width:900px;
    height:700px;
    text-align:center;
}

.slide2{
    color: yellow;
    font-size: 50px;
    font-weight:bold;
    width:900px;
    height:700px;
    text-align:center;
    background-image: url("imagenes/img1.jpg");
    background-repeat: no-repeat;
}

.slide3{
    color: yellow;
    font-size: 40px;
```

Estas clases definen estilos adicionales a nuestras diapositivas, de tal forma que nuestra presentación impacten más. En la imagen, observa que la clase **slide** tiene definido para los textos, color 395B91, tamaño de 50px y alineación centrada. Para la clase **slide2** cambia el color a amarillo, e incluye una imagen de fondo (background-image: url("imagenes/img2.jpg")).

Todas las clases, definen dimensiones de 900×700 píxeles; sin embargo, si lo deseas, puedes cambiar los valores en alguna clase. Es importante usar imágenes, libres de derechos de autor.

La clase **slide4**, usa un color de fondo de la diapositiva, igual a la de este libro "aliceblue".

## Las clases slide son opcionales

Impress.js es un lienzo (canvas) infinito en el que podemos imprimir, obviamente, infinitas diapositivas. Las diapositivas se pueden desplazar automáticamente cada 2 segundos, que puedes modificar en:

```
<!-- Duración, en milisegundos, de la transición  
entre diapositivas -->  
<div id="impress" data-transition-duration="2000">
```

## 4.2.2 Contenido de las diapositivas

Este segundo paso, ya lo tienes bien estudiado en los capítulos anteriores; sin embargo, sólo para este ejemplo, explicaremos cómo se incluyeron los contenidos en las siete diapositivas, haciendo especial énfasis en la última, por el efecto de rotación en el texto.

En este ejemplo, debes prestar atención a las coordenadas en el plano donde se localizará cada diapositiva.

**Diapositiva 1.** El contenido de esta diapositiva es sólo texto:



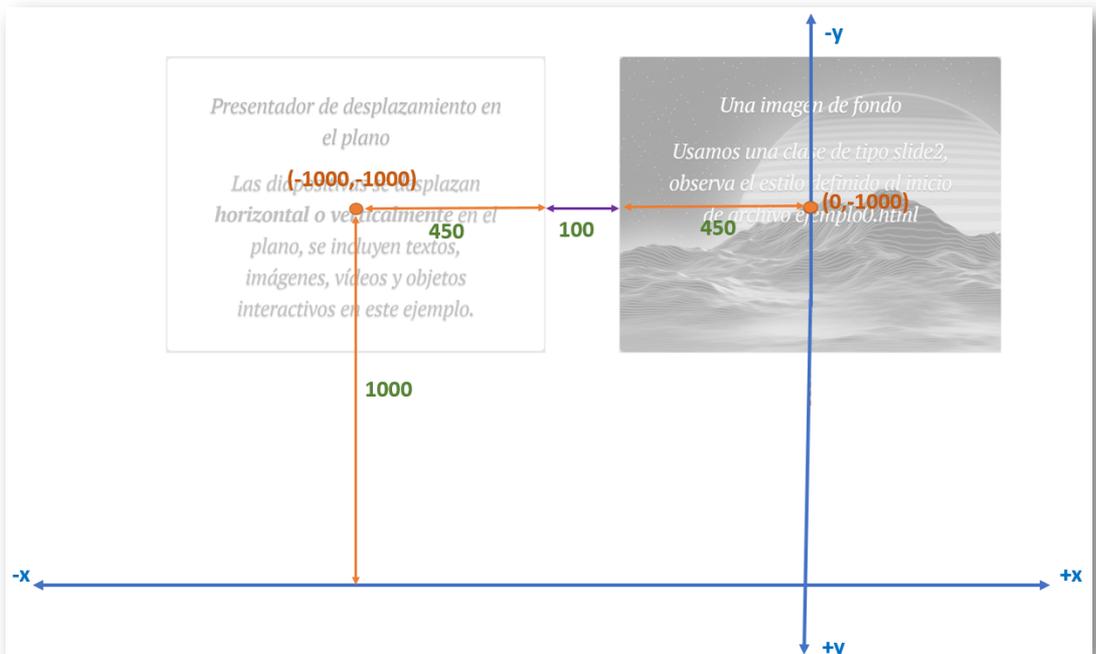
El código para esta diapositiva es:

```
<!-- Primera diapositiva (texto) -->
<div id="Inicio" class="step slide" data-x=
"-1000" data-y="-1000">
  <p style="color:red">Presentador de
desplazamiento en el plano</p>
  <p>Las diapositivas se desplazan <strong>
horizontal o verticalmente</strong> en el
plano, se incluyen textos, imágenes, vídeos
y objetos interactivos en este ejemplo.</p>
</div>
```

Cada diapositiva tendrá un identificador único, la primera la hemos identificado como Inicio (id = "Inicio"). Si por error, asignas el mismo identificador, las diapositivas se solaparán.

Luego del identificador viene las **clases** de la diapositiva (class="step slide"), donde la clase "paso" (step) es obligado para que la presentación tenga uniformidad y la clase "slide" es opcional (como ejercicio, elimina la clase slide en la segunda diapositiva).

Para comprender dónde posicionamos las diapositivas (coordenadas), nos apoyaremos en la siguiente imagen:



Como lo dijimos antes, impress.js es un lienzo infinito. El centro de este lienzo está en las coordenadas  $(-1000, -1000)$ , donde  $x$  crece a la derecha y  $y$  hacia abajo.

Nuestra primera diapositiva la hemos posicionado en este centro del lienzo, a través de las expresiones `data-x="-1000"` `data-y="-1000"`. Estas coordenadas corresponden al centro de la diapositiva, tal como lo muestra la imagen anterior.

Resumiendo, en el bloque `<div>` identificamos la diapositiva, asignamos las clases y la posición en el lienzo.

Finalmente, hemos incluido dos textos con la etiqueta párrafo `<p>`, de los cuales no hacemos mayor descripción, pues el estilo ya lo hemos explicado, excepto por el cambio de color en el primer texto.

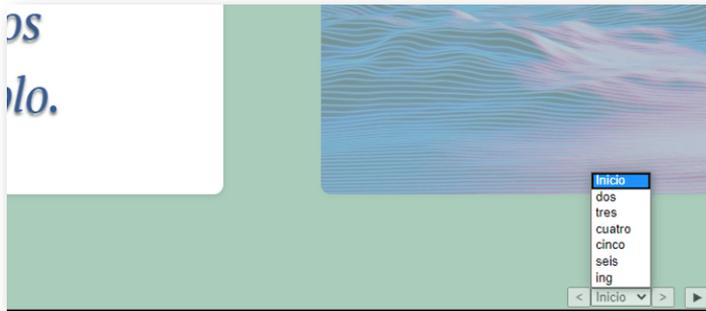
**Diapositiva 2.** El contenido de esta diapositiva es texto, más una imagen de fondo:



El código para esta diapositiva es:

