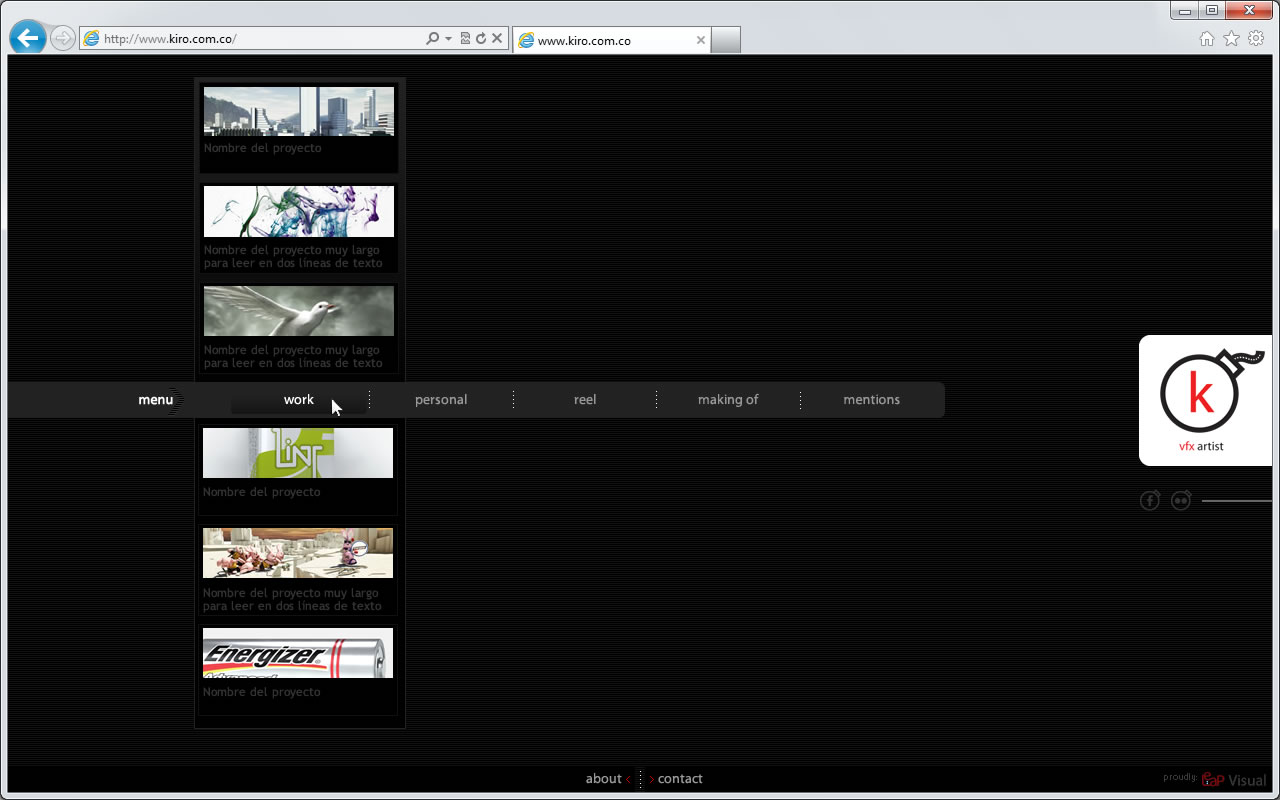
# Taller de HTML 5 y CSS 3

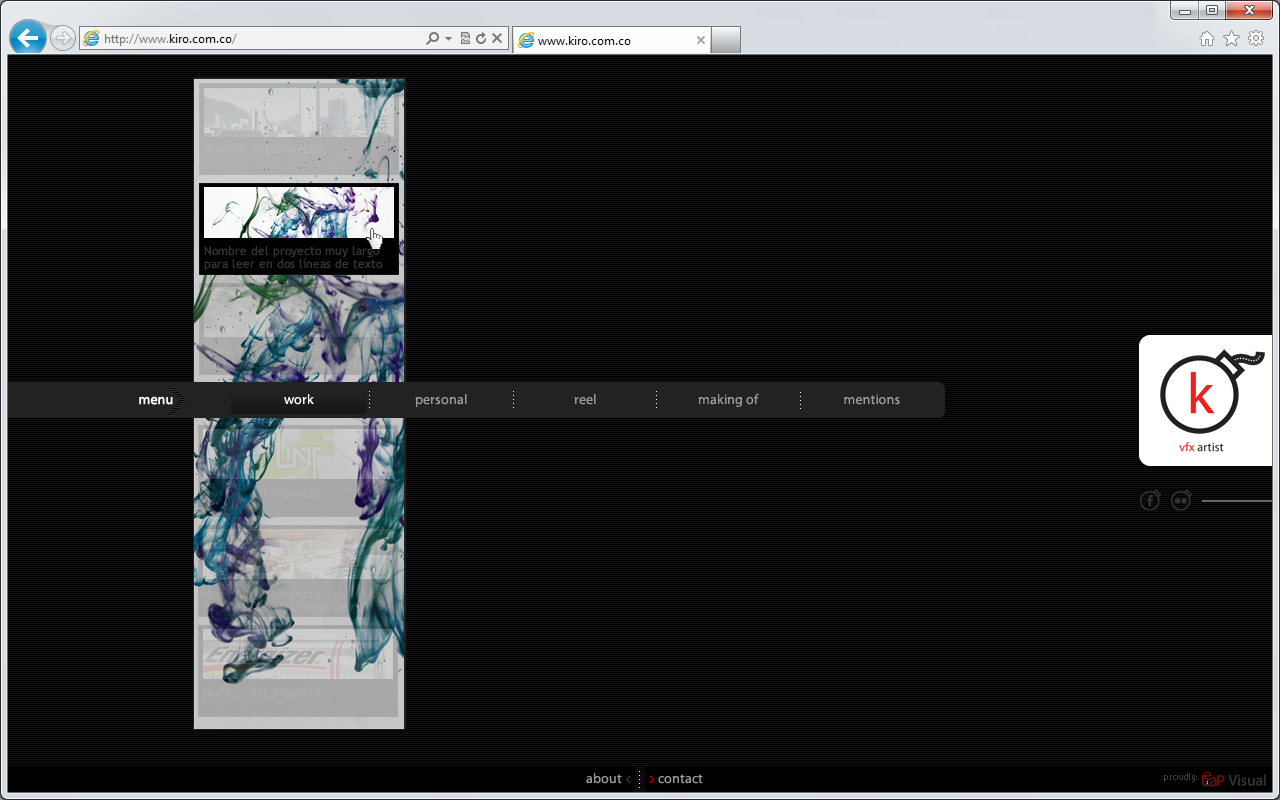
Para crear sitios web extraordinarios, debemos dar pasos cortos y seguros que nos permitan fortalecer los cimientos de sus conocimientos. Este taller fue adaptado del proyecto de web site para Kirobot, un proveedor de agencias de publicidad que modela y renderiza imágenes promocionales con alto realismo.

## Home del sitio

El diseño aprobado por el cliente muestra una distribución atípica de la identidad y el menú principal de navegación:

