

BLOQUE 2: Bootstrap

Proyecto: Landing Page

1 ¿Qué es Bootstrap?

Generar una plantilla básica sobre la que empezar un proyecto web que tenga en cuenta los distintos navegadores y tamaños de dispositivos consume gran cantidad de tiempo. Para facilitar y agilizar el proceso de diseño web, durante los últimos años han surgido varios entornos de trabajo que facilitan esta tarea. Uno de los frameworks que se han hecho más populares es Bootstrap¹.

Bootstrap o twitter-bootstrap es un framework creado por dos desarrolladores/diseñadores de Twitter para facilitar el diseño de nuevas aplicaciones web. El framework proporciona clases CSS y código javascript para definir la estructura de la página, crear componentes que respondan a eventos y estilizar los elementos html más habituales.

Las ventajas de Bootstrap dependen del perfil de la persona que vaya a utilizarlo. Para un desarrollador con poca experiencia en diseño le proporciona una forma muy rápida de crear un layout *responsive* básico de la página en la que empezar a introducir código. Para un diseñador, Bootstrap ofrece una gran cantidad de clases CSS que se pueden personalizar sin tener que partir de cero. En cualquier caso, es un compendio de buenas prácticas.

2 Empezando a trabajar con Bootstrap

La versión 3 de Twitter Bootstrap es un cambio importante respecto a sus versiones anteriores. Es un framework para móviles y puede presumir de ser uno de los mejores marcos de estilo CSS para la construcción de diseños de sitios web *responsive*.

Puedes descargar la carpeta con los ficheros necesarios en ². Una vez que se descomprima el archivo de almacenamiento, podrás ver un montón de archivos y carpetas. Ahora mismo, los ficheros que nos interesan están en el directorio *dist*. Copia todas las carpetas contenidas en *dist* a tu carpeta de trabajo.

Dentro de la carpeta *dist* tenemos tres directorios. Echemos un vistazo a cada uno de ellos. La carpeta *css* contiene cuatro archivos CSS: *bootstrap.css*, *bootstrap.min.css*, *bootstrap-theme.css* y *bootstrap-theme.min.css*. Nuestro principal archivo CSS es *bootstrap.css*, y hay que incluirlo en todas nuestras páginas HTML. *bootstrap.min.css* es sólo la versión comprimida de *bootstrap.css*. Es necesaria cuando estamos desplegando nuestra aplicación. *bootstrap-theme.css* es un archivo CSS adicional, que es opcional. Este archivo contiene las clases necesarias para efectos 3D en los botones y otros elementos. Del mismo modo, *bootstrap-theme.min.css* es el archivo CSS comprimido de *bootstrap-theme.css*. Si vamos a la carpeta *js*, tenemos dos archivos *bootstrap.js* y *bootstrap.min.js*. Estos archivos contienen las principales bibliotecas JavaScript de arranque para cosas como menús desplegables, búsqueda automática, sugerencias, y muchas otras poderosas funcionalidades de JavaScript. En la carpeta *fonts* se incluyen las fuentes de los iconos del proyecto Glyphicons.

3 ¿Qué vamos a construir con Bootstrap?

Vamos a construir una *Landing page* estática usando Twitter Bootstrap 3, que se llamará "Rental Vacations". Vamos a ver poco a poco su estructura.

Bootstrap 3 comprende elementos de HTML5. Vamos a crear una nueva página HTML y añadiremos las etiquetas básicas que están presentes en todos los documentos HTML:

¹<http://getbootstrap.com/>

²<http://getbootstrap.com/>

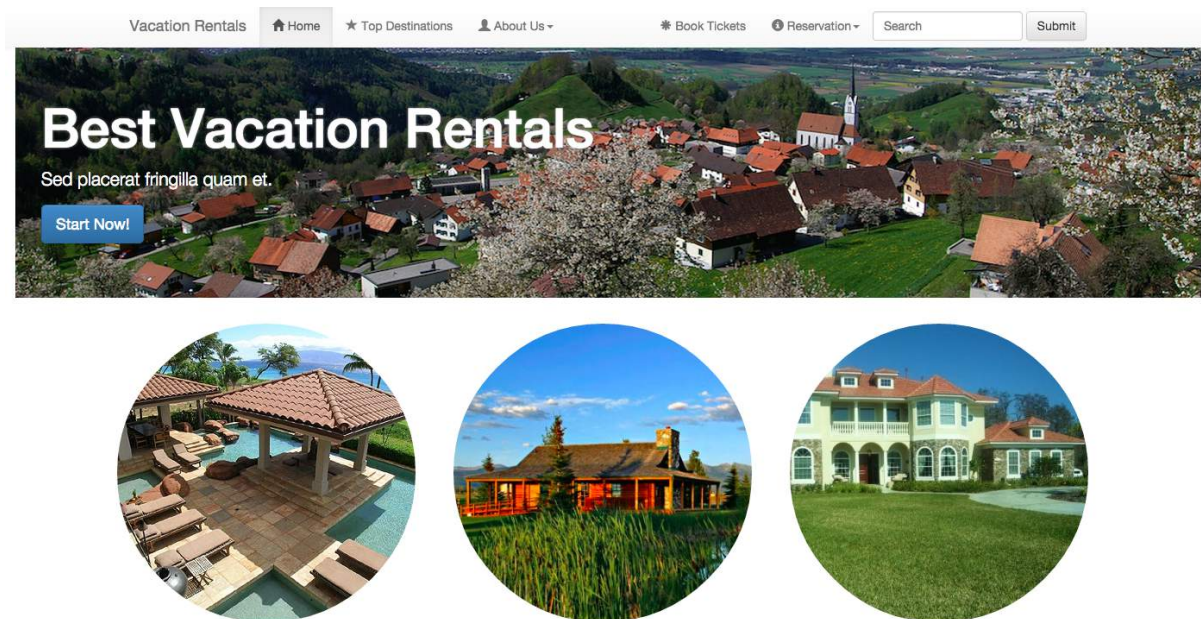


Figure 1: Ejemplo utilizando Bootstrap.

```
<html>
<head>
  <title>Bootstrap 101 Template</title>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <!-- Bootstrap -->
  <link href="css/bootstrap.min.css" rel="stylesheet" media="screen">
</head>

<body>
  <h1>Hello, world!</h1>

  <script src="//code.jquery.com/jquery.js"></script>
  <script src="js/bootstrap.min.js"></script>
</body>
</html>
```

Esta versión de Bootstrap permite desarrollar proyectos web que son compatibles con varios tipos de dispositivos (móviles, tabletas, ordenadores de sobremesa, pantallas de retina, etc), es necesario dejar que el navegador sepa que tiene que escalar la página web de forma adecuada en todos los dispositivos. La etiqueta *viewport* hace esto. Aquí, hemos establecido el *content-width* a la anchura del dispositivo.

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Después de fijar la etiqueta *meta viewport*, hemos importado el documento de Twitter Bootstrap CSS 3, *bootstrap.css*. Este archivo es completamente auto-explicativo. Cada etiqueta viene con su descripción. Si alguna vez quieres saber lo que hace cualquier clase de Bootstrap, basta con abrir el archivo *bootstrap.css* y encontrarás la respectiva etiqueta y podrás leer el comentario.

Vamos a pasar a la etiqueta *body* del fragmento HTML anterior. Inicialmente, para el ejemplo, hemos escrito "Hola mundo" dentro de la etiqueta *h1*. A continuación hemos incluido los archivos JavaScript necesarios de la carpeta *js*. Asegúrate de que estos archivos JavaScript se incluyen en cada página que utilices.

```
<script src="//code.jquery.com/jquery.js"></script>
```

```
<script src="js/bootstrap.min.js"></script>
```

Ahora que tenemos nuestra estructura básica lista, vamos a pasar a la edición de los diferentes componentes de nuestra página web. Estos componentes son la parte más importante de un sitio web.