```
<!-- Segunda diapositiva (imagen 1)-->
<div id="dos" class="step slide2" data-x="0"
data-y="-1000">
  <p style="color:white">Una imagen de fondo</p>
  <p>Usamos una clase de tipo slide2, observa
  el estilo definido al inicio de archivo
  ejemplo0.html</p>
</div>
```

La hemos identificado con la palabra "dos". La identificación es importante para enlazar la diapositiva o, como lo observarás en la siguiente imagen, para seleccionarla desde cualquier punto de la presentación, usando los controles que aparecen en la esquina inferior derecha.



Las clases de la diapositiva son class="step slide2", donde la segunda la hemos definido con una imagen de fondo (background-image: url("imagenes/img1.jpg");), en los estilos de la cabecera.

Volvamos, ahora, al posicionamiento de la diapositiva (observa la imagen en la diapositiva anterior).

Recordemos que la primera diapositiva se encuentra en las coordenadas (-1000, -1000). Por otra parte, el ancho de las diapositivas lo hemos definido en 900px; además, asignamos 100px de espacio horizontal entre diapositivas. Así las cosas, sin alterar la posición vertical, la próxima diapositiva se encontrará en  $(-1000+450+100+450, -1000) \rightarrow (0, -1000)$ ; por ello, hemos puesto `data-x="0" data-y="-1000"`. Dado este diseño posicional, bastará sumar 1000px a las abscisas de las siguientes diapositivas.

Los textos de la diapositiva, al igual que en la anterior, están encapsulados en etiquetas párrafo.

**Diapositiva 3.** El contenido de esta diapositiva es texto, más una imagen interna:



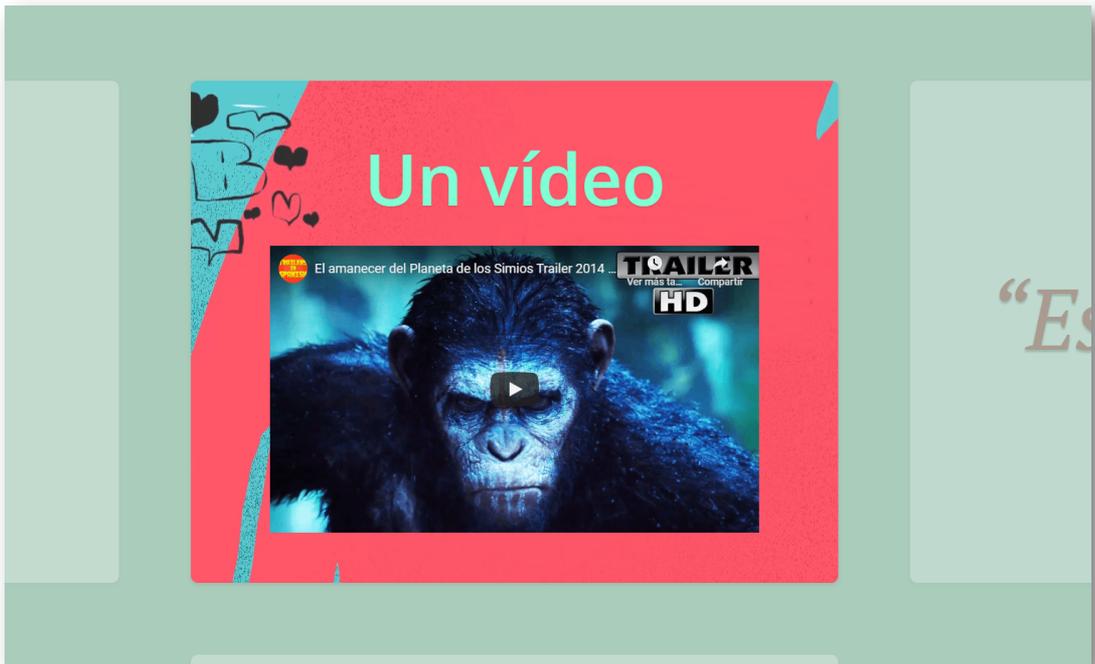
El código para esta diapositiva es:

```
<!-- Tercera diapositiva (imagen 2)-->
<div id="tres" class="step slide" data-x=
"1000" data-y="-1000">
<p style="color:blue">Una imagen dentro de
la diapositiva</p>