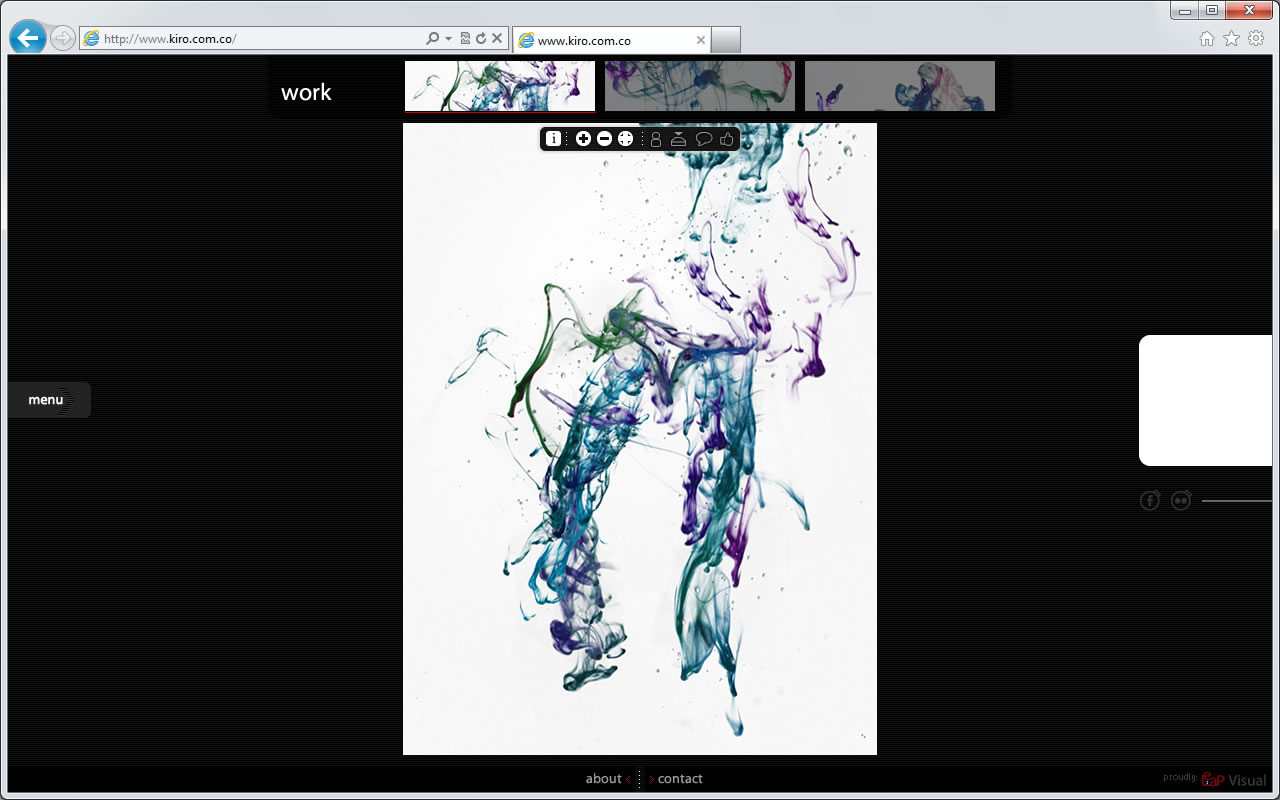


La identidad se encuentra a la derecha (no a la izquierda como tradicionalmente se utiliza) y cambia de tamaño cuando se interactúa con ella. El menú se ubica al lado izquierdo, pero inicia cerrado ocupando un mínimo de espacio.



Cuando se da click o tap en el área visible del menú, este se despliega horizontalmente y permite ver las opciones de navegación principales. Cuando se pasa el mouse o se toca la pantalla sobre el texto de las opciones "WORK" y "PERSONAL", se visualiza de forma animada un submenú vertical con las imágenes en miniatura de los proyectos publicados en el portafolio web.

Luego que el submenú es visible, en el momento que se interactúa con las miniaturas, el fondo del contenedor muestra una imagen de fondo igual a la miniatura (pero de mayor dimensión). Claramente, ocurre lo mismo con cada opción del submenú.



Finalmente, cuando se selecciona un proyecto del submenú vertical, el menú se cierra y desciende del borde superior una lista horizontal de las imágenes que componen el proyecto. Al tocar con el mouse o un dedo, la miniatura de cada imagen, se muestra una vista de alta resolución sobre la que se puede acercar o alejar según el deseo del usuario.

## Antes de Iniciar

Para llevar a cabo este laboratorio, descargue el archivo HTML Lab – resources.zip de recursos y descomprima su contenido en su computador.

Descargue el archivo Basicos Jquery.docx para trabajar algo de javascript entro del laboratorio.

Por último, use el presente archivo para seguir los pasos del laboratorio y por favor tenga en cuenta:

TODOS LOS TEXTOS CONTIENEN VÍNCULOS A DESCRIPCIONES OFICIALES DE LAS ETIQUETAS Y PROPIEDADES CSS.

Debe seguirlos, leerlos y asegurarse de entenderlos.

# INICIO DEL LABORATORIO

En este laboratorio aprenderemos a trabajar los estilos CSS con una metodología clara que facilite su edición y aprovecharemos las propiedades de CSS 3.0 de forma que la animación y la interacción requieran menos tiempo escribiendo código javascript. Debemos partir de la idea que trabajas con un diseñador que dejó listo la propuesta visual y una parte del código HTML o de las hojas de estilo en cascada.

#### Tecnologías que usaremos

* Html5: como tecnología base para nuestra implementación, estructura semántica y controles de multimedia. Pueden aprender las reglas y bases del HTML en la página W3Schools [si abre este vínculo](https://www.w3schools.com/html/html_elements.asp).
* CSS 3.0: Para controlar la apariencia visual (diseño) de la interfaz y mejorar la experiencia de usuario. Pueden aprender las reglas y bases del CSS en la página W3Schools, [si abren este vínculo](https://www.w3schools.com/css/css_syntax.asp).

