



Frontend Fundamentals

Interacción y animación

Hasta el momento has analizado cómo crear una estructura y diseños por medio de hojas de estilo; sin embargo, en la actualidad no basta con realizar un sitio bien estructurado, que, si bien el look and feel de un sitio tiene mucha relevancia, la misión es influir en la experiencia que el usuario tiene al entrar a un sitio web y esto podrás lograrlo gracias a la interactividad.

Martínez (2019) menciona que el diseño de la interactividad tiene como eje al usuario y todo girará alrededor de éste y a la forma en la que se relaciona con la computadora. En última instancia, el diseño de la interactividad debe responder a la pregunta ¿cómo debe funcionar un sitio web?

Planificar la interactividad es de suma importancia, ya que en este punto podrás guiar al usuario hacia los diversos contenidos del sitio o ayudarle a simplificar la búsqueda de la información que requiere. La interactividad podemos segmentarla en tres tipos:

1. Navegación.
2. Componentes interactivos.
3. Controles de interacción.



En temas anteriores se estudió sobre las barras de navegación y la creación de menús, así como los controles de interacción, como pueden ser los botones o enlaces activos dentro del contenido, sin embargo, en este tema te enfocarás en los componentes de interacción utilizando recursos del framework de Bootstrap.

Has llegado a una de las partes más interesantes del curso, que es la de empezar a activar el movimiento a los objetos mediante transiciones, animaciones y transformaciones. Estas funcionalidades pueden servir para atrapar la atención del usuario, para explicar puntos de manera impactante en la aplicación o como una solución para crear o enriquecer una interfaz de usuario.

Explicación

Componentes interactivos

Los componentes interactivos son los que llevan eventos que ocurren en cada pantalla de un sitio web, es decir, cada componente tiene una función importante, en donde su función principal es la de mostrar de cierta forma interactiva un contenido ya sea textual y/o imagen; su despliegue puede ser mediante animación, transición, clic, hover, etc.

Es importante mencionar que para que dichos componentes funcionen, en su mayoría son procesados por medio de JavaScript, en frameworks como Bootstrap; los complementos se pueden descargar individualmente usando el archivo de cada recurso por separado, por ejemplo: carousel.js o utilizar un servidor CDN (Content Delivery Network) para agregar todos los componentes a la vez usando bootstrap.js o bootstrap.min.js, de la siguiente forma:

Añadir bootstrap.bundle.min.js:

```
<script  
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/js/bootstrap.bundle.min.js"  
integrity="sha384-A3rJD856KowSb7dw1ZdYEk039Gagi7vIsF0jrRAoQmDKtQBHUuLZ9AsSv4jD4Xa"  
crossorigin="anonymous"></script>
```

Añadir bootstrap.min.js:

```
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/js/bootstrap.min.js"  
integrity="sha384-ODmDIVzN+pFdexxHEHFBQH3/9/vQ9uori45z4JjnFsRydbmQbmL5t1tQ0culUzyK"  
crossorigin="anonymous"></script>
```

Dichos JS, deben colocarse cerca del final de sus páginas, justo antes de la etiqueta de cierre </body>, para habilitarlas:

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Bootstrap demo</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-gH2yIJqKdNHPEq0n4Mqa/HGKIhSkIHeL5AyhkYV8i59U5AR6csBvApHHNl/vI1Bx" crossorigin="anonymous">
  </head>
  <body>

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/js/bootstrap.bundle.min.js" integrity="sha384-A3rJD856KowSb7dwZdYEk039Gagi7vIsF0jrRAoQmDKtQBHUuLZ9AsSv4jD4Xa" crossorigin="anonymous"></script>
  </body>
</html>
```

Estas pantallas se obtuvieron directamente del software que se está explicando en la computadora, para fines educativos.

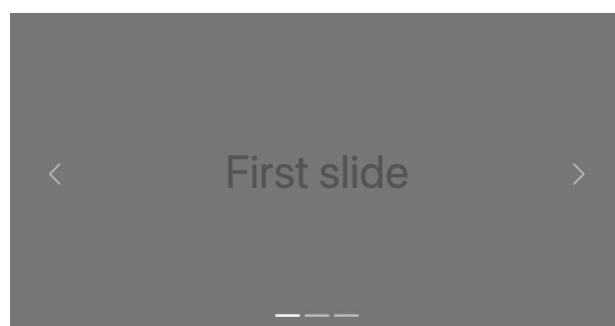
El código anterior es la base para iniciar un sitio con recursos interactivos.

A continuación, conocerás algunos componentes interactivos de Bootstrap:

Carrusel / Carousel

El carrusel es una herramienta importante para destacar un contenido, ya que un buen diseño o colocar imágenes atractivas, puede ser de gran impacto en un sitio web. Un carrusel es una presentación de diapositivas que se muestra una tras otra, por medio de transiciones CSS y un poco de JavaScript.

Ejemplo de carrusel:



Código:

```
<div id="carouselExampleControls" class="carousel slide" data-bs-ride="carousel">
  <div class="carousel-inner">
    <div class="carousel-item active">
      
    </div>
    <div class="carousel-item">
      
    </div>
    <div class="carousel-item">
      
    </div>
  </div>
  <button class="carousel-control-prev" type="button" data-bs-target="#carouselExampleControls" data-bs-slide="prev">
    <span class="carousel-control-prev-icon" aria-hidden="true"></span>
    <span class="visually-hidden">Previous</span>
  </button>
  <button class="carousel-control-next" type="button" data-bs-target="#carouselExampleControls" data-bs-slide="next">
    <span class="carousel-control-next-icon" aria-hidden="true"></span>
    <span class="visually-hidden">Next</span>
  </button>
</div>
```

Explicación:

1. El primer **div** requiere la utilización de un **id** (en el ejemplo `id="carouselExampleControls"`) para que los controles del carousel funcionen correctamente.

La clase **.carousel** indica que el `<div>` contiene un carousel.

La clase **.slide** añade una transición CSS en forma de animación, lo que hace que las diapositivas se “deslicen” cuando se pasa de una a otra. Si no deseas este efecto, omite esta clase.

El atributo **data-bs-ride =“carousel”** dice a Bootstrap que empiece a animar el carousel justo cuando se carga la página.

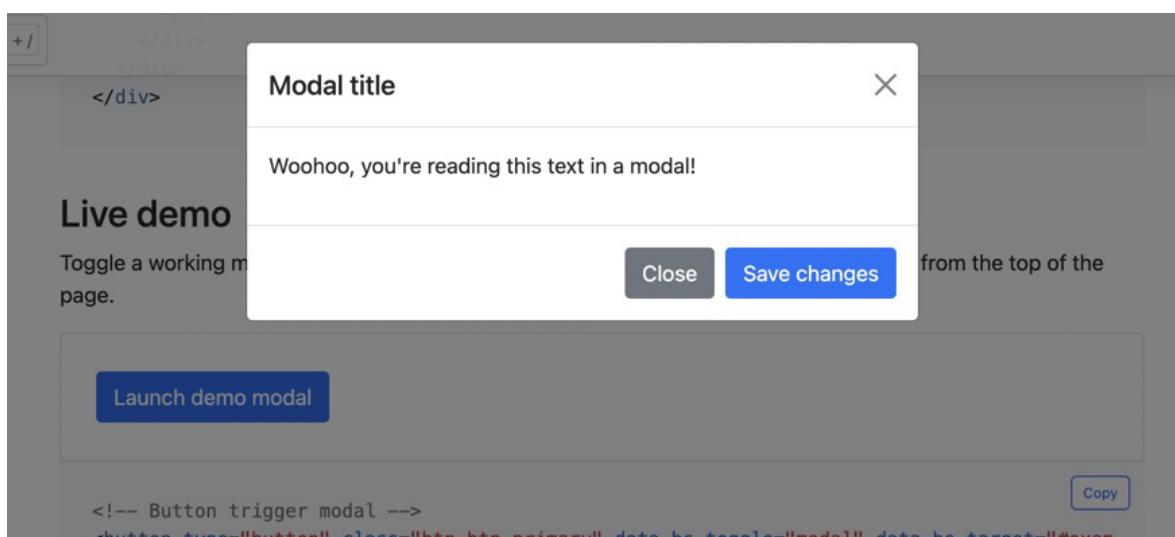
2. El segundo **div** con clase **.carousel-inner** es el que contiene todas las imágenes a mostrar.
3. Los divs que contienen las imágenes y/o textos, contienen la clase **.carousel-item** la cual indica que es un contenido de carousel y a alguno de estos, se le debe colocar la clase **.active**

1. Para que el carousel funcione de forma correcta debes colocar la clase `.active` a una de las diapositivas; de lo contrario, el carousel no será visible. En una página puedes colocar más de un carousel, sin embargo, debes asegurarte de colocar un identificador (id) diferente en cada uno para que los controles funcionen de forma correcta; recuerda revisar los elementos de control e identificadores, pues deben tener un atributo `data-bs-target` (o href para enlaces) que coincida con `id` del elemento `carousel`.

Modal

Es un complemento de JavaScript de Bootstrap para agregar cuadros de diálogo a tu sitio, es decir, contenidos ocultos que se muestran cuando el usuario presiona un botón o activador; los modales pueden ser usados para mostrar notificaciones o contenido completamente personalizado.

Ejemplo de modal:



Código:

```
<!-- Button trigger modal -->
<button type="button" class="btn btn-primary" data-bs-toggle="modal"
data-bs-target="#exampleModal">
    Launch demo modal
</button>

