

JavaScript (Parte II)

Añadiendo jQuery al proyecto Olivia Ros

Integrando jQuery

En los videos anteriores conocimos un poco sobre los conceptos básicos de JavaScript. Desde ahora comenzaremos a trabajar con jQuery.

¿Qué es jQuery?



Imagen 1. Logo jQuery.

jQuery es una biblioteca de JavaScript que nos permite escribir menos código y obtener los mismos resultados que con JavaScript puro. También hace más sencilla la captura y manipulación de los diferentes elementos de un sitio. Esto lo realiza encapsulando el código JavaScript en métodos más cortos y simples.

Además, jQuery es software libre y de código abierto, bajo una licencia MIT y GNU que permite su uso privado y comercial.

Características

La biblioteca de jQuery tiene las siguientes características:

- Manipulación del DOM/HTML.
- Manipulación de CSS.
- Métodos de eventos para HTML.
- Efectos y animaciones.
- Entre otras.

Formas de integrar jQuery

Para integrar jQuery a un proyecto tenemos dos opciones:

1. Bajar la biblioteca desde [jQuery](#), añadirlo dentro del directorio de nuestro proyecto y llamarlo desde la etiqueta `<script>` con el atributo `src`. Tal como llamamos a un archivo .js, ya que toda la biblioteca de jQuery está escrita en JavaScript.
2. La otra forma, y la que utilizaremos más a menudo, es usar un CDN desde [Code jQuery](#).

En nuestro proyecto Olivia Ros ya integramos jQuery junto con Popper JS y Bootstrap JS.

```
<!--jQuery 3.3.1-->
<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
integrity="sha384-q8i/X+965Dz00rT7abK41JStQIAqVgRVzbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous"></script>
<!--Popper JS 1.14.3-->
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"
integrity="sha384-ZMP7rVo3mIykV+2+9J3UJ46jBk0WLauAdn689aCwoqbBJiSnjAK/l8WvCWPIPm49"
crossorigin="anonymous"></script>
<!--Bootstrap JS 4.1.3-->
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"
integrity="sha384-ChfqxuZUCnJSK3+MXmPNIyE6ZbWh2IMqE241rYiqJxyMiz60W/JmZQ5stwEULTy"
crossorigin="anonymous"></script>
```

Lo que haremos entonces será cambiar jQuery slim por jQuery normal que sacaremos desde [Code jQuery](#). ¿Por qué? Porque jQuery slim pierde algunas propiedades de jQuery normal, como las animaciones, que necesitaremos para el desarrollo de la actividad. De todas formas copiaremos el CDN de jQuery minificado, para ahorrar tiempos de carga y peso del archivo.

```
<!--jQuery 3.3.1 (sin slim)-->
<script src="https://code.jquery.com/jquery-3.3.1.min.js" integrity="sha256-FgpCb/KJQlLNfOu91ta32o/NMZxtwRo8QtmkMRdAu8="
crossorigin="anonymous">
</script>
<!--Popper JS 1.14.3-->
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"
integrity="sha384-ZMP7rVo3mIykV+2+9J3UJ46jBk0WLauAdn689aCwoqbBJiSnjAK/l8WvCWPIPm49"
crossorigin="anonymous"></script>
<!--Bootstrap JS 4.1.3-->
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"
integrity="sha384-ChfqxuZUCnJSK3+MXmPNIyE6ZbWh2IMqE241rYiqJxyMiz60W/JmZQ5stwEULTy"
crossorigin="anonymous"></script>
```

Sintaxis jQuery

La sintaxis de jQuery está hecha para seleccionar elementos HTML y realizar alguna acción en el o los elementos seleccionados.

Su sintaxis básica es la siguiente:

```
$(selector).accion()
```

Es una buena práctica esperar a que el documento esté completamente cargado y listo antes de trabajar con él, es por eso que existe el **Document Ready Event**. Dentro de esa función debemos escribir todas las acciones que necesitemos ejecutar.

A través de las nuevas versiones esta función se ha simplificado hasta quedar así:

```
$(function(){
    // métodos de jQuery...
});
```

En los siguientes videos profundizaremos más en la sintaxis de jQuery a través de las distintas acciones que podemos hacer.

Selectores

Los selectores jQuery nos permiten seleccionar y manipular elementos HTML.

Los selectores jQuery se utilizan para "buscar" (o seleccionar) elementos HTML según su nombre, id, clases, tipos, atributos, valores de atributos y mucho más. Se basa en los selectores de CSS existentes y, además, tiene algunos selectores personalizados.

Todos los selectores en jQuery comienzan con el signo `$` y siguen con los paréntesis: `$()`.

Existen muchos tipos de selectores:

- Selector universal.
- Selector de elemento actual.
- Selector por etiqueta.
- Selector por id.
- Selector por clase.
- Selector por atributo.
- Selector por valor de atributo.
- Selector por número de elemento.
- Selector por elemento par.
- Selector por elemento impar.
- Entre muchos otros.

Puedes ver el [Selector tester de W3Schools](#), donde podrás conocer mejor los selectores.

Click a selector to see which element(s) gets selected in the result:

```
$(".lastname")
$("intro")
$(".intro, #lastname")
$("1")
$("p")
$("p:first")
$("p:last")
$("tr:even")
$("tr:odd")
$("td:only-child")
$("p:first-of-type")
$("p:last-of-type")
$("li:nth-child(1)")
$("li:nth-child(2)")
$("li:nth-of-type(2)")
$("li:nth-last-of-type(2)")
$("b:only-child")
$("h3:only-of-type")
$("div > p")
$("p")
$("ul + p")
$("ul ~ table")
$("ul li:even()")
$("ul li:odd()")
$("ul li:root")
$("header")
$("header:not(h1)")
$("animated")
$("focus")
$("contains(Duck)")
$("div:has(p)")
$("empty")
$("parent")
$("hidden")
$("tabVisible")
$("root")
$("p:lang(it)")
$("id")
$("myAddress")
$("id=Address")
$("id=ess")
 $("id=my")
 $("id^=l")
```

Result:

```
<h1> Welcome to My Homepage </h1>
<div class="intro">
<p>My name is Donald <span id="Lastname">Duck</span> </p>
<p id="my-Address">I live in Duckburg </p>
<p>I have many friends:</p>
</div>
<ul id="Listfriends">
  • <li>Goofy</li>
  • <li>Mickey</li>
  • <li>Daisy</li>
  • <li>Pluto</li>
</ul>
<p>I really like Daisy!! </p>
<p lang="it" title="Hello beautiful"> Ciao bella </p>
<h3> We are all animals! </h3>
<p> <b> My latest discoveries has led me to believe that we are all animals:</b> </p>
<table>
  <thead>
    <tr>
      <th> Name </th>
      <th> Type of Animal </th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td> Mickey </td>
      <td> Mouse </td>
    </tr>
    <tr>
      <td> Goofy </td>
      <td> Dog </td>
    </tr>
    <tr>
      <td> Daisy </td>
      <td> Duck </td>
    </tr>
    <tr>
      <td> Pluto </td>
      <td> Dog </td>
    </tr>
  </tbody>
</table>
```

Subscribe to my newsletter:

Name:

Imagen 2. Sección de Selectores.

Además de revisar esta [Referencia, también de W3Schools](#) donde enlistan los distintos selectores.

Consultando las referencias se te pueden hacer un poco más cercanos los selectores.

Lo importante es que selecciones bien el elemento al que le quieras dar la acción o atribuirle un método.

Eventos y Funciones

¿Qué son los eventos?

Las diferentes acciones que los visitantes pueden hacer dentro de una página web se denominan eventos.

Un evento representa el momento preciso en que algo sucede.

Ejemplos:

- Cuando el usuario clickea un elemento.
- Cuando el usuario presiona una tecla.
- Cuando el usuario hace scroll.
- Cuando el usuario cambia el tamaño de la ventana de visualización del navegador.

Aquí te presento algunos ejemplos de los eventos más usados:

Mouse	Teclado	Formularios	Documento/Ventana
click	keypress	submit	load
dblclick	keydown	change	resize
mouseenter	keyup	focus	scroll
mouseleave		blur	unload
hover			

Sintaxis de los eventos

La sintaxis es la siguiente:

```
$(selector).evento(function(){
    // La acción ocurre aquí
});
```

Como vemos es necesario pasarle una función al evento, que se ejecutará cuando el evento suceda.

Ejemplo:

```
$("#p").dblclick(function(){
    $(this).hide();
});
```

[Probemos](#)

Métodos

Los métodos son la acción a la que se ve enfrentada el selector.

```
$(selector).accion()
```

Son diferentes a los eventos.

Un evento es el resultado de una acción realizada por el usuario, como hacer click, desplazar, arrastrar, redimensionar, etc.

En cambio un método es la acción a la que se ve enfrentado. Éstos métodos pueden ser concatenados y afectar todos al selector. Muchas veces estos métodos son activados cuando un evento ocurre (como lo que hicimos en el ejercicio anterior).

Existen muchos métodos de jQuery, y podemos dividirlos en los siguientes tipos:

- Métodos de efectos.
- Métodos de manipulación de HTML y CSS.