3.1 Menu

El diseño de un menú en Bootstrap 3 es sencillo. Vamos a empezar la construcción del menú de la página web. Primero, dentro de la etiqueta del *body* del archivo HTML añadimos la siguiente etiqueta:

```
<nav class="navbar-wrapper navbar-default navbar-fixed-top" role="navigation">
</nav>
```

Dado que Bootstrap es compatible HTML 5, utilizamos la etiqueta `<nav>` para nuestro menú de navegación. Vamos a ver cada una de las clases que se han aplicado. La clase *navbar-wrapper*, como su nombre indica, es el envoltorio de los elementos del menú. La clase *navbar-default* da el color de fondo para el menú y también aplica un borde al mismo. La clase *navbar-fixed-top* ayuda a hacer que la barra de menú se quede fija en la parte superior de la página, incluso si se estamos desplazándonos. Si no quieres un menú superior fijo, entonces puedes eliminar esa clase.

Seguimos añadiendo la siguiente línea de código:

```
<nav class="navbar-wrapper navbar-default navbar-fixed-top" role="navigation">
  <div class="container">
    <div class="navbar-header">
      <a class="navbar-brand" href="#">Vacation Rentals</a>
    </div>
```

El *div container* se asegura de que su contenido esté alineado, con un margen apropiado a cada lado de la página web y también que esté centrado en la pantalla del navegador. El *div navbar-header* se utiliza a efectos de identificación. Añadimos un enlace de la clase *navbar-brand* para mostrar el nombre de la página web.

Ahora añade el siguiente código fuera del *div navbar-header*:

```
<div class="collapse navbar-collapse navbar-ex1-collapse">
  <div class="collapse navbar-collapse">
    <ul class="nav navbar-nav">
      <li class="active">
<a href="#"><span class="glyphicon glyphicon-home"></span>Home</a>
      </li>
      <li>
<a href="#"><span class="glyphicon glyphicon-star"></span> Top Destinations</a>
      </li>
      <li class="dropdown">
        <a href="#" class="dropdown-toggle" data-toggle="dropdown"><span class="glyphicon glyphicon-user"></span>About Us<b class="caret"></b></a>
        <ul class="dropdown-menu">
          <li><a href="#">Company Details</a></li>
          <li><a href="#">Contact Us</a></li>
        </ul>
      </li>
    </ul>
    <form class="navbar-form navbar-right" role="search">
      <div class="form-group">
<input type="text" class="form-control" placeholder="Search">
      </div>
      <button type="submit" class="btn btn-default">Submit</button>
    </form>
```

```

        <ul class="nav navbar-nav navbar-right">
            <li>
<a href="#"><span class="glyphicon glyphicon-asterisk"></span> Book Tickets</a>
            </li>
            <li class="dropdown">
<a href="#" class="dropdown-toggle" data-toggle="dropdown"><span class="glyphicon
glyphicon-info-sign"></span> Reservation<b class="caret"></b></a>
                <ul class="dropdown-menu">
                    <li><a href="#">Cancel</a></li>
                    <li><a href="#">Confirm</a></li>
                </ul>
            </li>
        </ul>
    </div><!-- /.navbar-collapse -->
</div>
</nav>

```

El código anterior es el contenido principal de nuestro menú. *navbar-collapse* se aplica para hacer el menú táctil y para cambiar su forma para dispositivos más pequeños. Los contenidos que se encuentran dentro son bastante sencillos. Hemos utilizado las etiquetas básicas *ul* y *li* para enumerar los elementos del menú. También hay un elemento *form* de la clase *navbar-right* para colocarlo a la derecha de la barra de navegación. Y finalmente se muestra la estructura básica para colocar un menú *drop down* dentro de la barra de navegación.

Lo más importante aquí es recordar las clases que se utilizan para la visualización de determinados tipos de elementos en Twitter Bootstrap. También se puede hacer referencia a la documentación, que es muy recomendable cuando estás en cualquier tipo de confusión.

Una vez hemos terminado con el menú de navegación, vamos a pasar a la construcción del resto de las partes del cuerpo.

3.2 Cabecera

Bootstrap 3 da una clase llamada *jumbotron* muy fácil de utilizar que se puede utilizar para mostrar las grandes cabeceras y contenidos. Esto se utiliza sobre todo en los sitios web de venta de productos. Por eso, tenemos que añadir la siguiente línea de código:

```

<div class="container-fluid">
    <div class="jumbotron">
        <h1>Best Vacation Rentals</h1>
        <p>Sed placentat fringilla quam et.</p>
        <p><a class="btn btn-primary btn-lg">Start Now!</a></p>
    </div>
</div>

```

La diferencia básica entre la clase *container* y la clase *container-fluid* es que la primera se centra en el navegador con un ancho determinado, mientras que la segunda se establece en el 100% del ancho del navegador.