<!-- Modal -->
<div class="modal fade" id="exampleModal" tabindex="-1" aria-
labelledby="exampleModalLabel" aria-hidden="true">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title" id="exampleModalLabel">Modal title</h5>
                <button type="button" class="btn-close" data-bs-dismiss="modal" aria-
label="Close"></button>
            </div>
            <div class="modal-body">
                ...
            </div>
            <div class="modal-footer">
                <button type="button" class="btn btn-secondary" data-bs-
dismiss="modal">Close</button>
            </div>
        </div>
    </div>
</div>
```

Estas pantallas se obtuvieron directamente del software que se está explicando en la computadora, para fines educativos

Explicación:

1. El botón utiliza los atributos `data-bs-toggle="modal"` para identificar el botón como un componente modal y `data-bs-target="#exampleModal"` para abrir el modal con ID que debe ser único. Este mismo ID debe usarse en el componente modal.
2. El contenedor div debe tener una clase CSS `".modal"`. La clase adicional `".fade"` se utiliza para atenuar el cuadro modal, que es opcional.
3. El ID debe ser el mismo que el valor de ID configurado para el componente del botón.
4. El contenido modal debe agregarse dentro etiquetas identificadas con la clase `".modal-dialog"`
5. El div con la clase `.modal-content` es el que engloba el contenido del modal, el cual tiene una estructura de header, body y footer.

6. La sección de header se identifica con la clase **.modal-header** y tiene dos componentes: título y botón. El componente de título usa **".modal-title"** y se usa para agregar un título a la ventana modal. Los componentes del botón se utilizan para agregar el icono de cierre (X) para cerrar el cuadro de diálogo modal.

7. El cuerpo modal usa la clase **".modal-body"** y contiene el cuerpo del modal, aquí es donde podrás personalizar el contenido que queremos mostrar.

8. La sección de pie de página usa **".modal-footer"** y contiene un botón para cerrar la ventana modal.

Asegúrate de incluir los atributos como **tabindex**, **role** y **aria-hidden** y colocar un **ID** único a cada modal que incluyas en tu sitio.

Animaciones con CSS

Se puede definir que una animación CSS es la transición gradual de unos estilos a otros en los elementos del HTML. Es decir que, en una hoja de estilos en cada selector se pueden definir un punto de inicio y otro de finalización para realizar dicha transición gradual.

Hernán Beati (2020) menciona que “la creación de animaciones con CSS consiste en escribir código CSS en dos zonas distintas de una hoja de estilos, con dos intenciones bien diferenciadas”.

1. Crear estilos en nuestro CSS.
2. Aplicar los estilos a uno o más elementos HTML.

Para empezar a crear las animaciones, es necesario crear una regla especial llamada keyframes en la hoja de estilos CSS. Al igual que en otros programas y lenguajes, los keyframes son una manera de definir puntos en la animación de inicio y fin.

Es necesario conocer cuáles son las propiedades que podemos definir desde el CSS:

- **animation-name**: el nombre del guion que implementará el objeto, como una clase.
- **animation-duration**: es la duración total que tardará el objeto en reproducirse.
- **animation-interation-count**: determina la cantidad de veces que se repetirá.
- **animation-direction**: define qué sucederá cuando termine de reproducirse.
- **animation-timing-function**: nos permite declarar los efectos de aceleración y desaceleración.
- **animation-delay**: define el tiempo de espera en milisegundos.
- **animation-fill-mode**: controla lo que sucede de principio a fin de la animación.

Ejemplo:

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