Efectos

- `animate()`
- `clearQueue()`
- `delay()`
- `dequeue()`
- `fadeIn()`
- `fadeOut()`
- `fadeTo()`
- `fadeToggle()`
- `finish()`
- `hide()`
- `queue()`
- `show()`
- `slideDown()`
- `slideToggle()`
- `slideUp()`
- `stop()`
- `toggle()`

Manipulación de HTML y CSS

- `addClass()`
- `after()`
- `append()`
- `appendTo()`
- `attr()`
- `before()`
- `clone()`
- `css()`
- `detach()`
- `empty()`
- `hasClass()`

- `height();`
- `html();`
- `innerHeight();`
- `innerWidth();`
- `insertAfter();`
- `insertBefore();`
- `offset();`
- `offsetParent();`
- `outerHeight();`
- `outerWidth();`
- `position();`
- `prepend();`
- `prependTo();`
- `prop();`
- `remove();`
- `removeAttr();`
- `removeClass();`
- `removeProp();`
- `replaceAll();`
- `replaceWith();`
- `scrollLeft();`
- `scrollTop();`
- `text();`
- `toggleClass();`
- `unwrap();`
- `val();`
- `width();`
- `wrap();`
- `wrapAll();`
- `wrapInner();`

Probando

Vamos a probar algunos dentro del mismo [editor de W3Schools](#). Ocupando tanto los eventos como los métodos.

```
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
    $("p").dblclick(function(){
        $(this).hide();
    });
});
</script>
</head>
<body>

<p>Si me clickeas dos veces, desapareceré.</p>
```

```

<p>¡Yo también!</p>
<p>¡Yo también!</p>

</body>
</html>

```

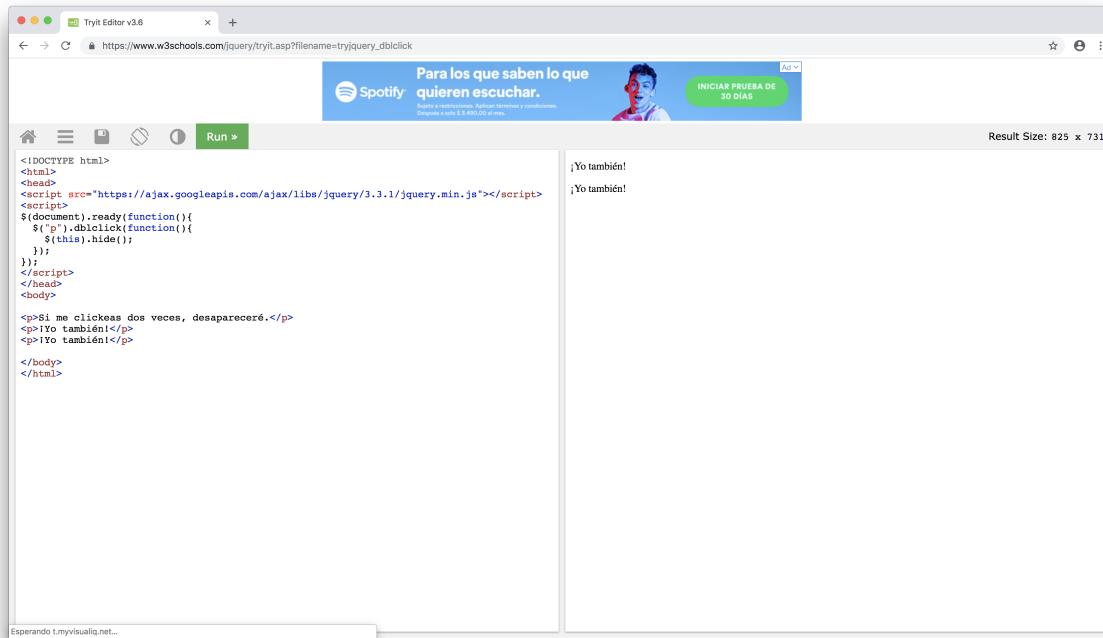


Imagen 3. Ejemplo con eventos y métodos, parte 1.

```

<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
    $("p").dblclick(function(){
        $(this).hide("slow");
    });
});
</script>
</head>
<body>

<p>Si me clickeas dos veces, desapareceré lentamente.</p>
<p>¡Yo también!</p>
<p>¡Yo también!</p>

</body>
</html>

```

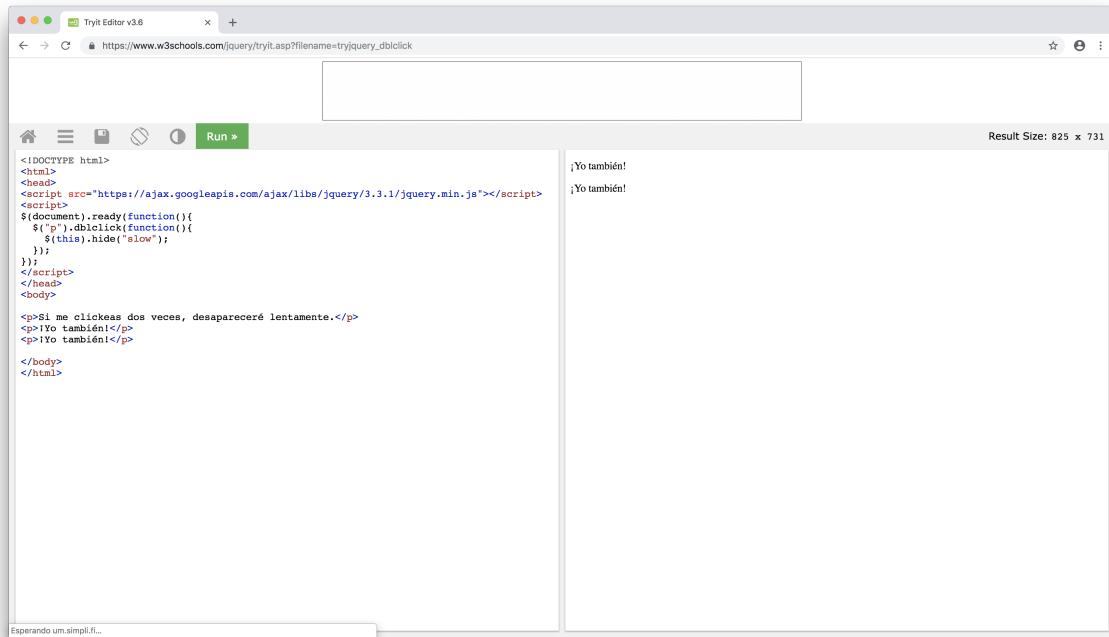


Imagen 4. Ejemplo con eventos y métodos, parte 2.

```
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
    $("p").dblclick(function(){
        $(this).css("color", "red");
    });
});
</script>
</head>
<body>

<p>Si me clickeas dos veces, me volveré rojo.</p>
<p>¡Yo también!</p>
<p>¡Yo también!</p>

</body>
</html>
```

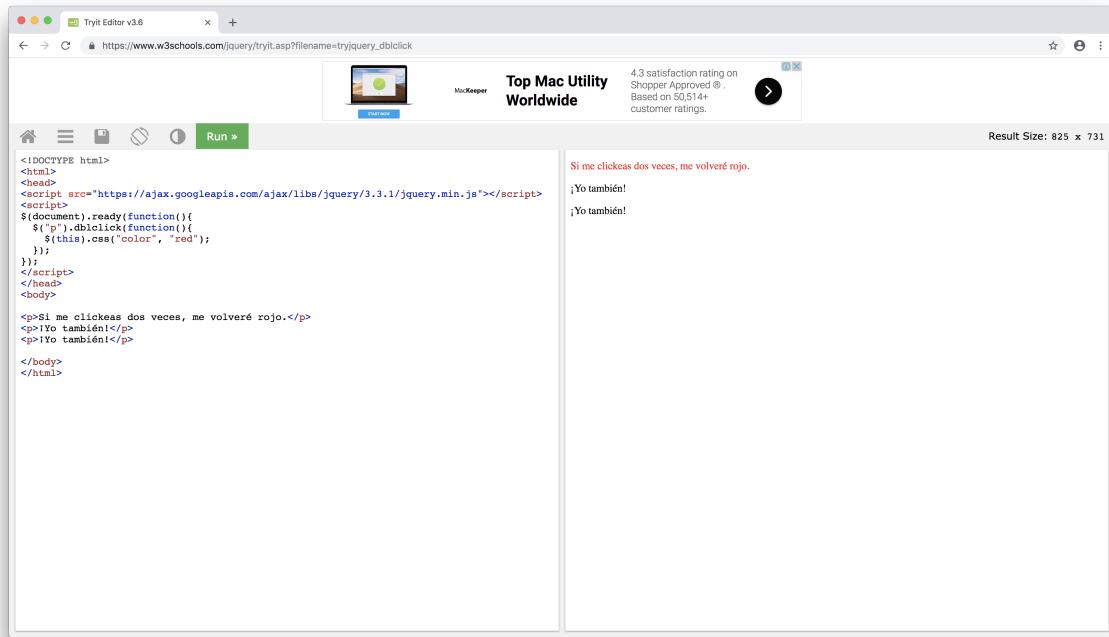


Imagen 5. Ejemplo con eventos y métodos, parte 3.

```

<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
    $("p").dblclick(function(){
        $(this).css({
            "color": "red",
            "font-size": "2em"
        });
    });
});
</script>
</head>
<body>

<p>Si me clickeas dos veces, me volveré rojo y aumentaré mi tamaño al doble.
</p>
<p>¡Yo también!</p>
<p>¡Yo también!</p>

</body>
</html>

```

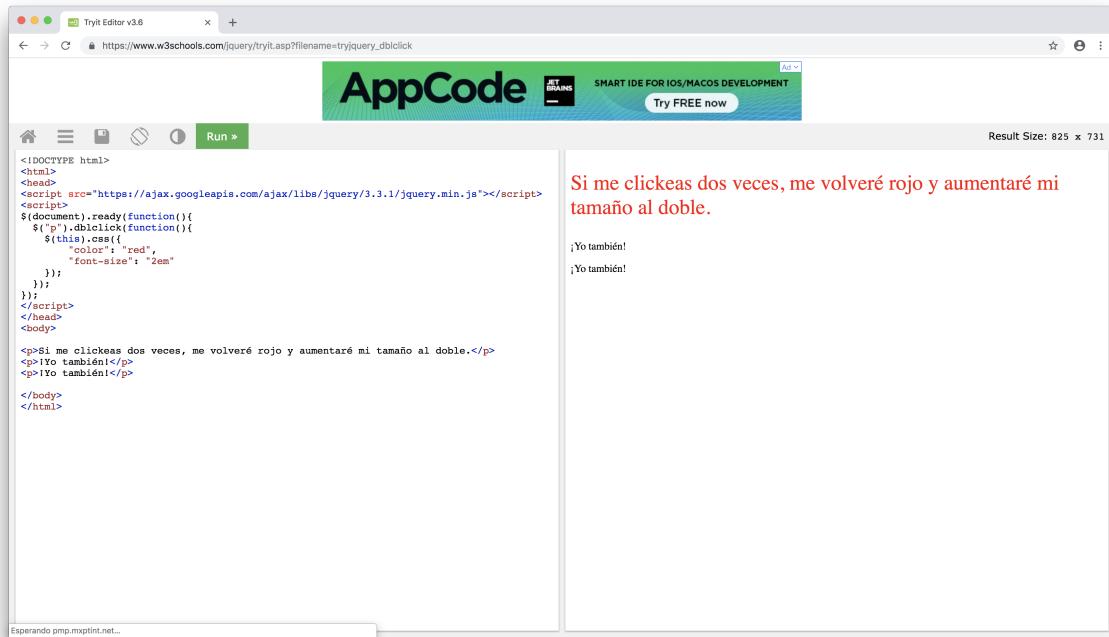


Imagen 6. Ejemplo con eventos y métodos, parte 4.

```
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
  $("p").dblclick(function(){
    $(this).html("<a href='#>Me convertí</a>");
  });
});
</script>
</head>
<body>

<p>Si me clickeas dos veces, me volveré un link.</p>
<p>¡Yo también!</p>
<p>¡Yo también!</p>

</body>
</html>
```