Luego tenemos el elemento *div* de la clase *jumbotron* y en su interior se ha colocado una etiqueta *h1*, una etiqueta *p*, y finalmente un botón.

3.3 Contenido

Ahora tenemos que dividir el área de contenido en tres áreas iguales y ponerlas una al lado de la otra. Para ello en Bootstrap 3 se utiliza un sistema de *grid* o rejilla (ver Figura 2).

Twitter Bootstrap es un sistema de grid de 12 columnas. Divide la pantalla en 12 partes iguales y tenemos que especificar qué elemento HTML ocupa cada parte del grid. Así que, en resumen, cualquier elemento ocupará como mínimo una parte del grid.

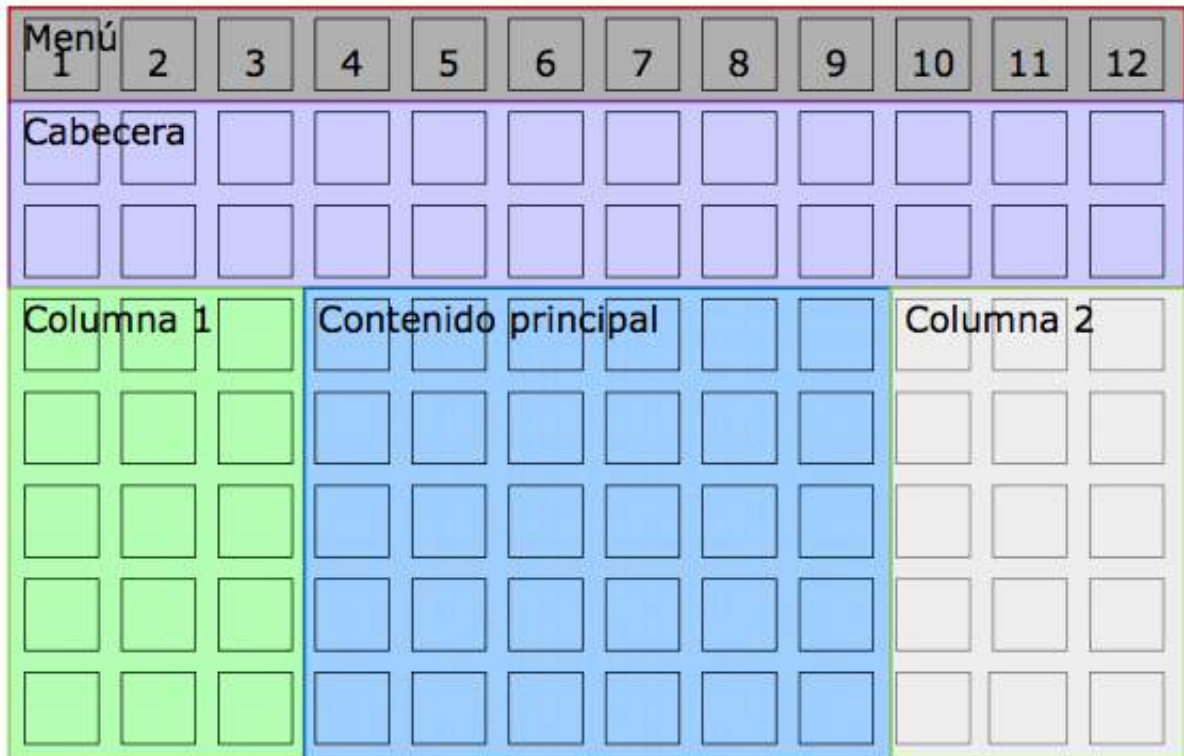


Figure 2: Ejemplo de grid o rejilla en Bootstrap.