#### Editor de código

Para este laboratorio podrás usar tu editor de código preferido incluso Bloc de notas si lo deseas, yo personalmente prefiero usar Visual Studio Code.

#### Imágenes de muestra

Para el laboratorio puedes usar las imágenes de muestra que encontrarás en el archivo de recursos, o puedes usar tus propios recursos visuales si lo deseas.

## Estructura del documento

Inicie el editor de código, cree un nuevo archivo y guárdelo con el nombre "index" y guárdelo con extensión "html". En ese archivo copie y pegue el siguiente código:

**<!DOCTYPE html>**

**<html>**

**<head>**

**<title>Kiro - VFx Artist</title>**

**</head>**

**<body>**

**</body>**

**</html>**

Todos los documentos HTML deben iniciar con la etiqueta **<DocType>**, en ella se registra la versión y el DTD utilizado, lo que ayuda al navegador a mostrar correctamente la página.

La declaración **<! DOCTYPE html>** define este documento como HTML5. El elemento **<html>** es el elemento raíz de una página HTML. El elemento **<head>** contiene metainformación sobre el documento (se usan como base para los buscadores). El elemento **<title>** especifica un título para el documento, se mostrará en la pestaña del navegador. El elemento **<body>** contiene el contenido de la página visible.

## Preparando la diagramación

#### Contenedor general

El archivo index.html, ubique la etiqueta **<body>** y cree una etiqueta div, con el atributo ID definido como "wraper". Debe ser igual al siguiente código:

**<body>**

**<!-- INICIO wrapper -->**

**<div id="wrapper">**

**</div>**

**<!-- FIN wrapper -->**

**</body>**

La etiqueta [<div>](https://www.w3schools.com/tags/tag_div.asp) corresponde a una división "lógica" del contenido, que lo separa en partes sin diferenciarlas por jerarquía o categorías. Este "wrapper" se utiliza comúnmente para centrar el contenido asignándole un ancho fijo y márgenes automáticas a los dos lados.

#### Encabezado de la página

Dentro de la etiqueta [<header>](https://www.w3schools.com/tags/tag_header.asp) siempre se incluyen: la identidad, la navegación, las redes... y, como opcional, podemos incluir el campo de texto usado para búsquedas.

**<body>**

**<!-- INICIO wrapper -->**

**<div id="wrapper">**

**<!-- INICIO header -->**

**<div id="header">**

**<header>**

**<!-- INICIO Identidad -->**

**<div id="identidad">**

**</div>**

**<!-- FIN Identidad -->**

**<!-- INICIO Redes -->**

**<div id="redes">**

**</div>**

**<!-- FIN Redes -->**

**<!-- INICIO menu principal -->**

**<div id="menu">**

**</div>**

**<!-- FIN Menu Principal -->**

**</header>**

**</div>**

**<!-- FIN header -->**

**</div>**

**<!-- FIN wrapper -->**

**</body>**

La identidad debe ser primero. Es lo más importante de la página, pues representa la propiedad sobre el sitio web y sobre sus contenidos.

DEBAJO del header, por medio de la etiqueta [<main>](https://www.w3schools.com/tags/tag_main.asp) vamos a establecer el espacio del documento que contiene los contenidos principales. Copie y pegue el siguiente código y asegúrese que que quede exactamente igual.

**<!-- FIN header -->**

**<!-- INICIO contenidos -->**

**<div id="contenidos">**

**<main>**

**</main>**

**</div>**

**<!-- FIN Contenidos -->**

**</div>**

**<!-- FIN wrapper -->**

**</body>**

Ahora, dentro de la etiqueta **<main>** se van a incluir varias etiquetas **<div>** para separar todos y cada uno de los proyectos (como secciones de "lógicas" de contenidos específicos).

**<!-- FIN header -->**

**<!-- INICIO contenidos -->**

**<div id="contenidos">**

**<main>**

**<!-- INICIO proyecto Work A -->**

**<div id="work1" class="proyecto">**

**</div>**

**<!-- FIN proyecto Work A -->**

**<!-- INICIO proyecto Work B -->**

**<div id="work2" class="proyecto">**

**</div>**

**<!-- FIN proyecto Work B -->**

**</main>**

**</div>**

**<!-- FIN Contenidos -->**

**</div>**

**<!-- FIN wrapper -->**

**</body>**

Cada proyecto contendrá un menú con la lista de miniaturas para seleccionar las imágenes de cada proyecto y además incluiremos las versiones en de alta resolución para cada una.