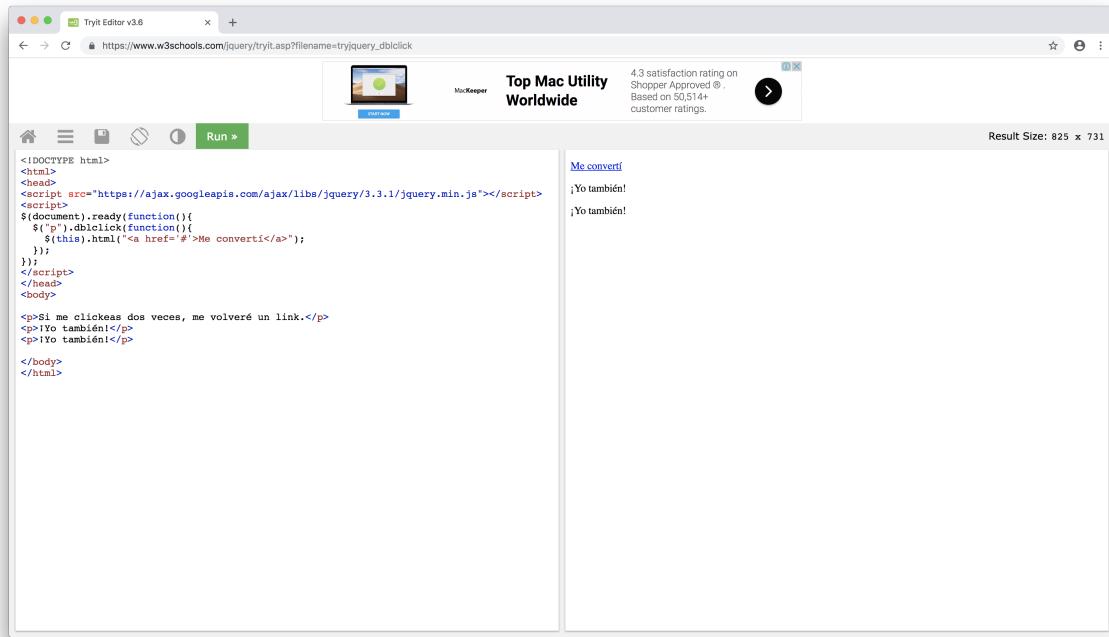


Imagen 7. Ejemplo con eventos y métodos, parte 5.

Te dejaremos una lectura con los métodos más utilizados y para qué sirven.

Smooth Scrolling

Llegó la hora de volver a nuestro proyecto de Olivia Ros.

Una vez que hayamos cambiado el CDN de jQuery slim por el CDN de jQuery normal podremos comenzar a realizar los diversos cambios.

En este video vamos a generar un smooth scrolling entre las secciones de la página principal del sitio Olivia Ros.

Lo primero que debemos hacer es generar las anclas por id. Eso lo haremos generando un id por las secciones correspondientes y vinculándolas con los ítems del menú.

```

<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Olivia Ros | Food Blogger</title>
    <!-- Bootstrap 4.1.3 CSS -->
    <link rel="stylesheet"
      href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"
      integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"
      crossorigin="anonymous">
      <!--Google Font: Raleway (Regular, Bold)-->
      <link href="https://fonts.googleapis.com/css?family=Raleway:400,700"
      rel="stylesheet">
      <!--Font Awesome 5.6.3-->

```

```

<link rel="stylesheet"
href="https://use.fontawesome.com/releases/v5.6.3/css/all.css"
integrity="sha384-UHrtZLI+pbxtHCWp1t77Bi1L4ZtiqrqD80Kn4Z8NTSRyMA2Fd33n5dQ8lWUE00s/"
crossorigin="anonymous">
    <!--Mi estilo-->
    <link rel="stylesheet" href="assets/css/style.css">
</head>
<body>
    <!--Navbar agregado-->
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark fixed-top">
        <div class="container">
            <a class="navbar-brand" href="#"></a>
            <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarNav">
                <ul class="navbar-nav ml-auto">
                    <li class="nav-item">
                        <a class="nav-link" href="#about">Acerca de mí</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="#work">Trabajos</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="#contact">Contacto</a>
                    </li>
                </ul>
            </div>
        </div>
    </nav>

    <header class="hero text-right text-light pt-5">
        <div class="container pt-5">
            <h1 class="display-1 font-weight-bold my-3">Comidas del fin del mundo</h1>
            <p class="my-3">Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.</p>
            <a class="btn btn-lg btn-outline-info mr-3" href="#work">Trabajos</a>
            <a class="btn btn-lg btn-info" href="#contact">Contacto</a>
        </div>
    </header>

    <section id="about" class="container text-center my-5">
        <h2 class="display-4 font-weight-bold my-3">Acerca de mí</h2>
        <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.</p>
        <div class="row">
            <div class="col-md-4">
                <i class="fas fa-3x fa-camera my-3"></i>
                <h3 class="my-3">Fotógrafa de comida</h3>
            </div>
        </div>
    </section>

```

```
<p class="my-3">Lorem ipsum dolor sit amet, consectetur adipisicing  
elit. Esse, eos perspiciatis qui consequuntur sint ducimus itaque inventore  
dolorem error dignissimos, illo magnam temporibus laborum voluptatem deserunt  
perferendis consectetur aut at.</p>  
</div>  
<div class="col-md-4">  
    <i class="fas fa-3x fa-bullhorn my-3"></i>  
    <h3 class="my-3">Critica culinaria</h3>  
    <p class="my-3">Lorem ipsum dolor sit amet, consectetur adipisicing  
elit. Provident libero reiciendis quod, quidem sequi ducimus dolore odio  
porro, quibusdam commodi, laborum ipsum! Voluptas velit cumque facilis  
blanditiis enim, maxime repellendus.</p>  
    </div>  
<div class="col-md-4">  
    <i class="far fa-3x fa-map my-3"></i>  
    <h3 class="my-3">Bloggera de viajes</h3>  
    <p class="my-3">Lorem ipsum dolor sit amet, consectetur adipisicing  
elit. Eum aliquid qui dolores veritatis quia facere aliquam cumque ex, maxime  
sunt incident, fugiat praesentium, ut obcaecati dignissimos voluptate.  
Cupiditate, et, vero.</p>  
    </div>  
</div>  
</section>  
  
<section id="work" class="container-fluid my-5 p-5 bg-info">  
    <h2 class="display-4 text-center font-weight-bold text-light mb-3">Mis  
trabajos</h2>  
    <div class="card-deck">  
        <div class="card">  
              
            <div class="card-body">  
                <h5 class="card-title">Asado en la Patagonia</h5>  
                <p class="card-text">Lorem ipsum dolor sit amet, consectetur  
adipisicing elit. Explicabo iure sint, perspiciatis distinctio molestias nihil  
repudiandae neque rem adipisci repellendus libero illo quo blanditiis, velit  
quasi ad, recusandae voluptates officia.</p>  
                <button type="button" class="btn btn-outline-info">Ver  
más</button>  
            </div>  
        </div>  
        <div class="card">  
              
            <div class="card-body">  
                <h5 class="card-title">Papas del fin del mundo</h5>  
                <p class="card-text">Lorem ipsum dolor sit amet, consectetur  
adipisicing elit. Deserunt a culpa doloremque voluptatem dicta nam. Maxime  
ipsum, consequatur quos error fuga excepturi quo. Laudantium optio pariatur  
laboriosam modi, ea molestias?</p>  
                <button type="button" class="btn btn-outline-info">Ver  
más</button>  
            </div>  
        </div>  
        <div class="card">  
              
            <div class="card-body">
```

```
        <h5 class="card-title">Salmones de Alaska</h5>
        <p class="card-text">Lorem ipsum dolor sit amet, consectetur
adipisicing elit. Obcaecati facere quibusdam sint id laborum magni velit
ducimus corporis facilis qui architecto dolor, accusamus eveniet
exercitationem fugit. Quod autem neque ducimus.</p>
        <button type="button" class="btn btn-outline-info">Ver
más</button>
    </div>
</div>
</div>
</section>

<section id="contact" class="container my-5">
    <h2 class="display-4 text-center font-weight-bold my-3">Contáctame</h2>
    <p class="text-center">Lorem ipsum dolor sit amet, consectetur
adipisicing elit. Dolores harum eaque, hic quod rerum officiis. Nemo veritatis
minima illum at quaerat expedita! Accusantium, iste. Harum corporis
accusantium repudiandae consequuntur, magni?</p>
    <form>
        <div class="form-group">
            <label for="formGroupExampleInput" class="font-weight-
bold">Nombre</label>
            <input type="text" class="form-control" id="formGroupExampleInput"
placeholder="Ingresa tu nombre">
        </div>
        <div class="form-group">
            <label for="exampleInputEmail1" class="font-weight-bold">Correo
electrónico</label>
            <input type="email" class="form-control" id="exampleInputEmail1"
aria-describedby="emailHelp" placeholder="Ingresa tu correo elecrtrónico">
        </div>
        <div class="form-group">
            <label for="exampleFormControlTextarea1" class="font-weight-
bold">Mensaje</label>
            <textarea class="form-control" id="exampleFormControlTextarea1"
rows="3" placeholder="Escribe tu mensaje"></textarea>
        </div>
        <div class="contact-button-inner text-right">
            <button type="button" class="btn btn-lg btn-info">Enviar</button>
        </div>
    </form>
</section>
<!-- Footer -->
<footer class="bg-dark text-center text-light p-3">
    <p>Olivia Ros 2019. Todos los derechos reservados.</p>
</footer>

<!--jQuery 3.3.1 (sin slim)-->
<script src="https://code.jquery.com/jquery-3.3.1.min.js"
integrity="sha256-FgpCb/KJQlLNfOu91ta32o/NMZxltwRo8QtmkMRdAu8="
crossorigin="anonymous"></script>
<!--Popper JS 1.14.3-->
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"
integrity="sha384-ZMP7rVo3mIykV+2+9J3UJ46jBk0WLauAdn689aCwoqbBJiSnjAK/l8WvCWPIPm49"
crossorigin="anonymous"></script>
<!--Bootstrap JS 4.1.3-->
```

```

<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"
integrity="sha384-ChfqqxuZUCnJSK3+MXmPNIyE6ZbWh2IMqE241rYiqJxyMiZ6OW/JmZQ5stwEULTy"
crossorigin="anonymous"></script>
</body>
</html>

