

Proyecto Final

Diseño Web con HTML, CSS y JavaScript



Silvia Rubio Izquierdo

01

Introducción



 **Peluditos**
Travel

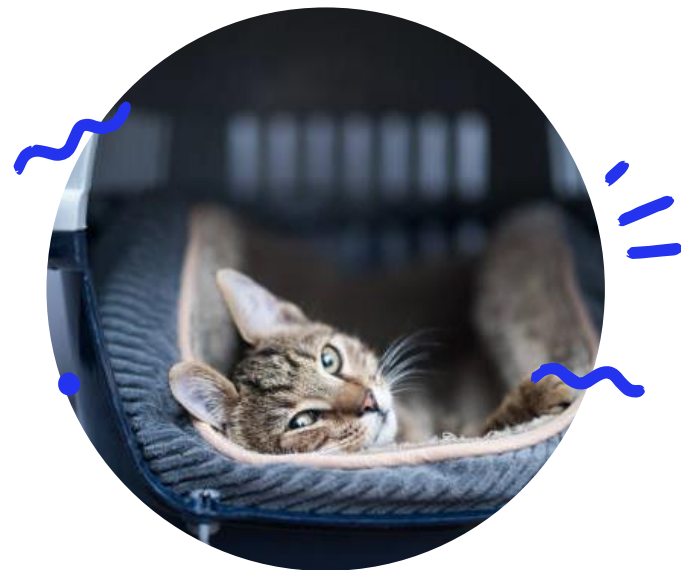
Peluditos travel está destinada a los viajes con mascotas.

El objetivo principal es proporcionar información de calidad para los amantes de los animales. En la página se encontrará todo tipo de información para que el cliente pueda planificar las mejores vacaciones con peluditos sin tener que preocuparse de nada



02

Descripción del proyecto



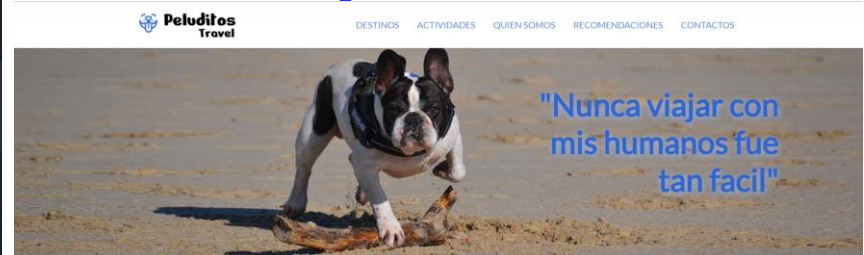
 **Peluditos**
Travel

El proyecto se ha realizado en HTML, CSS y JavaScript. Este consiste en una web de viajes en la que los usuarios encontrarán recomendaciones para sus viajes, actividades y los mejores destinos para viajar con sus mascotas. Además, para el diseño de imágenes he utilizado PhotoShop.

Código

```
17 <!--El menu esta creado a traves de una lista con un menu que deriva a cada pagina enlazada,
18 tambien esta adaptado a version movil, despues meto el logo de la pagina con una etiqueta de img -->
19 <section class="header_nav">
20 <div>
21 <a href="index.html"
22 </a>
23 </div>
24 <div id="menu_toggle" type="checkbox">
25 <label class="menu_button_container" for="menu_toggle">
26 <div class="menu_button"></div>
27 </label>
28 <ul class="menu_menu_nav">
29 <li class="menu_li"><a href="destinos.html">Destinos</a></li>
30 <li class="menu_li"><a href="actividades.html">Actividades</a></li>
31 <li class="menu_li"><a href="quienesomus.html">Quiénes somos</a></li>
32 <li class="menu_li"><a href="recomendaciones.html">Recomendaciones</a></li>
33 <li class="menu_li"><a href="contactos.html">Contactos</a></li>
34 </ul>
35 </section>
36 <!--fin del header-->
37
38 <!--Inicio del main-->
39 <!--En el main creo el contenido de la pagina principal con el primer div enlazado a una foto principal metida en una etiqueta img,
40 con un título creado en un h1 y finalizo con otro div contenedor de varias fotografías referidas al contenido de la temática
41 de la pagina, cada imagen deriva a la pagina de destinos con un a href-->
42 <main>
43 <div class="banner_home">
44 <h1 class="title_banner_home">
45 "Nunca viajar con mis humanos fue tan facil"
46 </h1>
47 
48 </div>
49 </main>
50
51 </body>
52 </html>
```