Vamos a ver las etiquetas que se utilizan:

```
<div class="container">
  <div class="row">
    <div class="col-md-4">
      <a href="#"></a>
      <h3 class="text-center">Type 1</h3>
      <p>Dummy Text..</p>
      <a href="#" class="btn btn-success">Book Now @ $599</a>
    </div>

    <div class="col-md-4">
      <a href="#"></a>
      <h3 class="text-center">Type 2</h3>
      <p>Dummy Text..</p>
      <a href="#" class="btn btn-danger">Book Now @ $899</a>
    </div>

    <div class="col-md-4">
      <a href="#"></a>
      <h3 class="text-center">Type 3</h3>
      <p>Dummy Text ...</p>
      <a href="#" class="btn btn-info">Book Now @ $699</a>
    </div>
  </div>
</div>
```

En primer lugar, estamos utilizando *container* en vez de *container-fluid*. Luego hemos definido una

fila de elementos hermanos. En nuestro caso, hay tres elementos *div* que están contenidos dentro de una sola fila. Así que empezamos una nueva fila añadiendo un nuevo *div* con la clase *row*.

La fila ahora tiene tres etiquetas más *div* de la clase *col-md-4*. El número 4 en la clase nos dice que ocupa cuatro cuadrículas. Así que los tres *divs* hermanos con cuatro columnas cada uno ocuparán las 12 columnas del grid. Asegúrate que no se está especificando más de 12 columnas en una sola fila. Otras clases para especificar la posición de los elementos serían *col-md-** que nos dice que en los dispositivos de una pantalla media ($\geq 992\text{px}$) los elementos hermanos serán colocados lado el uno al otro, mientras que en los dispositivos más pequeños se van a apilar uno encima de la otro. También tenemos clases para la pequeña pantalla (*col-sm-**).

Ahora que tenemos cada *div* colocado adecuadamente, vamos a rellenarlos con algo de contenido. En nuestro caso, el contenido es único para cada uno de ellos con diferentes imágenes.

Las imagenes añadidas son de la clase *img-responsive* que hace que encaje con el tamaño del *div* padre independientemente de su propio tamaño. La imagen es sensible al tamaño de cada *div* hermano. Luego tenemos a las etiquetas normales etiqueta *h3* y *p*. Finalmente, hemos añadido un enlace que parece un botón. Con la clase *btn* se puede dar la apariencia de un botón a cualquier elemento. *btn-success* se utiliza para que se vea de color verde. Hay otras clases como: *btn-info*, *btn-default*, *btn-warning*, y *btn-danger*.

El área del pie es similar a la del contenido.

```
<footer>
  <div class="container">
    <div class="row">
      <div class="col-md-4">&copy; 2013</div>
      <div class="col-md-4">
        <ul class="nav nav-pills">
          <li class="active"><a href="#">About Us</a></li>
          <li><a href="#">Support 24x7</a></li>
          <li><a href="#">Privacy Policy</a></li>
        </ul>
      </div>
      <div class="col-md-4">
        <h3 class="text-right">Vacation Rentals</h3>
      </div>
    </div>
  </div>
</footer>
```

3.4 Pie

Hemos utilizado la etiqueta *footer* HTML5. A continuación, un *div* de la clase *container* y una fila de la clase *row*. Hemos dividido toda la zona en tres secciones iguales con el sistema 12 cuadrículas en mente.

4 Aplicando Bootstrap a una página de viajes.

Una vez que hemos visto con un ejemplo cómo funciona Bootstrap para hacer una *landing page*, te proponemos utilizar las clases CSS que ofrece bootstrap para cambiar el estilo de la página HTML que te proporcionamos (*landingPage.html*) para que el aspecto cambie del que aparece en la Figura 3 al que aparece en las Figuras 4 y 5. Algunas de las clases que tendrás que utilizar serán:

- *nav* para los elementos de navegación
- *pull-left* y *pull-right* para alinear los elementos a la izquierda o a la derecha.
- *row* para indicar que vamos a empezar una fila de elementos
- *col-md-4* para indicar el número de columnas del grid van a ocupar los elementos.