```

Luego, generaremos la carpeta `js` dentro de la carpeta `assets` y ahí crearemos el archivo `script.js`. A este archivo lo llamaremos desde el final del `<body>`, justo después del llamado al Bootstrap JS .

```

<!--jQuery 3.3.1 (sin slim)-->
<script src="https://code.jquery.com/jquery-3.3.1.min.js" integrity="sha256-FgpCb/KJQlLNfOu91ta32o/NMZxltwRo8QtmkMRdAu8=" crossorigin="anonymous">
</script>
<!--Popper JS 1.14.3-->
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"
integrity="sha384-ZMP7rVo3mIykV+2+9J3UJ46jBk0WLauAdn689aCwoqbBJiSnjAK/l8WvCWPIPm49"
crossorigin="anonymous"></script>
<!--Bootstrap JS 4.1.3-->
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"
integrity="sha384-ChfqqxuZUCnJSK3+MXmPNIyE6ZbWh2IMqE241rYiqJxyMiZ6OW/JmZQ5stwEULTy"
crossorigin="anonymous"></script>
<!-- Mi script -->
<script src="assets/js/script.js"></script>

```

Dentro del archivo `script.js` vamos a declarar la función de document Document Ready Event. Recordemos que con las nuevas versiones la función se escribe así:

```

$(function(){
    // métodos de jQuery...
});

```

Dentro de la función vamos a seleccionar a los links `<a>` con el selector de etiqueta `$("a")` .

El evento que desencadenará la función será el evento `click` , o sea, cuando el usuario clickee la etiqueta `<a>` .

```

$(function(){
    $("a").click(function(event) {
        });
    });

```

Ahora debemos evaluar una condicional.

Para ello utilizaremos el método `hash`, que evalúa el atributo `href` de esto, y obtiene la parte de la URL que comienza con `#`.

Hagamos una prueba sólo para revisar. Cuando clickeemos un link nos aparecerá una alerta con lo que sigue después del `#` en el atributo `href`.

```
$(function(){
    $("a").click(function(event) {
        alert(this.hash)
    });
});
```

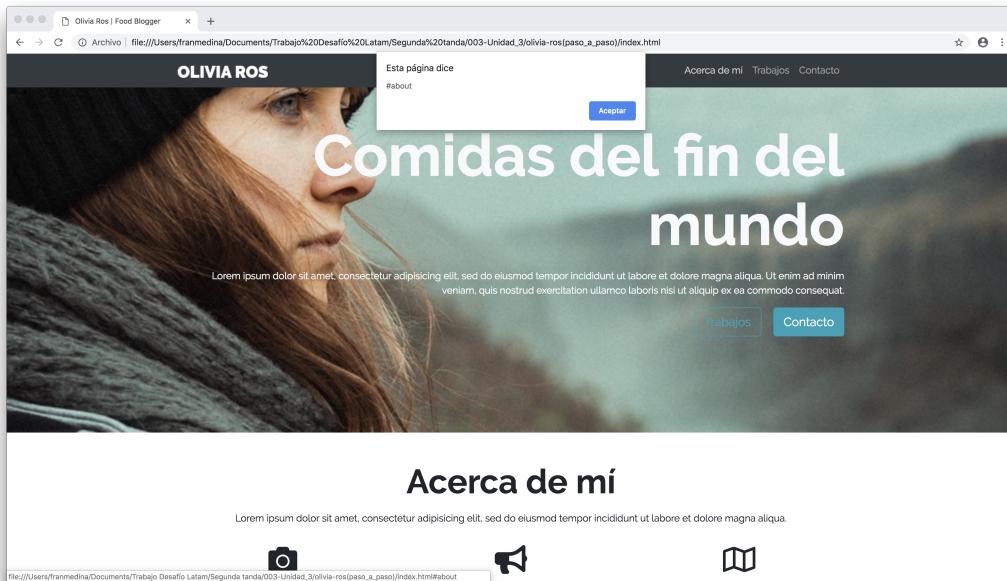


Imagen 8. Primera prueba.

Ya los tenemos identificados.

Ahora escribiremos lo siguiente

```
$(function(){
    $("a").click(function(event) {
        // Con este if se asegura que this.hash tenga un valor antes de anular el
        // comportamiento predeterminado
        if (this.hash !== "") {
            // Previene el comportamiento de click predeterminado
            event.preventDefault();
            // Guarda el valor del hash en una variable llamada gato
            var gato = this.hash;
            // Usa el método animate para animar el scroll y hacerlo de una forma
            // suave
            // El número opcional (800) especifica el número de milisegundos que se
            // demorará en llegar hasta el área
            $('html, body').animate({
                scrollTop: $(gato).offset().top
            }, 800, function(){

```

```
// Agrega hash (#) a la URL cuando haya terminado de desplazarse
// (comportamiento de click predeterminado)
    window.location.hash = gato;
);
} // Fin del if
);

});

});
```

Ahora guardamos y recargamos el navegador y podemos ver que se genera el scroll suave entre secciones.

Bootstrap JS: Popover, Modal, Tooltip y Dropdown

El Framework Bootstrap, contiene muchos elementos visuales que funcionan con JavaScript, específicamente necesitan jQuery, Popper JS y su propio archivo `.js` (Bootstrap JS) para funcionar.

Cuando elegimos algún elemento de Bootstrap lo primero que tenemos que hacer es ir a su [documentación](#). Una vez dentro de ella debemos verificar que la versión sea la que utilizaremos, esto lo podemos verificar en la zona superior.

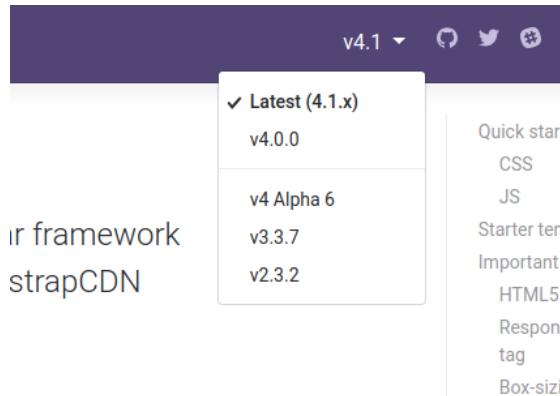


Imagen 9. Trabajando con bootstrap.

Siempre es recomendable copiar el código que aparece en la documentación ya que este se actualiza de forma constante, en base a la versión seleccionada.

Ahora conoceremos algunos elementos de Bootstrap. Algunos de ellos los implementaremos en las siguientes unidades.

Popover

Un `popover` es un elemento visual que se utiliza para mostrar un mensaje emergente, con una referencia específica a un tema.

Podremos verlo en acción en el siguiente enlace [documentación](#).

Si queremos añadirlo en nuestro proyecto solo debemos seguir las instrucciones.

```
$(function () {
  $('[data-toggle="popover"]').popover()
})
```

Esto añadirá un tooltip a cada elemento que tenga el atributo `data-toggle="popover"`.

Example

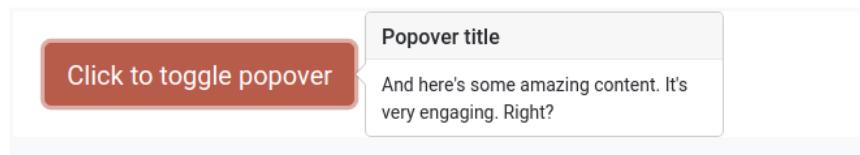


Imagen 10. Ejemplo tooltip.

Modal

Un modal es otro elemento importante que podemos disfrutar gracias a las características de Bootstrap. Sirve para añadir diálogos o mostrar notificaciones.

Podemos ver su efecto en la siguiente [documentación](#).

Podremos observar su funcionamiento en la documentación de Bootstrap, en el botón `Launch demo modal`.

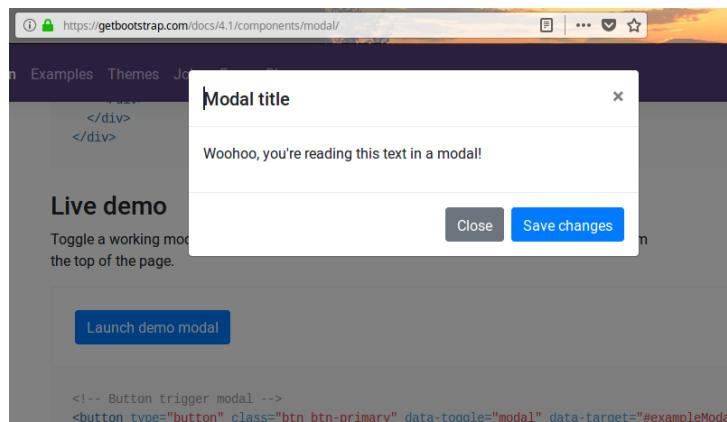


Imagen 11. Funcionamiento de Modal.

Tooltip

Un tooltip es una herramienta aparentemente similar al `popover` pero que tiene un concepto semántico diferente. El tooltip solo mostrará una ayuda cuando pasemos el puntero sobre la palabra o botón donde se haya añadido. Como su nombre lo dice, es una pista respecto al funcionamiento del elemento en particular.

Si accedes a su [documentación](#) podrás ver la forma de implementarlo en tu proyecto.

```
$(function () {
  $('[data-toggle="tooltip"]').tooltip()
})
```

Este código añade un tooltip a cada elemento que tenga el atributo `data-toggle="tooltip"`, entonces si lo añadimos dentro de algún elemento, por ejemplo un botón.

```

<button type="button" class="btn btn-secondary" data-toggle="tooltip" data-placement="top" title="Tooltip on top">
    Tooltip on top
</button>

```

Podremos ver el funcionamiento de esto en el ejemplo de la documentación, si presionamos el botón `Click to toggle popover`.