Respuesta




03

Colores y tipografía




Peluditos
Travel


Colores principales


 rgb(51,102,204)
(encabezados y
menú)


 #000 (texto)


 rgb(255,255,255)
(texto footer)


Colores secundarios


 rgb(51,102,204) con
opacidad 1


 #333
(menú responsive)

 #222
(menú responsive)

 #3B5998
(Facebook)

 #4099FF
(Twitter)

 #d34836
(Google)

 #f39c12
(correo)

Tipografías

"Lato", *sans-serif*
(Fuente principal de la
web)

Italic
(Footer)

04

///
**Mi
experiencia**
~~~~~





El proyecto lo he realizado con el editor de código *Visual Studio Code*, utilizando como herramientas los lenguajes: HTML, CSS y JavaScript.

A modo de resumen, en aquello en lo que he encontrado mayor dificultad es:

- Media Query.
  - Constantes fallos en el responsive en móvil y Tablet.
- Imágenes.
  - Problemas en la transformación de imágenes en media query.
- Sliders de JavaScript.
  - Tuve que investigar el modo de ejecutar varios Sliders en el mismo código.

# Media Query

```
#menu_toggle ~ .menu li {  
  height: 0;  
  margin: 0;  
  padding: 0;  
  border: 0;  
  transition: height 400ms cubic-bezier(0.23, 1, 0.32, 1);  
}  
#menu_toggle:checked ~ .menu li {  
  height: 2.5em;  
  border: 1px solid #333;  
  padding: 0.5em;  
  transition: height 400ms cubic-bezier(0.23, 1, 0.32, 1);  
}
```

De primeras puse una única clase para todos los títulos y me dificultaba la colocación de las imágenes. Por ello, finalmente acabé atribuyéndole una clase diferente a cada título.

Una de las soluciones que encontré, fue la implementación de una transición con altura, como se ve en la imagen.

```
.title_banner.home {  
  font-size: 1em;  
  top: 11%;  
  left: 60%;  
  z-index: 0;  
}  
.title_banner.actividades {  
  font-size: 1.2em;  
  top: 20%;  
  left: 60%;  
  z-index: 0;  
}  
.title_banner.recomendaciones {  
  font-size: 1.2em;  
  top: 10%;  
  left: 68%;  
  z-index: 0;  
}  
.title_banner.qs {  
  font-size: 1.2em;  
  top: 11%;  
  left: 60%;  
  z-index: 0;
```

# Imágenes en Media Query

```
.banner_qs_person_img{  
  width: 200% !important;  
  margin-left: 10% !important;  
  overflow: hidden !important;  
}  
.banner_qs_person_img_r{  
  width: 200% !important;  
  margin-right: 10% !important;  
  overflow: hidden !important;  
}  
.banner_qs_person_text{  
  padding: 5px 15px !important;  
}
```

En la pantalla de inicio, para que esta no perdiese su sentido estético, tuve que implementar clases diferentes a las imágenes, para que se viese en dos columnas en Tablet, y en una en móvil.

Para que las imágenes mantuviesen su estética en el media query, y no cambiase de tamaño de manera ilógica, encontré la regla de estilos !important, para que estas no perdiesen su sentido.

```
@media (max-width: 800px) {  
  .column {  
    flex: 50%;  
    max-width: 50%;  
  }  
}  
@media (max-width: 600px) {  
  .column {  
    flex: 100%;  
    max-width: 100%;  
  }  
}
```

# Sliders JavaScript

```
const destinos = ['nieve', 'playa', 'montana'];

destinos.forEach(function(destino){
  let contador = 0;

  const sliderText = document.querySelector( 'div.slider_destinos.' + destino + ' > button.slider_button.next');
  const sliderPrev = document.querySelector( 'div.slider_destinos.' + destino + ' > button.slider_button.prev');
  const sliderImg = document.querySelectorAll( 'div.slider_destinos.' + destino + ' > div.slider_container > img.slider_img');
  const sliderPunto = document.querySelectorAll( 'div.slider_destinos.' + destino + ' > ul.slider_ul > li.slider_li > button.slider_punto');

  sliderText.addEventListener( 'click', ()=>{
    contador = contador + 1;
    if( contador >= 6 ){
      contador = 0;
    }
    // You, your + 20.04 ...