```
<title>Documento sin título</title>
<style>
div {
    width: 100px;
    height: 100px;
    background-color: red;
    animation-name: example;
    animation-duration: 6s;
}
@keyframes example {
from {
background-color: red;
}
to {
background-color: yellow;
}
}
</style>
</head>
<body>
<h1>Animación con CSS</h1>
<div></div>
</body>
</html>
```

Estas pantallas se obtuvieron directamente del software que se está explicando en la computadora, para fines educativos.

Explicación:

Con el atributo **from** se define el punto de partida, en este caso va a tener un fondo rojo, y con el atributo **to** definimos el punto final, en este caso a color amarillo. El siguiente punto es definir a qué elemento o selector se agregará la animación por medio de **animation-name**, donde se define la animación que se quiere para este elemento del HTML y con **animation-duration** cuánto tiempo es necesario que dure la animación desde un estado a otro.

Por medio de las propiedades en CSS se puede animar cualquier número de elementos en HTML, por lo que es importante practicar para que te familiarices con cada una de ellas.

Si bien, ya se analizó cómo crear animaciones, así como el framework de Bootstrap, también existen librerías con animaciones predefinidas que pueden ayudar a realizar más rápido el proceso. Algunas librerías son:

- Animate.CSS
- Bounce.JS
- Animate.js

“No olvides que la animación es más un arte que una ciencia, por lo que es mejor experimentar y probar tus decisiones con los usuarios”, Skytskyi (2018).

Puedes acceder a los archivos .txt de código en el apartado “Archivos adjuntos”.

Tecmilenio no guarda relación alguna con las marcas mencionadas como ejemplo. Las marcas son propiedad de sus titulares conforme a la legislación aplicable, se utilizan con fines académicos y didácticos, por lo que no existen fines de lucro, relación publicitaria o de patrocinio.

La interactividad en las páginas web tiene como finalidad que el usuario sea un sujeto activo, que interactúe con un contenido lo más personalizado que se pueda y que le haga sentirse partícipe de toda la navegación durante el mayor tiempo posible.

Aunque la interactividad es importante, hay que ser cauteloso con su uso, ya que un sitio con demasiados recursos puede llegar a ser confuso o abrumador para el usuario, por lo que se recomienda planificar y estructurar los contenidos para obtener mejores resultados.

Al igual que la interacción, la animación debe tener ciertas consideraciones para su uso.

Skytskyi (2018) menciona que “la animación en la interfaz debe reflejar los movimientos que se conocen del mundo físico: fricción, aceleración, etc., imitando el comportamiento de los objetos del mundo real se puede crear una secuencia que permita a los usuarios comprender qué esperar de la interfaz”.



Referencias bibliográficas

- Beati, H. (2016). *HTML5 y CSS3 para diseñadores*. Argentina: Alfaomega.
- Martínez, X. (2019). *Diseño de páginas web: WordPress para todos los públicos*. España: Editorial UOC.
- Skytskyi, T. (2018). *The ultimate guide to proper use of animation in UX*. Recuperado de <https://uxdesign.cc/the-ultimate-guide-to-proper-use-of-animation-in-ux-10bd98614fa9>

Para saber más

Los siguientes enlaces son externos a la Universidad Tecmilenio, al acceder a ellos considera que debes apegarte a sus términos y condiciones.

Videos

Para conocer más acerca de **Carouseles**, te sugerimos revisar lo siguiente:

- Adrian Twarog. (2022, 27 de julio). *Carousel Slider Bootstrap 5 | Slider Carousel Tutorial* [Archivo de video]. Recuperado de https://www.youtube.com/watch?v=ku_97a6Bgkg

Para conocer más acerca de **Modales**, te sugerimos revisar lo siguiente:

- Código AV. (2021, 14 de abril). [2022] *Cómo hacer un modal con Bootstrap - Curso de Bootstrap* [Archivo de video]. Recuperado de <https://www.youtube.com/watch?v=D7ZvKM9WjXk>

Para conocer más acerca de **Animación CSS**, te sugerimos revisar lo siguiente:

- Gustavo Machado // Desarrollador web. (2020, 30 de octubre). *Como utilizar Animate.css fácil 2020* [Archivo de video]. Recuperado de <https://www.youtube.com/watch?v=1-b7td9mdjU>

Asegúrate de:

- Comprender la importancia de la interactividad en los sitios web.
- Comprender cómo funcionan los componentes interactivos.
- Conocer qué es la animación con CSS.

Requerimientos Técnicos

- Contar con una computadora con acceso a internet.
- Contar con un editor de código como Dreamweaver, Sublime Text, Brackets o Visual Studio.

Prework

Antes de iniciar con la actividad guiada, es importante dar un repaso a los temas anteriores, en especial revisar el tema de Frameworks.

La obra presentada es propiedad de ENSEÑANZA E INVESTIGACIÓN SUPERIOR A.C. (UNIVERSIDAD TECMILENIO), protegida por la Ley Federal de Derecho de Autor; la alteración o deformación de una obra, así como su reproducción, exhibición o ejecución pública sin el consentimiento de su autor y titular de los derechos correspondientes es constitutivo de un delito tipificado en la Ley Federal de Derechos de Autor, así como en las Leyes Internacionales de Derecho de Autor.

El uso de imágenes, fragmentos de videos, fragmentos de eventos culturales, programas y demás material que sea objeto de protección de los derechos de autor, es exclusivamente para fines educativos e informativos, y cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por UNIVERSIDAD TECMILENIO.

Queda prohibido copiar, reproducir, distribuir, publicar, transmitir, difundir, o en cualquier modo explotar cualquier parte de esta obra sin la autorización previa por escrito de UNIVERSIDAD TECMILENIO. Sin embargo, usted podrá bajar material a su computadora personal para uso exclusivamente personal o educacional y no comercial limitado a una copia por página. No se podrá remover o alterar de la copia ninguna leyenda de Derechos de Autor o la que manifieste la autoría del material.