Imagen 12. Funcionamiento de tooltip.

Dropdown

De acuerdo a la [documentación](#), un dropdown servirá para mostrar un listado desplegable con opciones o links. Existen varias formas de utilizarlo. La mas simple es la siguiente:

```

<div class="dropdown">
    <button class="btn btn-secondary dropdown-toggle" type="button"
    id="dropdownMenuButton" data-toggle="dropdown" aria-haspopup="true" aria-
    expanded="false">
        Dropdown button
    </button>
    <div class="dropdown-menu" aria-labelledby="dropdownMenuButton">
        <a class="dropdown-item" href="#">Action</a>
        <a class="dropdown-item" href="#">Another action</a>
        <a class="dropdown-item" href="#">Something else here</a>
    </div>
</div>

```

Lo que incluirá un `dropdown button` que mostrará los ítems (Action, Another action, Something else here).

Basta solo implementar este código y nuestro **Dropdown**, comenzara a funcionar, ya que nuestro Bootstrap, utiliza un llamado via `Data attributes`. De todas formas, también podemos llamarlo desde JS añadiendo lo siguiente en nuestro archivo de configuración.

```
$( '.dropdown-toggle' ).dropdown()
```

Lo que asignara, la función `dropdown` con la convención `data-toogle`. Si lo añadimos de esta forma, tendremos mas control sobre su funcionamiento.

Conclusión

Como hemos visto, existen diferentes plugins ya incorporados dentro de nuestro querido Bootstrap. En las siguientes unidades integraremos algunas de ellas a nuestro proyecto para darle interactividad.

Implementando modales

Vamos a implementar ventanas modales al proyecto de Olivia Ros.

Tiene que sentido que en la sección de "Mis trabajos", cada botón de cada card genere una ventana modal con mayor información de ese trabajo en específico. Por lo tanto el botón será el que abrirá la ventana modal.

Primero lo primero. Leer la [documentación](#) de las ventanas modales.

Según lo que leemos aquí a un botón debemos añadirle los atributos `data-toggle="modal" data-target="#exampleModal"`. Tenemos que reemplazar el valor `exampleModal` por el `id` que le demos al div contenedor de la ventana modal.

Como haremos una por cada trabajo nos convendría ponerle un `id` relacionado al plato que se quiere ver en mayor profundidad. El primero es "Asado en la Patagonia" así que le pondremos `asado` como `id`.

Por lo tanto al botón le daremos esos atributos: `<button type="button" class="btn btn-outline-info" data-toggle="modal" data-target="#asado">Ver más</button>`.

```
<div class="card">
  
  <div class="card-body">
    <h5 class="card-title">Asado en la Patagonia</h5>
    <p class="card-text">Lorem ipsum dolor sit amet, consectetur adipisicing elit. Explicabo iure sint, perspiciatis distinctio molestias nihil repudiandae neque rem adipisci repellendus libero illo quo blanditiis, velit quasi ad, recusandae voluptates officia.</p>
    <button type="button" class="btn btn-outline-info" data-toggle="modal" data-target="#asado">Ver más</button>
  </div>
</div>
```

Y luego escribiremos la modal fuera de la sección:

```
<div class="modal fade" id="asado" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel" aria-hidden="true">
  <div class="modal-dialog" role="document">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="exampleModalLabel">Modal title</h5>
        <button type="button" class="close" data-dismiss="modal" aria-label="Close">
          <span aria-hidden="true">&times;</span>
        </button>
      </div>
      <div class="modal-body">
        ...
      </div>
      <div class="modal-footer">
```

```

        <button type="button" class="btn btn-secondary" data-
dismiss="modal">Close</button>
        <button type="button" class="btn btn-primary">Save changes</button>
    </div>
</div>
</div>
</div>

```

Ojo que debemos reemplazar su id de `exampleModal` por asado.

Probemos qué sucede:

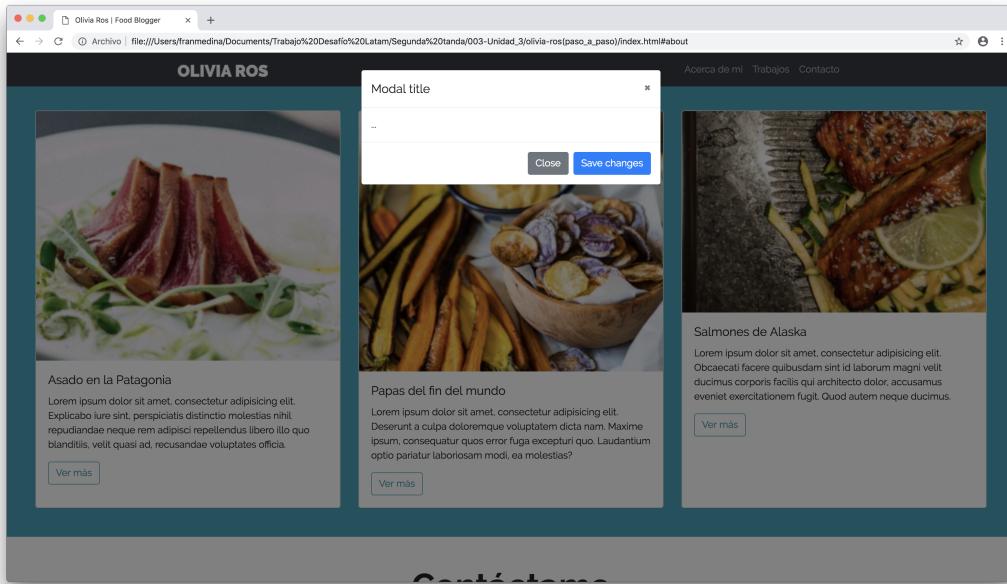


Imagen 13. Modificando Modal Title.

Ahora sólo falta rellenarla con contenido.

Modal title lo reemplazaremos por el título de la card: Asado en la Patagonia.

Dentro de `<div class="modal-body">...</div>` colocaremos la imagen correspondiente con la clase `img-fluid` para que ocupe el ancho del contenedor y la clase `my-3` para que genere espacio vertical.

Luego, también dentro de `<div class="modal-body">...</div>` Colocaremos 3 párrafos con Lorem Impsum.

Además le sacaremos el botón de `Save changes`.

```

<div class="modal fade" id="asado" tabindex="-1" role="dialog" aria-
labelledby="exampleModalLabel" aria-hidden="true">
    <div class="modal-dialog" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title" id="exampleModalLabel">Asado en la
Patagonia</h5>
                <button type="button" class="close" data-dismiss="modal" aria-
label="Close">
                    <span aria-hidden="true">&times;</span>
                </button>
            </div>

```

```

<div class="modal-body">
    
    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do
eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim
veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea
commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit
esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat
non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
</p>
    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do
eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim
veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea
commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit
esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat
non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
</p>
    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do
eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim
veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea
commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit
esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat
non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
</p>
</div>
<div class="modal-footer">
    <button type="button" class="btn btn-secondary" data-
dismiss="modal">Close</button>
</div>
</div>
</div>

```

Guardemos y recarguemos el navegador.

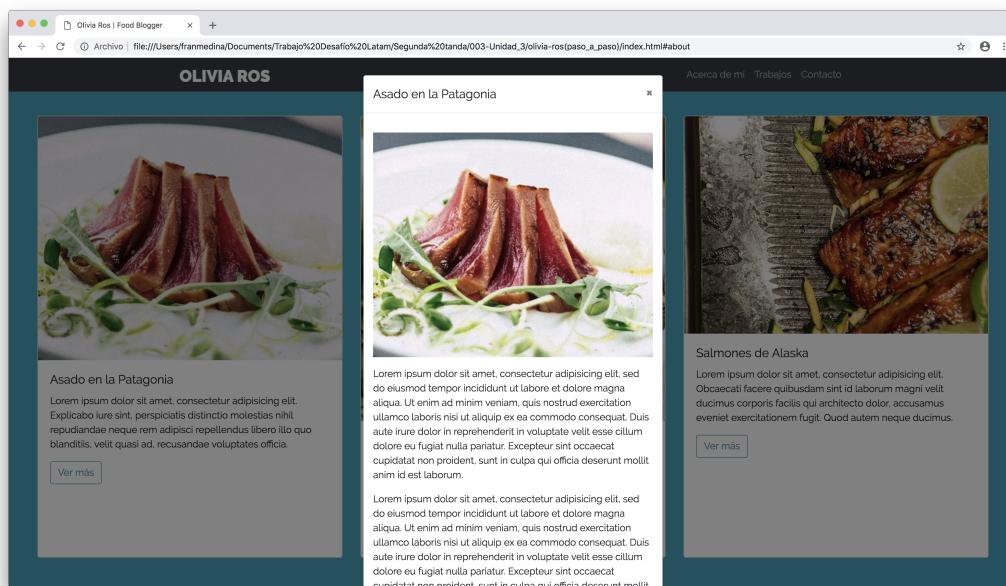


Imagen 14. Modal completo.

¿Te animas a generar las otras modales?

Integrando carousel de Bootstrap

Ahora integraremos un carousel de Bootstrap.

Un carousel es componente de presentación de diapositivas para recorrer los elementos (imágenes o diapositivas de texto). Para hacerlo iremos a la [documentación](#).

Queremos lograr un carousel así:



Imagen 15. Ejemplo de carousel.

Utilizaremos los íconos de Font Awesome que te dejaré en la guía:

- `fab fa-codepen`
- `fab fa-apper`
- `fab fa-angrycreative`
- `fab fa-google`
- `fab fa-500px`
- `fab fa-apple`
- `fab fa-aviato`
- `fab fa-d-and-d`

Y a esos íconos le daremos la clase `fa-2x` para agrandarlos.

Comencemos generando el código.

Este código lo colocaremos entre el `<header>` y la sección de Acerca de mí.

Podemos identificar una divisor con clase `containter` que contendrá la frase "Me has visto en:" y el carousel. A ese divisor también le daremos la clase `my-5 py-3` para generar márgenes y paddings verticalmente.

Generaremos un row donde pondremos dos divs con sus respectivas columnas.

Una será ocupando 4 columnas (la del texto "Me has visto en:") y la segunda será ocupando 8 columnas (carousel).

Al primer div le pondremos la clase `text-right` para alinear el texto a la derecha y `p-4` para generar paddings a todos lados. Además de un `<h2 class="font-weight-bold">Me has visto en:</h2>` como contenido.