    sliderImg.forEach( ( eachimg , i )=>{
      sliderImg[i].classList.remove( 'isActive');
    });

    sliderImg[contador].classList.add( 'isActive');

    sliderPunto.forEach( ( eachpunto , i )=>{
      sliderPunto[i].classList.remove( 'isActive');
    });

    sliderPunto[contador].classList.add( 'isActive');

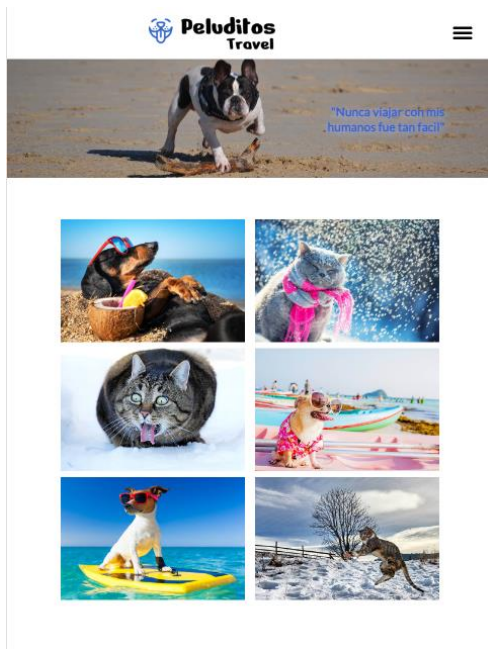
  })

  sliderPrev.addEventListener( 'click', ()=>{
    contador = contador - 1;
    if( contador <= -1 ){

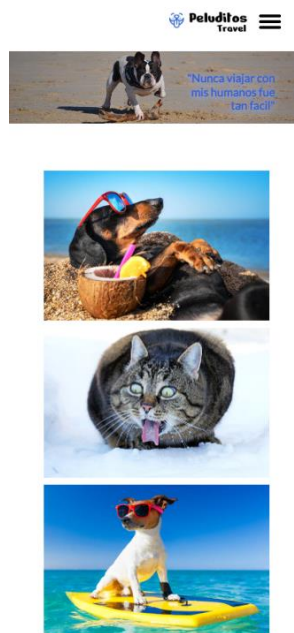
```

Para lograr que todos los Sliders funcionasen tuve que buscar información en internet, puesto que al meter un único slider si que funcionaba, pero encontré dificultades para meter varios. Finalmente, conseguí que todos los sliders funcionasen en el mismo código.

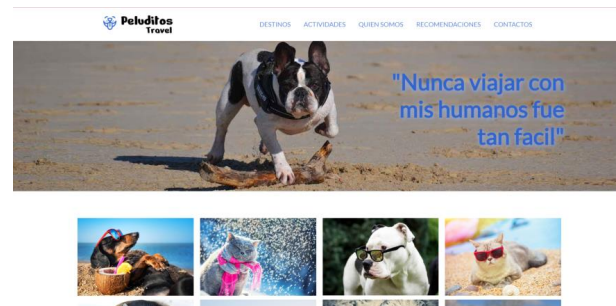
# Responsive final



**Tablet**



**Móvil**



**Desktop**



05

# Conclusiones



En este proyecto he visto la oportunidad de explotar mi creatividad al máximo y poner a prueba mis conocimientos en distintos lenguajes de programación web, y diseño.

El proyecto me ha servido también para darme cuenta de que la programación nos permite dar rienda suelta a la imaginación, y que sabiendo donde buscar, no hay casi nada que no se pueda hacer.

La realización de esta página, con lo que a investigación respecta, me ha ayudado a adquirir muchos conocimientos, no ya solo de programación y diseño como tal, sino también del cuidado de mascotas, que a fin de cuentas es lo que buscaba transmitir.

La persecución de una estética más profesional, me ha llevado a investigar más a fondo el lenguaje CSS, pues ha sido con el que más problemas he encontrado.



06

# Referencias y webgrafía





## Referencias fotográficas:

- <https://unsplash.com/es>
- <https://pixabay.com/es/>
- <https://www.pexels.com/es-es/>
- <https://tinypng.com/>

## Referencias estéticas:

- <https://pets.travel/>
- <https://www.travelguau.com/home>
- <http://www.mascotour.com/es>

## Referencias de código:

- <https://developer.mozilla.org/es/docs/Web/CSS/Reference>

Realizado por: Silvia Rubio Izquierdo