</div>
```

En esta diapositiva, solo resta aclarar que podemos incluir una imagen interna, usando la etiqueta <img>.

**Diapositiva 4.** El contenido de esta diapositiva es un vídeo:



Como lo hemos hecho en todos los presentadores de este libro, incluimos un vídeo, el cual se comporta excelentemente, incluso si la diapositiva esta inclinada en el plano o en 3D, como lo veremos más adelante. Quizá, aclarar que el parámetro **rel=0** ya no es funcional en TouTube, pero por costumbre lo hemos puesto.

**Diapositiva 5.** El contenido es un objeto interactivo:



Igualmente, como en el vídeo, el comportamiento de este puzle, diseñado con DescartesJS, es excelente. Pero, lo novedoso de esta diapositiva, es que la hemos diseñado para que se desplace verticalmente.

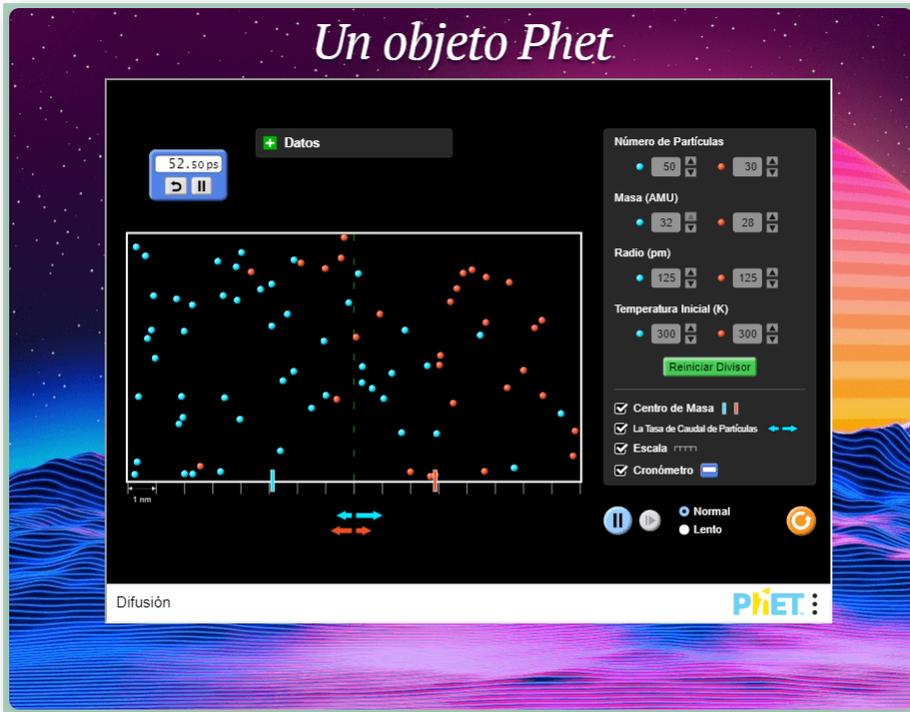
Con esta diapositiva, y la siguiente, evidenciamos una de las fortalezas de impress.js, y es la posibilidad de navegar por su infinito lienzo. Podemos, entonces, diseñar presentaciones al estilo de Flowtime.js, de tal forma que generemos secciones con este tipo de desplazamiento:

```
<!-- Quinta diapositiva (objeto interactivo)-->
<div id="cinco" class="step slide" data-x="2000"
data-y="-200">
  <p style="color:purple">Un objeto interactivo
  </p>
  <iframe style="width: 790px; height: 450px;"
  src=
  "https://proyectodescartes.org/plantillas/mat
  eriales_didacticos/Puzle_giratorio2-JS/index.
  html"></iframe>
</div>
```

Observa que hemos conservado la abscisa de la diapositiva anterior y hemos avanzado 800 píxeles en el eje y; es decir,  $-1000+800 = 200$ . Los 800 surgen de sumar la mitad de la altura de cada diapositiva, más 100 píxeles de espacio vertical entre ellas:  $350+350+100$  (recuerda que en los estilos de la cabecera, definimos una altura (height) de 700px).

Así las cosas, el posicionamiento de la diapositiva es `data-x="1000" data-y="-200"`. Un efecto interesante se observa si usando el cuadro de control, hacemos clic sobre una diapositiva que esté en una sección a la derecha o a la izquierda, pues el desplazamiento será diagonal... ¡pruébalo!

**Diapositiva 6.** El contenido es otro objeto interactivo:



Otra diapositiva diseñada para la sección anterior; es decir, aumentamos en 800px la ordenada y, quedando `data-x="1000"` `data-y="600"`. Hemos usado una simulación de Phet de Colorado y un fondo para la diapositiva con la clase `slide2`.

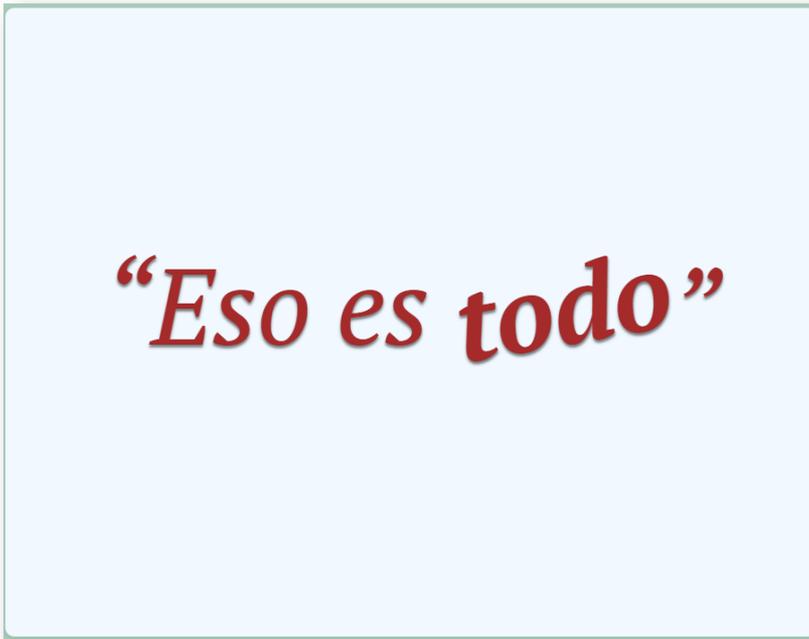
**Diapositiva 7.** El contenido es un texto con efecto de rotación:

En este modelo de presentación, hemos incorporado, en la hoja de estilos externa (`style.css`), el complemento `gin`, que permite tres posibles efectos en los textos: posición, escala y rotación. Este último lo hemos aplicado en la palabra "todo". Para usarlo, el identificador deber ser `gin`; obviamente, sólo lo podemos usar en una diapositiva, para evitar solapamientos.

Este es el código de la diapositiva:

```
<!-- Séptima diapositiva (texto)-->
<div id="ing" class="step slide4" data-x="3000"
data-y="-1000">
    <q style="color:brown; font-size:120px;
    position:relative; top:200px">Eso es <b class
    ="rotating">todo</b></q>
</div>
```

Observa que modificamos color, tamaño y posición del texto: `style="color:brown; font-size:120px; position:relative; top:200px"` y, para la palabra rotado, la clase `rotating`.



Bueno... como dice la diapositiva, ¡Eso es todo, para este modelo!

Bueno... casi todo.

En el siguiente vídeo, presentamos con más detalle lo explicado en este apartado.



A continuación, presentamos un segundo ejercicio que nos conduce a un nuevo modelo, en el que combinamos el escalado y el posicionamiento de las diapositivas en diferentes puntos del lienzo impress.js.

Para simplificar, sólo usaremos textos e imágenes, pero recuerda que puedes incluir objetos interactivos en tu presentación.

## 4.3 Ejercicio 2. Diapositivas con escala y rotación



Las imágenes de este ejemplo, son tomadas de <https://ar.pinterest.com/>

El ejercicio es planteado en el libro de [Rakhitha Nimesh Ratnayake](#)

## Diapositiva panorámica o resumen

Iniciamos con la última diapositiva, que olvidamos reseñar en el modelo anterior. Se trata de la diapositiva panorámica, la cual nos presenta una **visión de conjunto** de nuestro diseño:



El resumen no tiene ninguna información y muestra todos los pasos de nuestra presentación. Podemos permitir que los usuarios hagan clic en los pasos de la descripción general y carguen directamente el paso en la pantalla. Por defecto, la descripción general también se considera un paso y se podrá hacer clic en ella (Ratnayake, 2013).

En este modelo, tal como lo sugiere Ratnayake, hemos permitido que los usuarios puedan hacer clic en cualquier diapositiva de la panorámica, por contraste al modelo anterior que, si prestas atención a lo que vamos explicar a continuación, lo puedes modificar a tu gusto.

Para los dos modelos, el código HTML es igual, excepto por el posicionamiento en x (`data-x`) del modelo 1, que tuvimos que correrlo 2000px para centrar la diapositiva. En este modelo, el código es el siguiente::

```
<!-- Diapositiva panorámica o resumen -->
<div id="overview" class="step" data-x="-1000"
data-y="-1000" data-scale="5">
</div>
```

La identificación debe ser *overview* (resumen), la clase *step*, las coordenadas que garanticen el centrado de la diapositiva que, para este ejemplo, coincide con el centro del lienzo y, finalmente, una escala (`data-scale`) que elegido en cinco. He aquí un nuevo atributo de `impress.js`, que se comporta como un `zoom` que amplía o reduce la diapositiva, según la escala que le asignemos.

Pero, con este código no garantizamos que las diapositivas del resumen estén activas. Para lograrlo, desarrolla el siguiente procedimiento:

- **Hoja de estilos externa.** En el archivo `ejemplo2.css` hemos agregado el comando `display: none;`, que nos permite seleccionar una diapositiva haciendo clic sobre ella.

```
53 #overview {
54     background-color: transparent;
55     border: none;
56     box-shadow: none;
57     display: none;
58 }
```

- **Hoja de estilos de la cabecera.** Al final, agregamos líneas de comando CSS para que aparezca el puntero del ratón, cuando lo pasemos sobre la panorámica y aumentamos la opacidad.

```
71     .impress-on-overview .step {  
72     opacity: .6;  
73     cursor: pointer;  
74     }
```

## Primera diapositiva - Inicio

Es la diapositiva central cuyo centro coincide con el del lienzo y con un id="Inicio". Tiene una escala de 3 (data-scale=3), lo que genera un cambio en sus dimensiones de 900×700 pixeles a 2700×2100 pixeles, los cuales debes tener anotadas para la posición de las siguientes diapositivas.

```
<div id="Inicio" class="step slide2" data-x=  
"-1000" data-y="-1000" data-scale="3">  
  <p>Construyendo Presentaciones Impresionantes  
  </p>  
  <q>Impress.js</q>  
</div>
```

## Segunda diapositiva - Norte

La hemos posicionado arriba de la central (Inicio), con id="N" (norte), lo que implica que su posición en x permanece igual.

Pero en y, debemos sumarle -1050px (mitad de la central), -100px (espacio) y -350px (mitad de la diapositiva N); es decir,  $\text{data-y} = -1000 - 1050 - 100 - 350 = -2500$  (recuerda que decrece hacia arriba).

```
<div id="N" class="step slide3" data-x="-1000" data-y="-2500" >
  <h1>Diapositiva Norte</h1>
</div>
```

## Tercera diapositiva - Este

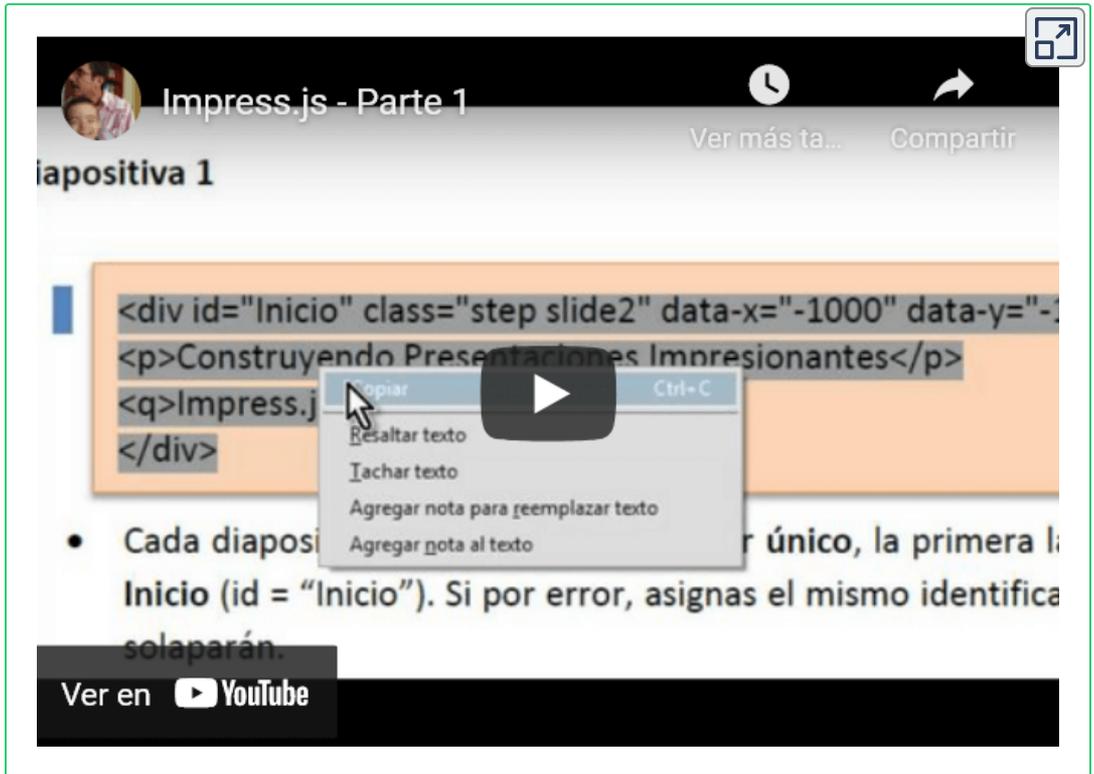
Aquí la ordenada es igual a la centra (-1000), pero en x debemos sumar 1350px (la mitad de la central), +100px (espacio), +450 (la mitad de la diapositiva E); es decir  $\text{data-x} = -1000 + 1350 + 100 + 450 = 900$ px. Finalmente, la hemos rotado 90° en el sentido de las manecillas del reloj ( $\text{data-rotate}=90$ ).