Al segundo div le pondremos, aparte de la clase para asignarle el tamaño de columnas, las clases: `bg-secondary text-white rounded pt-4 pb-5`, para generar el color de fondo, las letras blancas, el borde redondeado y los padding top y bottom.

```

<div class="container my-5 py-3">
  <div class="row">
    <div class="col-4 p-4 text-right">
      <h2 class="font-weight-bold">Me has visto en:</h2>
    </div>

    <div class="col-8 bg-secondary text-white rounded pt-4 pb-5">
    </div>
  </div>
</div>

```

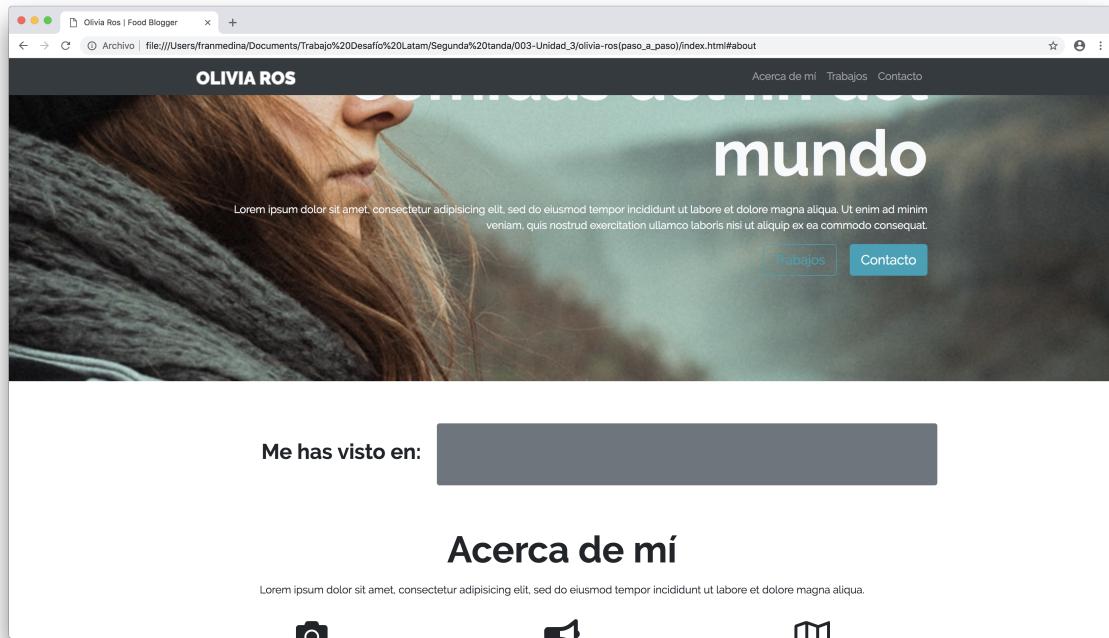


Imagen 16. Carousel incorporado, parte 1.

Ahora dentro del div de 8 columnas pegaremos el siguiente código que analizaremos poco a poco:

```

<div id="blogCarousel" class="carousel slide" data-ride="carousel">

  <ol class="carousel-indicators">
    <li data-target="#blogCarousel" data-slide-to="0" class="active"></li>
    <li data-target="#blogCarousel" data-slide-to="1"></li>
  </ol>

  <div class="carousel-inner">

    <div class="carousel-item active text-center">
      <div class="row">
        <div class="col-3">
          <i class="fab fa-codepen fa-2x"></i>
        </div>
        <div class="col-3">
          <i class="fab fa-apper fa-2x"></i>
        </div>
        <div class="col-3">
          <i class="fab fa-angrycreative fa-2x"></i>
        </div>
      </div>
    </div>
  </div>

```

```

<div class="col-3">
    <i class="fab fa-google fa-2x"></i>
</div>
</div>

<div class="carousel-item text-center">
    <div class="row">
        <div class="col-3">
            <i class="fab fa-500px fa-2x"></i>
        </div>
        <div class="col-3">
            <i class="fab fa-apple fa-2x"></i>
        </div>
        <div class="col-3">
            <i class="fab fa-aviato fa-2x"></i>
        </div>
        <div class="col-3">
            <i class="fab fa-d-and-d fa-2x"></i>
        </div>
        <div class="col-3">
            <i class="fab fa-d-and-d fa-2x"></i>
        </div>
    </div>
</div>
</div>

```

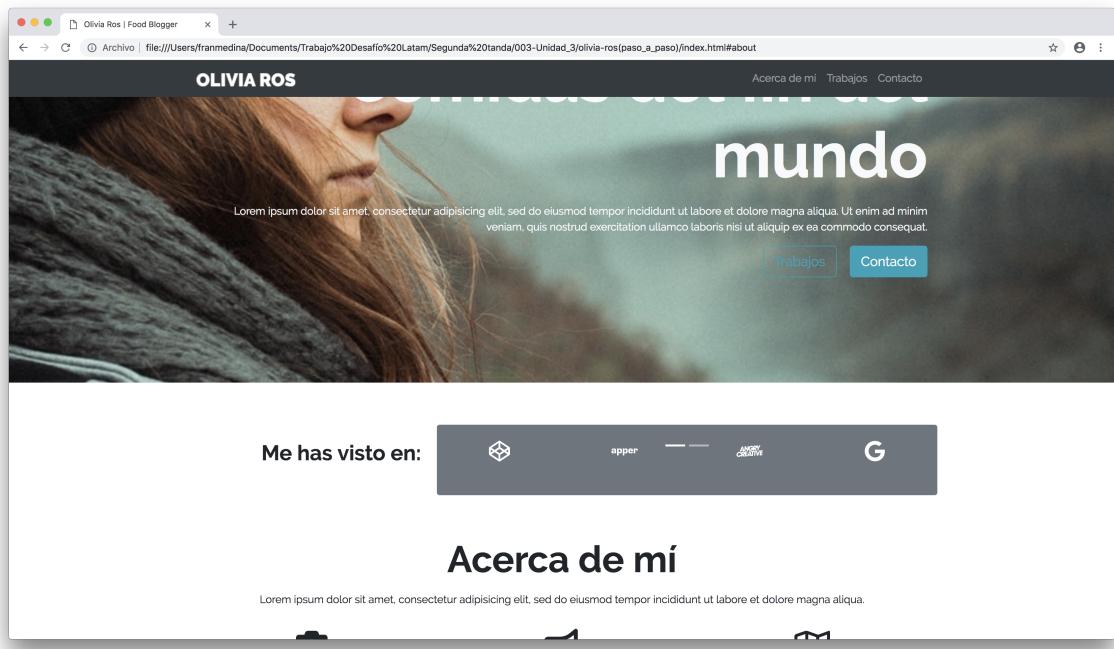


Imagen 17. Carousel incorporado, parte 2.

Primero lo que hicimos fue crear el `<div id="blogCarousel" class="carousel slide" data-ride="carousel">`. Esto genera un carousel.

Dentro de ese div generamos una lista ordenada contenedora de los indicadores `<ol class="carousel-indicators">`, que son esas líneas para cambiar el ítem del carousel.

Éstas parten del cero. Como tenemos dos ítems del carousel escribimos: `<li data-target="#blogCarousel" data-slide-to="0" class="active">` y `<li data-target="#blogCarousel" data-slide-to="1">` en representación de los dos ítems del carousel. La primera, o sea, la `0` tiene la clase `active`, que corresponde a ser la primera activa.

Después generamos un `<div class="carousel-inner">` donde estarán los ítems del carousel. Ahí generamos dos `<div class="carousel-item text-center">`, les asignamos el `text-center` para centrar los íconos y el primero tiene la clase `active` por ser el primero y el que debería ser mostrado en un primer lugar.

Dentro de este `carouse-item` generamos un `<div class="row">` con 4 divs e su interior con clase `col-3`. Eso es para usar 3 columnas y por ende poder hacer 4 contenedores. Dentro de cada uno se colocó la etiqueta `<i>` con su clase correspondiente de cada ícono.

Y eso sería a grandes rasgos.

Mirando el producto final nos faltaría cambiar la forma de los indicadores y ponerlos por bajo los íconos.

Para ello usaremos CSS.

```
.carousel-indicators {  
    top: 50px;  
}  
  
.carousel-indicators li {  
    width: 10px;  
    height: 10px;  
    border-radius: 50%;  
}
```

Ahora guardemos y recarguemos:

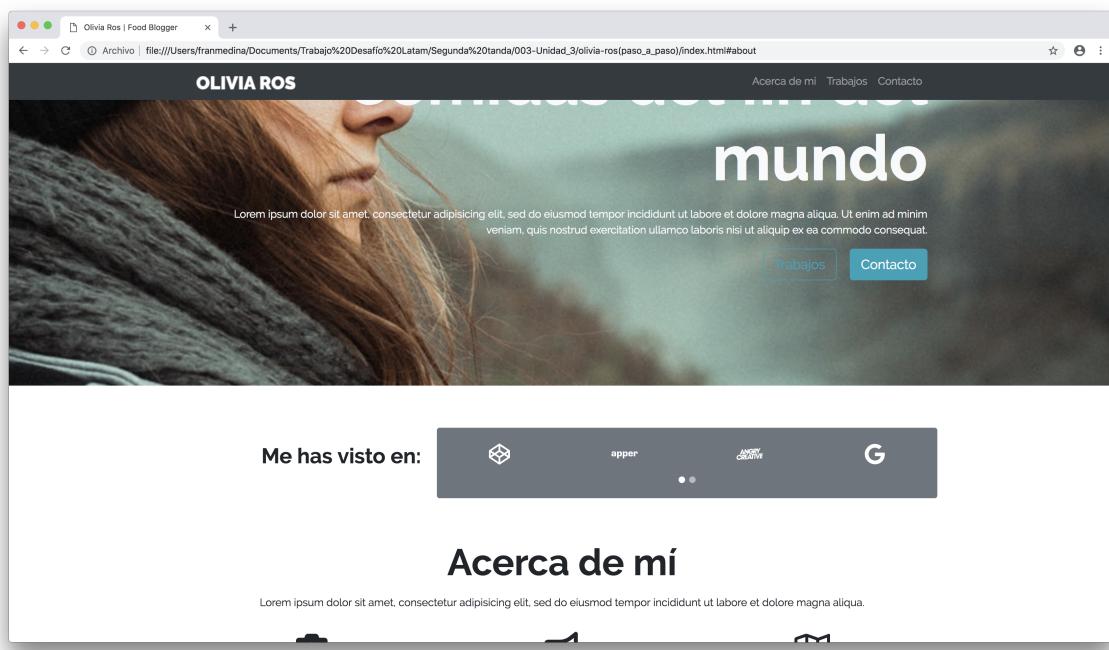


Imagen 18. Carousel incorporado, parte 3.