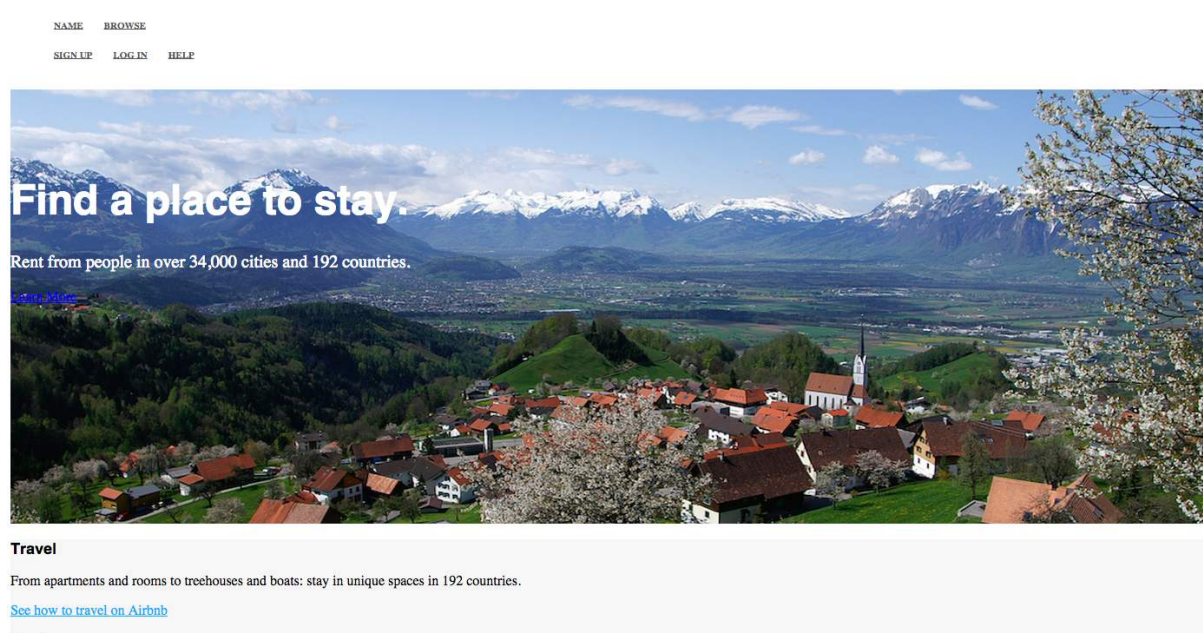


Figure 3: Aspecto Inicial.

- *container*
- *thumbnail* que facilita mostrar grids de imágenes

```
<div class="thumbnail">
  
</div>
```

Los cambios en el estilo de la página los tendrás que hacer en el fichero que te proporcionamos *main.css*. Algunas de las reglas que tendrás que añadir en el fichero son las siguientes reglas:

- Seleccionar la clase *neighborhood-guides* y establece el color de fondo a #efefef, border-bottom a 1px solid #dbdbdb
- Seleccionar el elemento h2 dentro de *neighborhood-guides* y establece el color a #393c3d y el tamaño de la fuente a 24px.
- Seleccionar el elemento p dentro de *neighborhood-guides* y establecer el tamaño de la fuente a 15px y el margen inferior a 13px.

No olvides incluir el fichero *main.css* en *landingPage.html*.

```
<link rel="stylesheet" href="main.css">
```

Lo que hemos visto hasta ahora es una pequeña introducción a Bootstrap. Si quieres conocer más, visita los siguientes sitios web:

- <http://www.w3schools.com/bootstrap/default.asp>
- <http://getbootstrap.com/getting-started/>