```
<div id="E" class="step slide3" data-x="900" data-y="-1000" data-rotate="90">
  <h2>Diapositiva Este</h2>
</div>
```

He aquí otro atributo interesante de impress.js, la posibilidad de rotar nuestras diapositivas, que se suma a la rotación de textos que vimos en el modelo anterior. Es decir, ya tienes buenos elementos para impresionar a tu audiencia.

Dejamos para tu análisis las demás diapositivas.

En este vídeo, publicado en 2014, presentamos el diseño realizado para la primera versión de impress.js que, en esencia, sirve para que realices el ejercicio con la versión 2020.



¿Qué ocurre si elegimos una diapositiva central en las coordenadas (0, 0)? La respuesta es que el impress.js se posiciona sin problemas en estas coordenadas, tal como lo veremos en el siguiente ejercicio, en el que practicaremos el posicionamiento en forma circular.

## 4.4 Ejercicio 3. Diapositivas circulares



Observa la diapositiva resumen, para que tengas una idea del ejercicio a realizar.

### Primera diapositiva - Central

En este modelo la diapositiva central (Inicio), estará en las coordenadas (0, 0), la cual tiene una imagen de fondo ([i.pinimg.com/](http://i.pinimg.com/)), texto y un enlace.

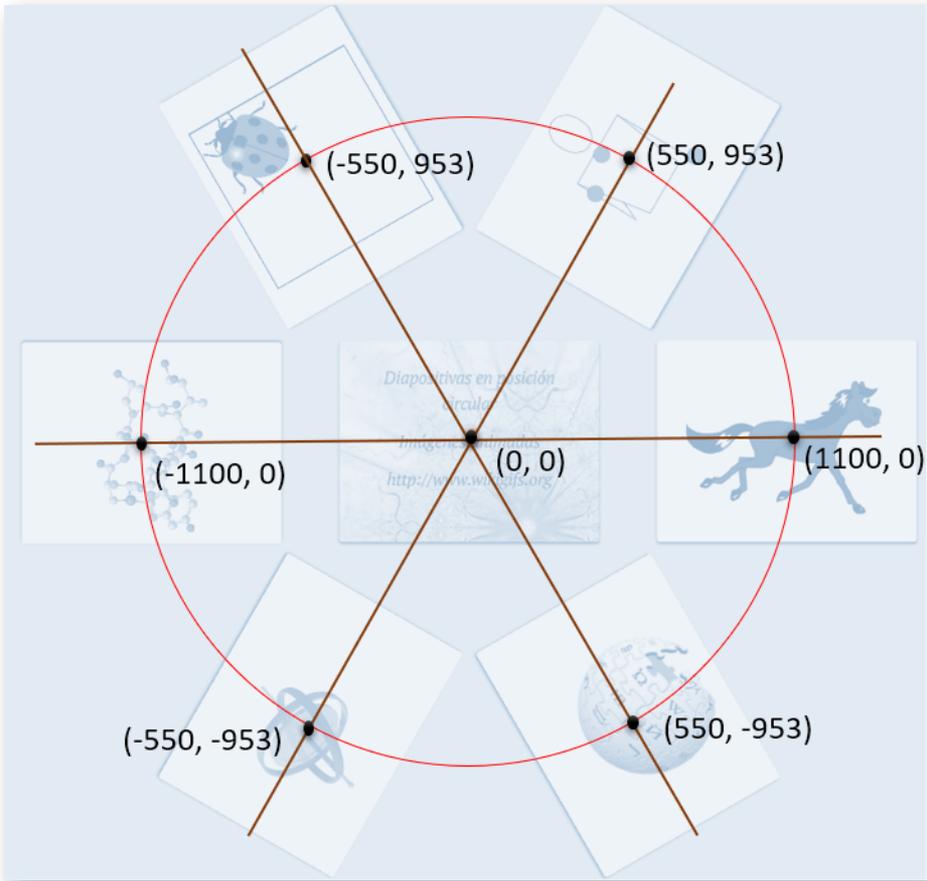
En el código, de esta primera diapositiva, se asignan las coordenadas  $\text{data-x} = 0$  y  $\text{data-y} = 0$ , además de una rotación de  $0^\circ$  ( $\text{data-rotate} = "0"$ ), la cual se puede omitir, pues por defecto esa es la rotación que `impress.js` asigna a las diapositivas.