Implementando popovers

Le implementaremos popovers a cada ícono. Para ello debemos leer la [documentación](#).

Lo primero que debemos hacer es colocar el script que piden en la documentación.

Lo pondremos dentro de nuestro `script.js`:

```
$('[data-toggle="popover"]').popover();
```

Como ya tenemos la función de **Document Ready Event** no necesitamos volver a colocarla.

Ahora copiaremos los atributos necesarios para generar un popover que se coloque bajo el elemento `data-container="body" data-toggle="popover" data-placement="bottom" data-content="Vivamus sagittis lacus vel augue laoreet rutrum faucibus."`. Reemplazaremos el valor del `data-content` por el nombre de la marca.

```
<i class="fab fa-codepen fa-2x" data-container="body" data-toggle="popover" data-placement="bottom" data-content="Codepen."></i>
```

Veamos si funciona:

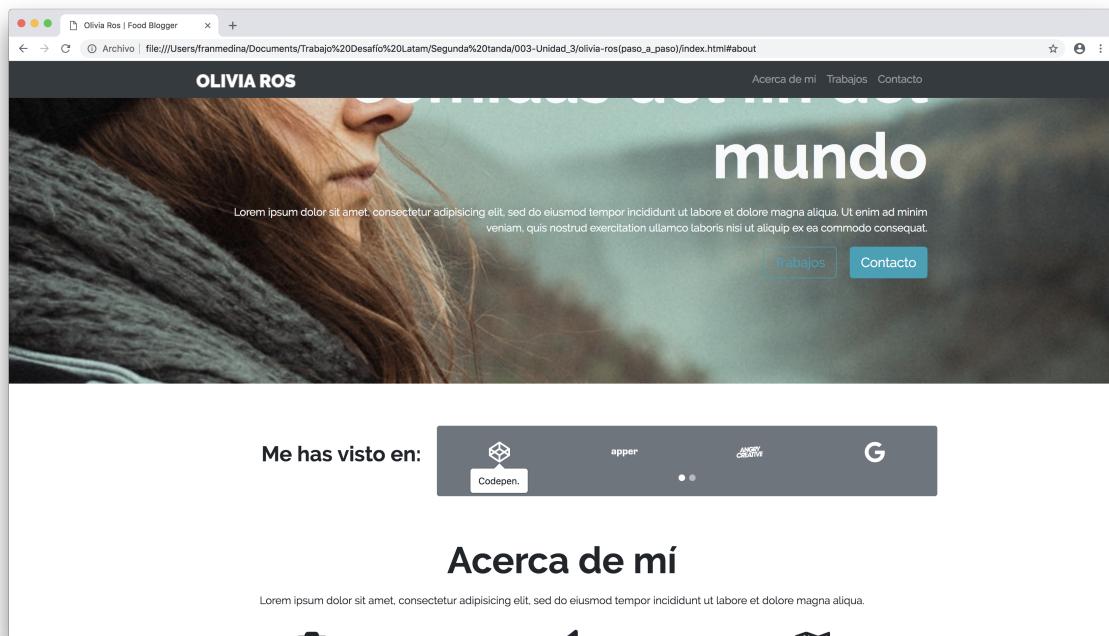


Imagen 19. Implementando popovers.

Ahora le agregaremos el atributo `data-trigger="hover"` para que aparezca al hacerle `hover`, o sea, al pasar el mouse por encima y no al clickearlo.

```
<i class="fab fa-codepen fa-2x" data-container="body" data-toggle="popover" data-trigger="hover" data-placement="bottom" data-content="Codepen."></i>
```

¿Te animas a hacerlo con las otras marcas?

Conociendo Typeform

Formularios con Typeform (Parte I)

Regularmente las páginas o landing pages buscan mantener una forma de contacto con el usuario. Una de las formas más eficientes de lograr esto es través de un formulario.

Normalmente estos formularios, envían un email al administrador de la página con los datos que los usuarios ingresen. La lógica detrás de enviar un email de forma automática regularmente es manejada por un servidor, el cual es configurado e integrado con nuestra página. A pesar que no es una tarea en extremo compleja, se sale de los alcances de este curso. Para suplir esta necesidad utilizaremos una herramienta de terceros, que integra formularios y envío de correo electrónico.

Este es el caso de **Typeform**, un servicio proporcionado por una empresa que manejará la información que llenarán los usuarios, al mismo tiempo que nos entrega mejoras visuales para estos formularios, y nos enviará un correo.

Si ingresamos a su [página](#), podremos ver en detalle lo que la empresa ofrece. En resumen, muchos ejemplos de formularios para nuestras páginas.

Vamos a incorporar uno a nuestro proyecto, así que comencemos por registrarnos.

Una vez que hayas llenado los datos para la cuenta gratuita e ingresado tu correo y confirmado podemos loggearnos para realizar un formulario.

Panel de typeform

Typeform tiene una interfaz muy intuitiva y su versión gratuita ofrece 100 contactos mensuales gratis.

Para añadir un formulario podríamos utilizar un template, pero en este caso lo crearemos nosotros mismos. Para ello debemos ir a:

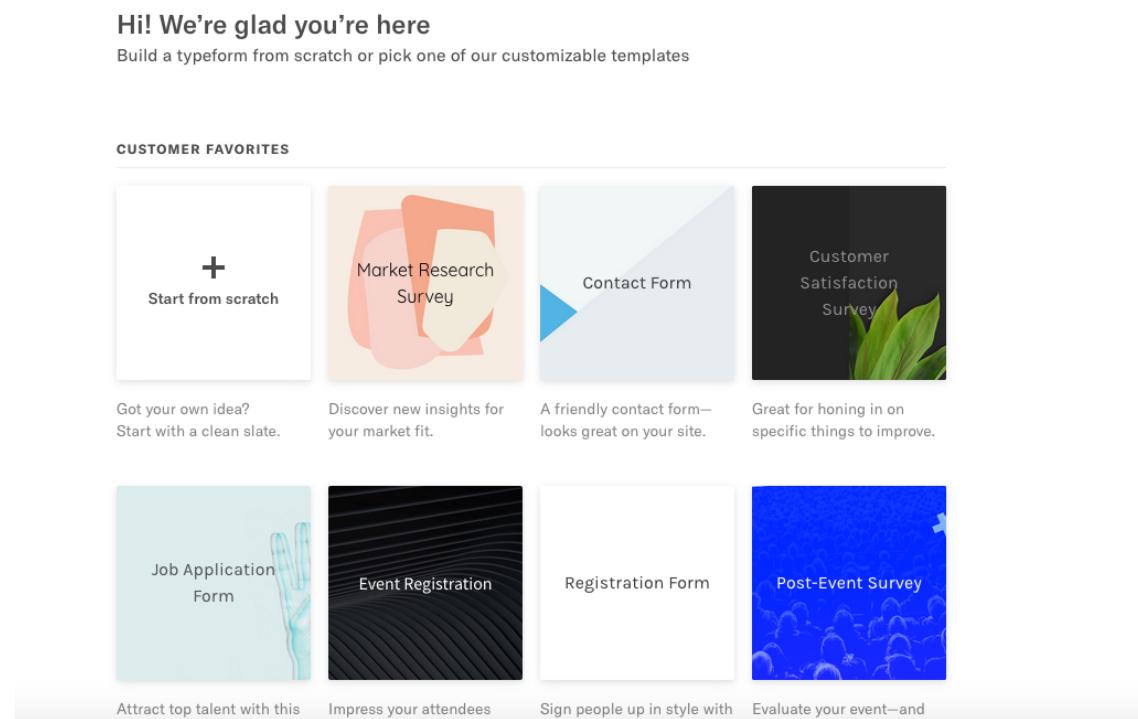


Imagen 20. Creación de template, parte 1.

y seleccionar **start from scratch**.

1. Nos ubicaremos al interior de nuestro **workspace** y hacemos *click* sobre **star from scratch**
2. Hacemos *click* donde dice **blocks**

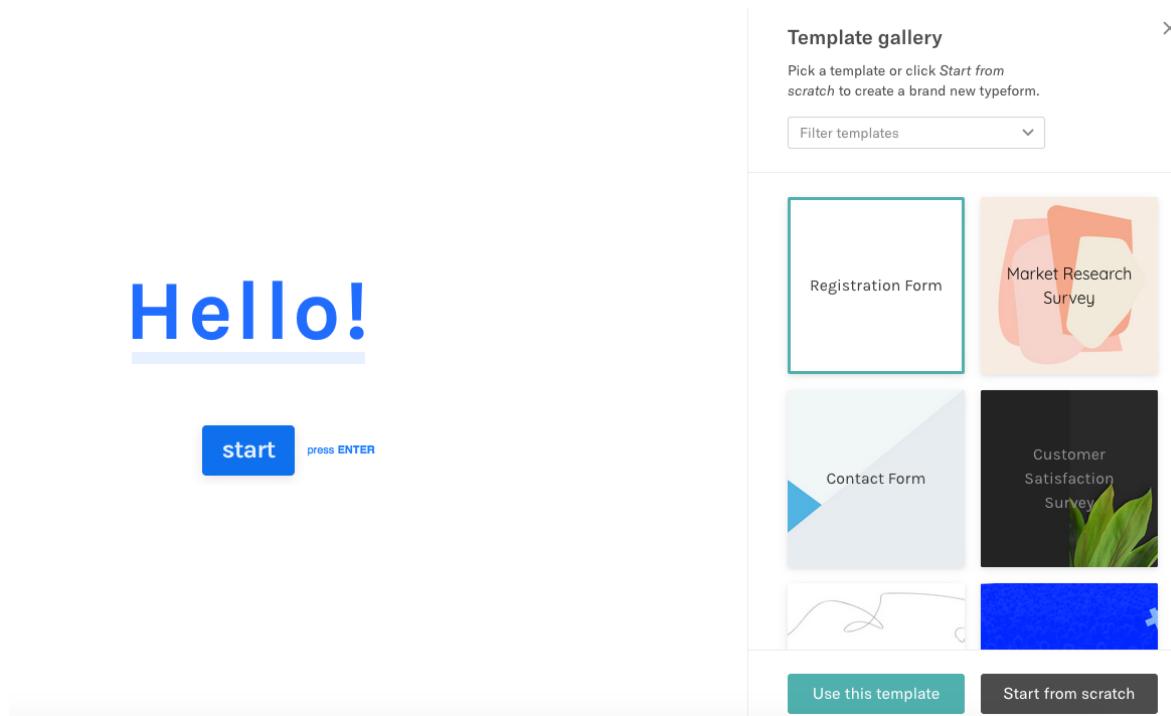


Imagen 21. Creación de template, parte 2.