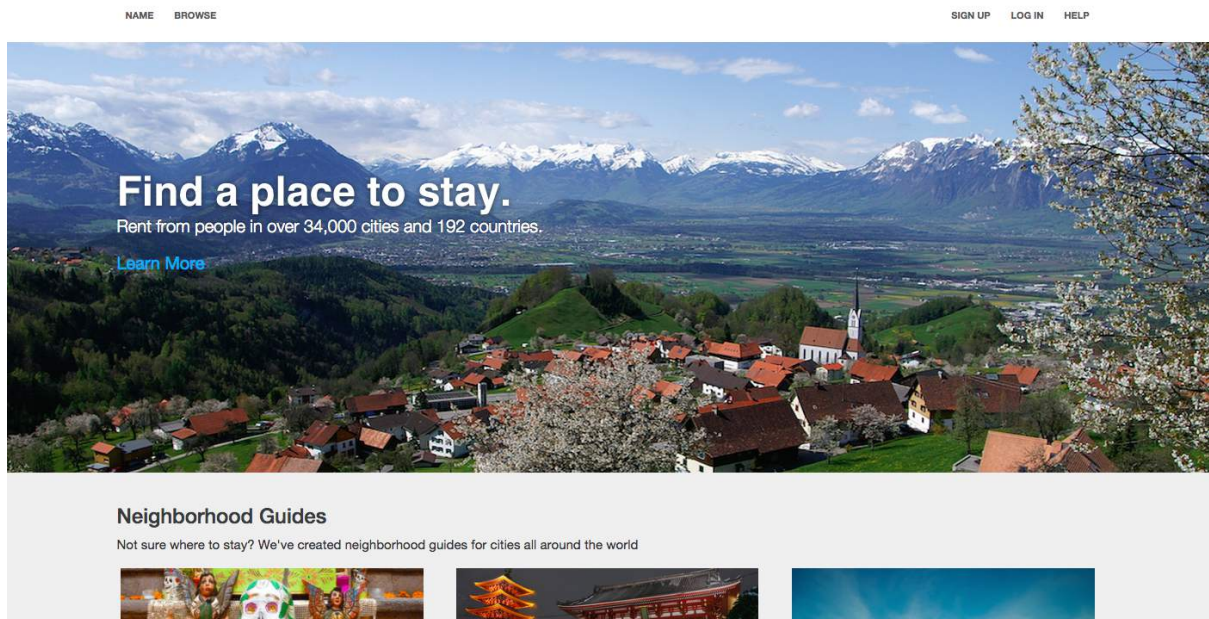


Figure 4: Aspecto Final.

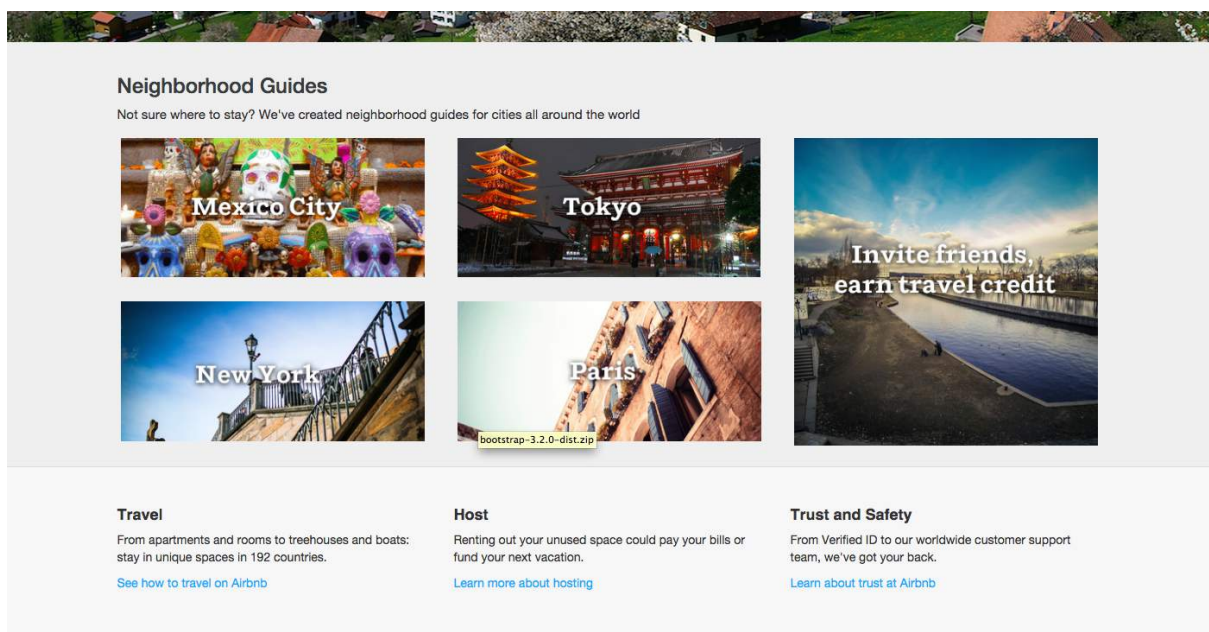


Figure 5: Aspecto Final.