```
<!-- Diapositiva central -->
<div id="Inicio" class="step slide2" data-x="0"
data-y="0" data-rotate="0">
  <p style="color:red">Diapositivas en
posición circular</p>
  <p><font style="vertical-align: inherit;"
><font style="vertical-align: inherit;"
Imágenes animadas</font></font></p>
  <p><a href=
"http://www.wikigifs.org/#Tripframe.gif"
target="_blank">http://www.wikigifs.org</a>
</font></font></p>
</div>
```

## Diapositivas circulares

La idea es generar seis diapositivas que formen una circunferencia alrededor de la diapositiva central, tal como se observa en la imagen de la siguiente página.

Aquí es necesario aplicar un poco de trigonometría. El radio de la circunferencia es  $r = 1100\text{px}$ , por lo que cualquier diapositiva sobre esta circunferencia estará en las coordenadas  $(r\cos\theta, r\sin\theta)$ , donde  $\theta$  lo medimos en sentido contrario a las manecillas del reloj.



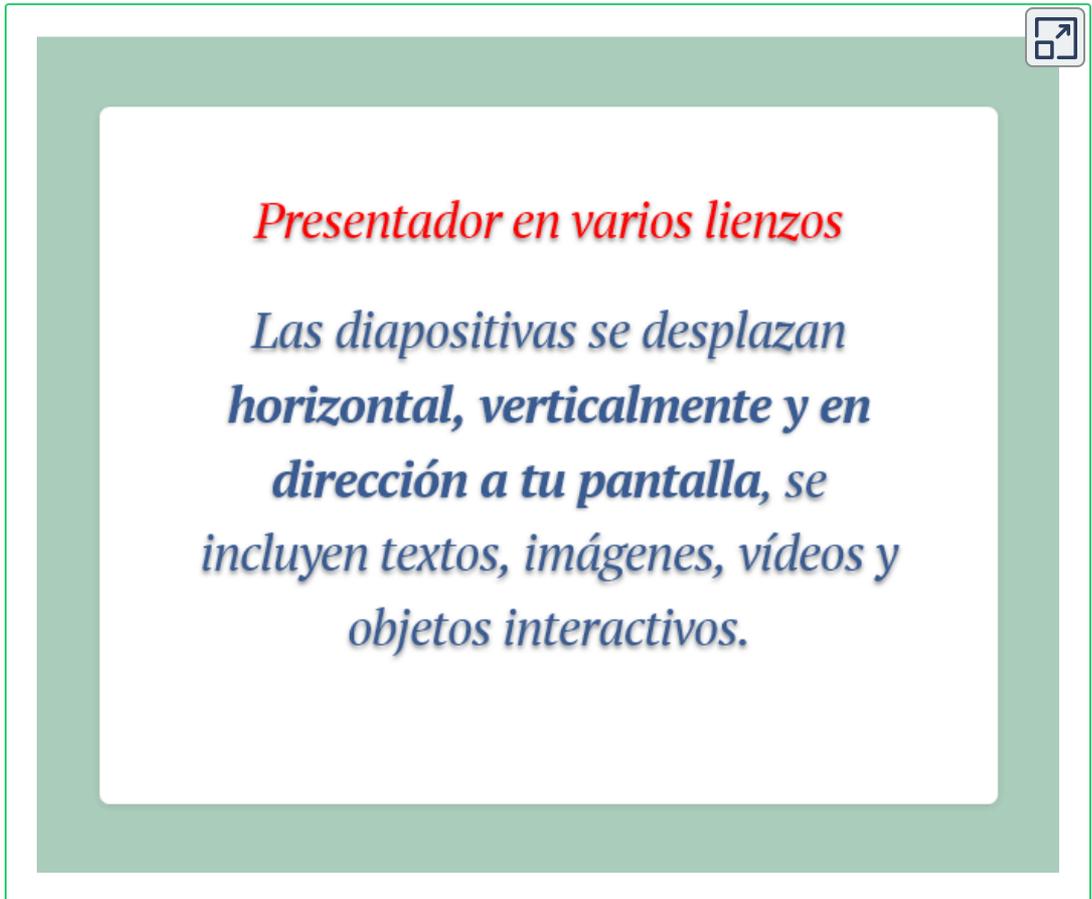
Como son seis diapositivas, el ángulo entre diapositivas contiguas es  $360^\circ/6 = 60^\circ$ ; es decir, a  $0^\circ, 60^\circ, 120^\circ, 180^\circ$ ... Por ejemplo, para  $120^\circ$ , las coordenadas serían:  $\text{data-x} = 1100 \cdot \cos(120^\circ) = -550$ ,  $\text{data-y} = 1100 \cdot \sin(120^\circ) = 953$ .

```
<!-- Diapositiva a 120 grados -->
<div id="Tres" class="step slide" data-x="-550"
data-y="953" data-rotate="120">
    
</div>
```

Si eres poco amante de las matemáticas, simplemente usa el modelo para insertar el contenido de las diapositivas. Hemos incluido dos ejemplos más en el modelo, para ocho y diez diapositivas (index8.html e index10.html), haz clic en las imágenes, para que los explores:



## 4.5 Ejercicio 4. Diapositivas en varios planos



*Presentador en varios lienzos*

*Las diapositivas se desplazan horizontal, verticalmente y en dirección a tu pantalla, se incluyen textos, imágenes, vídeos y objetos interactivos.*

Si bien es posible imprimir muchas diapositivas en el plano XY, también es posible que perdamos el control de las mismas cuando algunas se nos escapan de la panorámica. Una de las propiedades interesantes de impressi.js es que permite crear varios planos o, si se prefiere, varios lienzos, con sólo incluir la coordenada z.

Hemos usado el primer ejemplo de este capítulo, para reubicar las últimas cuatro diapositivas en diferentes planos o lienzos. La cuarta diapositiva la desplazamos 500px a la derecha y la ubicamos en un plano 3000px por debajo del plano inicial (`data-z="-3000"`):

```
<!-- Tercera diapositiva (imagen 2)-->
<div id="tres" class="step slide" data-x="1000"
data-y="-1000">
  <p style="color:blue">Una imagen dentro de
  la diapositiva</p>
  
</div>

<!-- Cuarta diapositiva (vídeo)-->
<div id="cuatro" class="step slide3" data-x=
"1500" data-y="-1000" data-z="-3000">
  <h2 style="color:aquamarine">Un vídeo</h2>
  <iframe width="680" height="400" src=
  "http://www.youtube.com/embed/Y6Fr0MSMTXo?rel
  =0" frameborder="0" allowfullscreen></iframe>
</div>
```

La quinta y sexta diapositiva están en los planos `data-z="-5000"` y `data-z="-7000"` respectivamente. La séptima la hemos puesto en un plano que está 1000px por encima del plano inicial.

La diapositiva resumen, te da una idea de cómo quedaron distribuidas las diapositivas. En esta diapositiva panorámica hemos usado una rotación alrededor del eje y, para una mejor comprensión de la distribución realizada (`data-rotate-y="-45"`).

Así aparece la diapositiva resumen:



y este es el código:

```
<!-- Diapositiva panorámica o resumen -->  
<div id="overview" class="step" data-x="-2000"  
data-y="-1000" data-scale="1.2" data-rotate-y=  
"-45">  
</div>
```

Puedes observar que las diapositivas 4, 5 y 6 se ven más pequeñas, pues se encuentran bastante lejos del plano inicial, la séptima por estar en un plano superior se ve más grande.

El uso de rotaciones alrededor de los tres ejes, nos permite diseños 3D como veremos en el siguiente ejemplo:

## 4.6 Ejercicio 5. Diapositivas en un cubo



En el ejemplo anterior usamos la instrucción `data-rotate-y="-45"` para la diapositiva resumen, lo que genera una rotación de  $45^\circ$  en el plano XZ. Impress incluye otros tipos de rotaciones con respecto a cualquier eje. Las seis diapositivas que se muestran en este modelo, utilizan la instrucción `data-rotate-x`, `data-rotate-y` y `data-rotate-z`.

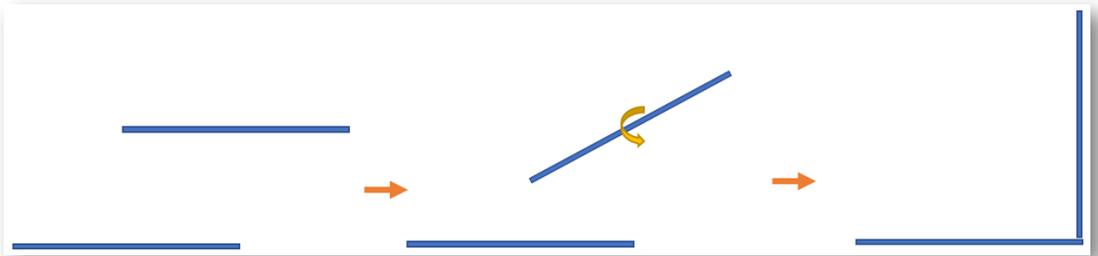
Uno de los grandes problemas, en nuestra formación, es la pérdida del pensamiento tridimensional - 3D -, pues desde que iniciamos la escuela, nos acostumbramos a las representaciones en el plano (plana es la pizarra, planos son nuestros cuadernos y libros, plana es la pantalla en la que lees este texto, etc.). Por ello, haremos un esfuerzo por explicar cómo podemos generar las caras de un cubo con impress.js pero, obviamente, confiamos en que puedas imaginar, mentalmente, cómo se mueven los objetos en el espacio tridimensional. Ahora, si crees que es un esfuerzo innecesario, simplemente usa el modelo para el diseño de una que otra de tus presentaciones, sin preocuparte por el pensamiento 3d. Como se expresan en España... vamos a por ello:



En este modelo, hemos definido las dimensiones de la diapositiva en  $800 \times 800$  píxeles, pues las caras del cubo son cuadradas. La primera diapositiva, identificada como cara1 (cara frontal), se encuentra en las coordenadas (0, 0) con un texto como el que aparece en la imagen anterior.

## Diapositiva lateral derecha del cubo

Nuestra primera tarea es posicionar la diapositiva de la cara lateral derecha del cubo (`id="cara2"`), la cual es perpendicular a la cara1, desplazada a la derecha 400px (`data-x=400`) y hacia al fondo otros 400px (`data-z=-400`,  $z$  crece en dirección hacia a tí y decrece penetrando la pantalla)...¿Te estás imaginando el movimiento?, si es así, observarás que hay que rotarla.



Si observas desde arriba (haz el esfuerzo), verás dos líneas, tal como aparece en el lado izquierdo de la imagen, con la segunda diapositiva desplazada hacia la derecha (400px) y hacia el fondo (-400px); ahora, solo nos resta rotarla  $90^\circ$  en el sentido contrario a las manecillas del reloj y alrededor del eje y (`data-rotate-y="90"`).

## Diapositiva posterior del cubo

Esta es más sencilla, pues sólo debemos enviar la primera diapositiva (frontal) al fondo y rotarla  $180^\circ$  con respecto al eje  $y$ . Observa que no nos hemos preocupado por el contenido, pues ya tienes suficiente información en los modelos anteriores.

Este es el código de las caras uno y dos:

```
<!-- Diapositiva cara 1 -->
<div id="Cara1" class="step slide2" data-x="0"
data-y="0" data-rotate="0">
  </br><h2 style="color:orange">Diapositivas
  en un cubo</h2>
  <p><font style="vertical-align: inherit;">
  Fotos de ciudades</font></p>
</div>

<!-- Diapositiva cara 2 -->
<div id="Cara2" class="step slide" data-x="400"
data-y="0" data-z="-400" data-rotate-y="90">
  
</div>
```

y este, el de la cara tres del cubo (posterior):

```
<!-- Diapositiva cara 3 -->
<div id="Cara3" class="step slide" data-x="0"
data-y="0" data-z="-800" data-rotate-y="180">
  
</div>
```

Te invitamos a que sigas analizando el código de las cinco caras restantes (lateral izquierda, superior e inferior), las cuales se pueden entender haciendo siguiendo cuidadoso de los movimientos realizados, tanto de desplazamiento como de rotación.

¿Qué podemos hacer si tenemos más de ocho diapositivas en nuestra presentación?, la respuesta es otro cubo, observa:



En el modelo 5 (ejercicio 5), que puedes descargar al inicio de este apartado, se encuentra esta variante al modelo (sólo hemos puesto las primeras cuartos caras en el segundo cubo). Puedes, también, intentar con diferentes modelo de presentación 3D, tal como lo expresa Szopka: "el límite es tu imaginación"; no obstante, si buscas un mínimo esfuerzo, el último apartado de este libro te sorprenderá.

## 4.7 Ejercicio 6. Diapositivas sin marcos

Seguramente, te habrás preguntado ¿cómo eliminar los marcos de las diapositivas, tal como aparece en la presentación de la introducción?, la respuesta es crear un *slide* que lo elimine. En el modelo uno, hemos dejado un archivo `index2.html`, modificando el `slide4`, así:

```
.slide4{
  color: black;
  font-size: 40px;
  font-weight:bold;
  width:900px;
  height:700px;
  text-align:center;
  background-color: rgba(0,0,0,0) ;
  box-shadow: 0 2px 6px rgba(0, 0, 0, 0) ;
}
```

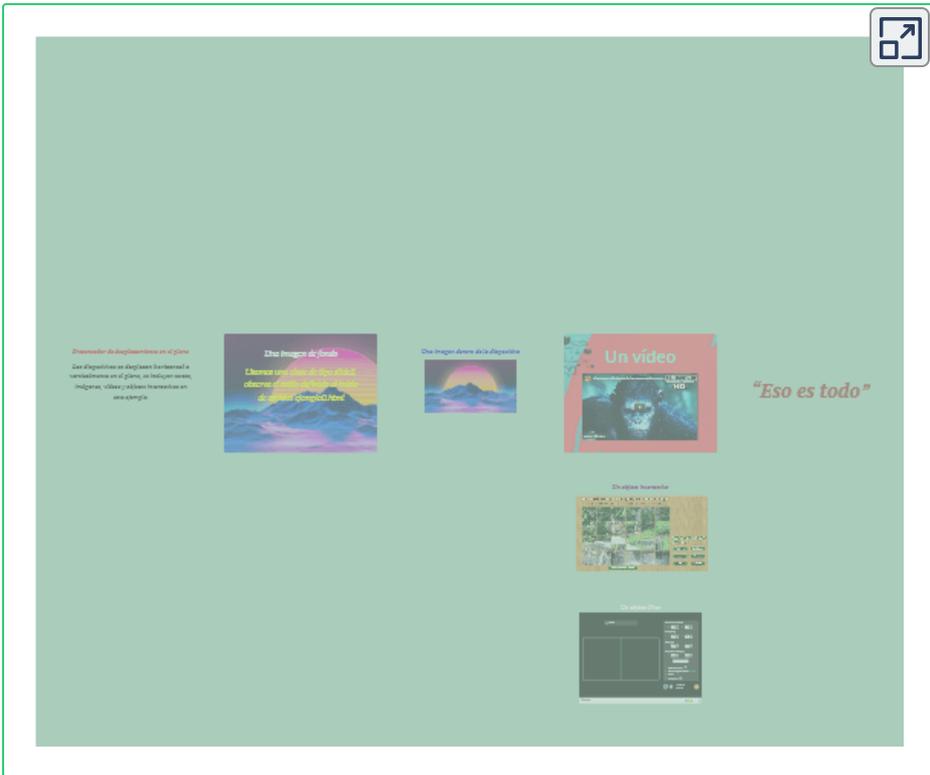
Observa que en el fondo de la diapositiva hemos puesto una opacidad cero o, de otra forma, una transparencia total:

`background-color: rgba(0,0,0,0);`

Como la caja aún muestra el borde, le damos el mismo tratamiento:

`box-shadow: 0 2px 6px rgba(0, 0, 0, 0);`

He aquí el resultado en algunas diapositivas:



Sólo hemos dejado los fondos de las diapositivas 2 y 4. En la última diapositiva, usamos los tres efectos de texto: positioning, scaling y rotating:

```
<!-- Séptima diapositiva (texto)-->
<div id="ing" class="step slide4" data-x="3000"
data-y="-1000">
  <q style="color:brown; font-size:120px;
position:relative; top:200px"><b class=
"positioning">Eso</b> <b class="scaling">es
</b> <b class="rotating">todo</b></q>
</div>
```

## 4.8 Editor de diapositivas - Strut

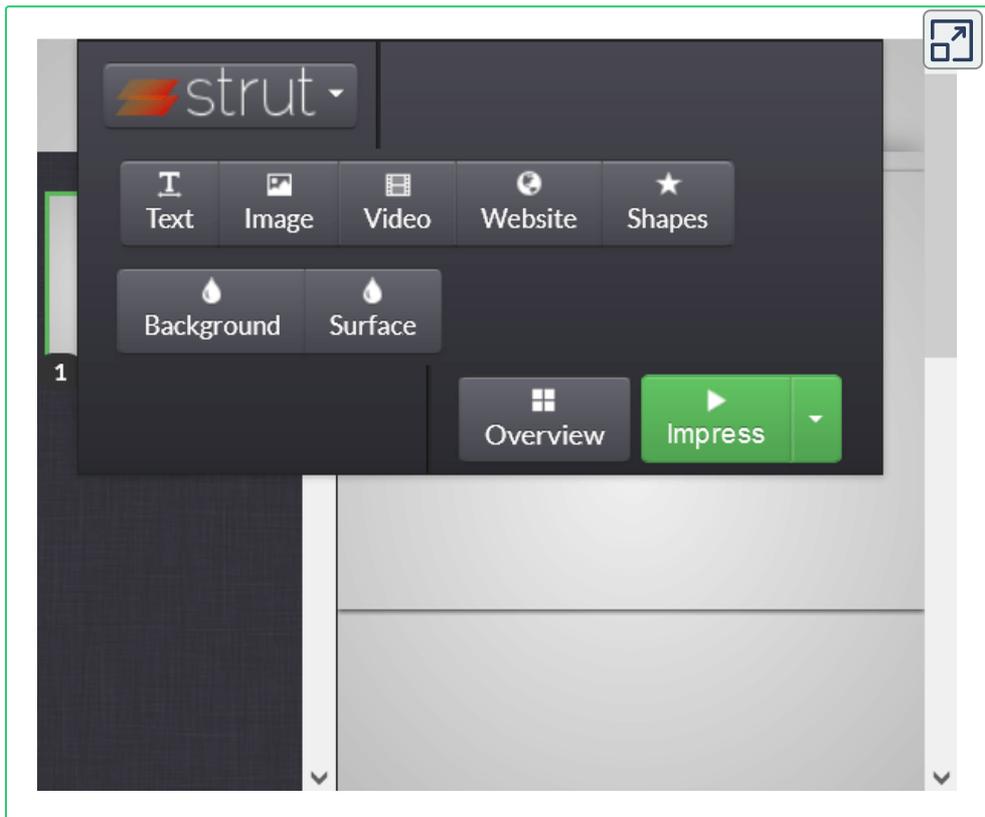
Este editor, diseñado por [Matthew Wonlaw](#) (tantaman), te permite diseñar presentaciones en impress.js y bespoke.js, con un mínimo esfuerzo. Dada su amigabilidad, no nos detendremos en mayores explicaciones, basta que observes el siguiente vídeo (ignora la última parte, relacionada con el Google Drive, pues esa utilidad fue eliminada).



Existen varias versiones, pero con la versión que compartimos en este libro es suficiente para el diseño expuesto en el vídeo.

Para insertar imágenes, estamos obligados (al menos, en esta versión) a usar un enlace de la web, pero sugerimos usar el editor para la configuración inicial (posiciones, rotaciones, estilos, etc.) y luego intervenir el resultado de acuerdo a lo explicado en los apartados anteriores.

He aquí el editor (puedes ir a la última versión en <http://strut.io/>). Recuerda guardar la presentación haciendo clic en impress.js o bespoke.js y luego usar las teclas ctrl + s.



¡Practica con el editor en pantalla completa!

¡Eso es todo!