3. Ahora tenemos muchas opciones para añadir elementos. Comenzaremos por crear algo muy simple, primero arrastra el bloque que dice **Welcome Screen** hacia la derecha.



Imagen 22. Creación de template, parte 3.

4. Aquí podemos añadir un mensaje de saludo. Yo escribiré "Hola, ¡Contáctenos!", luego debemos presionar **enter** para que automáticamente aparezca un nuevo bloque.

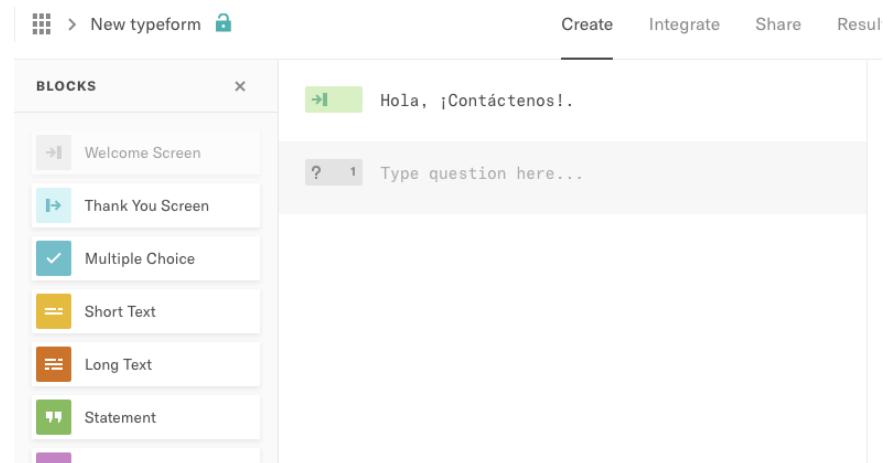


Imagen 23. Creación de template, parte 4.

5. En el nuevo bloque vamos a ingresar "¿Cuál es tu nombre?" y presionaremos **enter**.

6. En el nuevo cuadro escribiremos "¿Cuál es tu correo?" y presionaremos **enter**. Luego, debemos cambiar el tipo de este mensaje por 'Email', haz *click* sobre el cuadro y cámbialo. Saldrá un mensaje de advertencia, en el cual debes hacer click en aceptar.

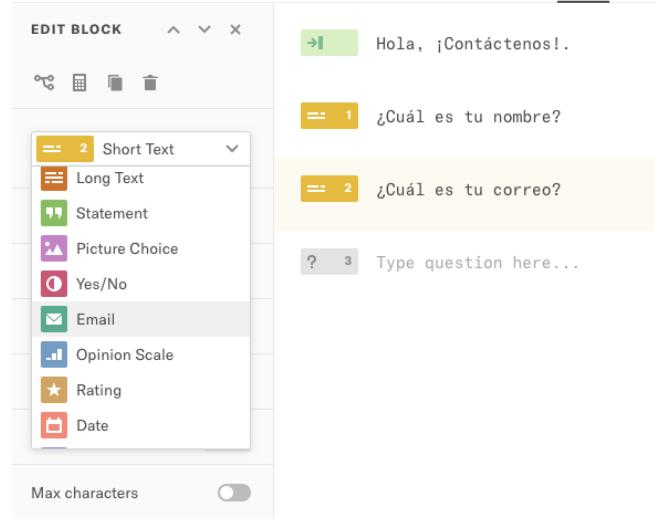


Imagen 24. Creación de template, parte 4.

7. El último bloque será añadir un lugar para el mensaje. Escribiremos en este bloque "Ingresa tu mensaje", luego cambiaremos el bloque a **long text** y presiona **enter**.

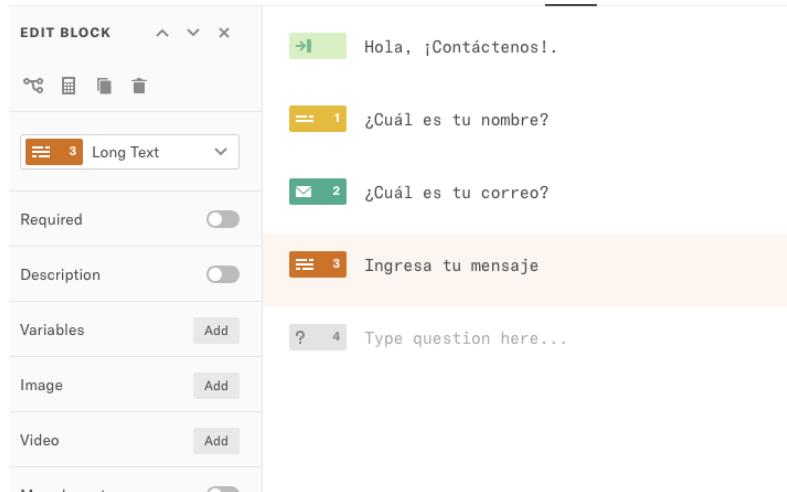


Imagen 25. Creación de template, parte 5.

8. Podemos eliminar el último bloque que apareció automáticamente, si lo seleccionamos y luego presionamos la papelera.

Ahora que tenemos una estructura básica del formulario, podremos iniciar el segundo paso que será obtener un formato válido para añadirlo en nuestra página.

9. Debes seleccionar **share** en el menu superior.

Your typeform lives in this link

Share it with people so they can visit and respond to your typeform

<https://adacher.typeform.com/to/MJ7jOP> Copy

Social media and email

Share your typeform on social media or send it in an email

f in t G+ e m Add meta settings

What else can you do?

Imagen 26. Creación de template, parte 6.

10. Esto te llevará a la pantalla de compartir, donde existen varias formas de integrar este formulario. Para nuestros fines, debemos seleccionar **EMBED**.
11. Desde esta página podremos ver una vista previa de nuestro formulario, elegir un tamaño y realizar algunos ajustes, así que haz los cambios que necesites.
12. El último paso para obtener nuestro formulario será presionar el botón `Get the code`. Lo que hará aparecer una nueva ventana con nuestro código.
13. Presiona el botón `copy code`. Este es el código que debemos añadir en nuestro proyecto.

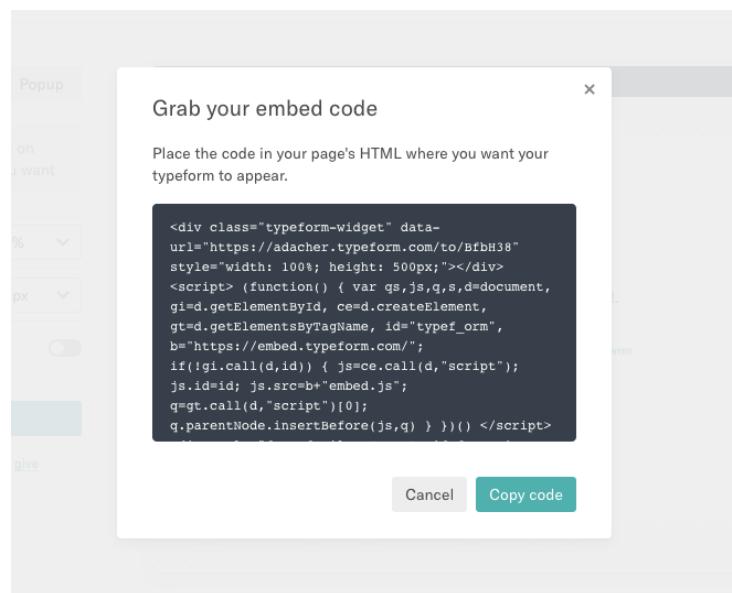


Imagen 27. Creación de template, parte 7.

Vamos a nuestro `index.html` donde está nuestro formulario de contacto, que ahora remplazaremos por nuestro nuevo código.

```
<div class="typeform-widget" data-url="https://franibe.typeform.com/to/XavIQj"
style="width: 100%; height: 500px;"></div> <script> (function() { var
qs,js,q,s,d=document, gi=d.getElementById, ce=d.createElement,
gt=d.getElementsByTagName, id="typef_orm", b="https://embed.typeform.com/";
if(!gi.call(d,id)) { js=ce("script"); js.id=id; js.src=b+"embed.js";
q=gt("script")[0]; q.parentNode.insertBefore(js,q) } })() </script>
<div style="font-family: Sans-Serif;font-size: 12px;color: #999;opacity: 0.5;
padding-top: 5px;"> powered by <a href="https://admin.typeform.com/signup?
utm_campaign=XavIQj&utm_source=typeform.com-13260463-
Basic&utm_medium=typeform&utm_content=typeform-embedded-
poweredbytypeform&utm_term=EN" style="color: #999"
target="_blank">Typeform</a></div>
```

Guardemos el archivo y recarguemos nuestro navegador para ver nuestro nuevo formulario.

Ahora lo vamos a probar. Llenaremos los datos, presionaremos enviar y volveremos a la página de **Typeform** para verificar los datos. Esto lo podemos hacer en las pestaña de **Result**.

Typeform nos servirá para hacer un formulario llamativo y personalizado sin requerir mucho código. La cuenta gratis nos servirá hasta para 100 mensajes.

Conclusiones

En esta unidad conocimos los conceptos básicos de JavaScript. Luego conocimos sobre la biblioteca jQuery. Utilizamos los componentes JS de Bootstrap como las modales, los carouseles y los popovers y finalmente implementamos un formulario con Typeform.

Recuerda seguir practicando y no olvides leer las documentación de los elementos mencionados.

Tomar el ritmo al desarrollo es cosa de tiempo y práctica, así que sigue aprendiendo a través de los desafíos.