Para construir el menú usaremos la etiqueta <menu>, la cual es usada para crear herramientas útiles para los contenidos (por ejemplo los botones para compartir, like, guardar o eliminar de una publicación en facebook o twitter)

**<!-- FIN header -->**

**<!-- INICIO contenidos -->**

**<div id="contenidos">**

**<main>**

**<!-- INICIO proyecto Work A -->**

**<div id="work1" class="proyecto">**

**<!-- INICIO Lista Imagenes proyecto Work A -->**

**<div class="listaImagenesProyecto">**

**<menu>**

**</menu>**

**</div>**

**<!-- FIN Lista Imagenes proyecto Work A -->**

**</div>**

**<!-- FIN proyecto Work A -->**

**<!-- INICIO proyecto Work B -->**

**<div id="work2" class="proyecto">**

**<!-- INICIO Lista Imagenes proyecto Work B -->**

**<div class="listaImagenesProyecto">**

**<menu>**

**</menu>**

**</div>**

**<!-- FIN Lista Imagenes proyecto Work B -->**

**</div>**

**<!-- FIN proyecto Work B -->**

**</main>**

**</div>**

**<!-- FIN Contenidos -->**

**</div>**

**<!-- FIN wrapper -->**

**</body>**

Debajo de la sección del código de los contenidos, vamos a incluir el pié de página. En el pie de página se suele incluir toda la información de contacto. En ocasiones un menú de navegación rápida y, si no están en el encabezado de la página, los links a las redes sociales.

Para el pié de página se utiliza la etiqueta [<footer>](https://www.w3schools.com/tags/tag_footer.asp)

**<!-- FIN Contenidos -->**

**<!-- INICIO Footer -->**

**<div id="footer">**

**<footer>**

**</footer>**

**</div>**

**<!-- FIN Footer -->**

**</div>**

**<!-- FIN wrapper -->**

**</body>**

Para esta página, pondremos un texto de presentación del sitio y un formulario de contacto distribuido en dos etiquetas **<div>.**

**<!-- FIN Contenidos -->**

**<!-- INICIO Footer -->**

**<div id="footer">**

**<footer>**

**<!-- INICIO About -->**

**<div id="infoAbout" class="infoFooter">**

**</div>**

**<!-- FIN About -->**

**<!-- INICIO Contacto -->**

**<div id="infoContact" class="infoFooter">**

**</div>**

**<!-- FIN Contacto -->**

**</footer>**

**</div>**

**<!-- FIN Footer -->**

**</div>**

**<!-- FIN wrapper -->**

**</body>**

## Trabajemos el HTML

**1.** Incluyamos el código de la identidad con [HTML Semántico](https://es.wikipedia.org/wiki/HTML_semántico) dentro del **<div>** con el id="identidad". Debe finalizar mas o menos así:

**<div id="identidad">**

**<h1>**

**<a href="#">**

**<span>**

**Kiro**

**</span>**

**<strong>**

**VFX Artist**

**</strong>**

**</a>**

**</h1>**

**</div>**

Cuando una etiqueta **<header>** es la primera del HTML, los buscadores asumen que es el encabezado de la página (a diferencia de un encabezado dentro de una [<article>](https://www.w3schools.com/tags/tag_article.asp)). Si el encabezado tiene una etiqueta [<h1>](https://www.w3schools.com/tags/tag_hn.asp), este se convierte en el contenido más importante y por ende, los buscadores asumen que es lo principal a rastrear de la página. Para la identidad de la página, usaremos un **<h1>** para darle mayor relevancia.

Por otro lado, en la mayoría de las páginas se acostumbra a asignarle un vínculo a la identidad que lleva al inicio del sitio o home. Para esto se utilizará una etiqueta [<a>](https://www.w3schools.com/tags/tag_a.asp). El vínculo en este caso podrá tener una de 3 opciones para lograrlo:

* Debería apuntar a la dirección del mismo [dominio del sitio](https://es.wikipedia.org/wiki/Dominio_de_Internet) p.ejem: "www.paginaweb.com"
* Debería apuntar al nombre del archivo de p. ejem: "index.html"
* Debería apuntar a la carpeta raíz del sitio, usando el caracter "/", que representa el acceso a carpetas en el estándar de una [URL](https://es.wikipedia.org/wiki/Localizador_de_recursos_uniforme).

Para la semántica del HTML tenemos el texto dentro del título y dentro de la etiqueta <a>, pero como la marca es una imágen, no podremos mostrar el nombre en el diseño cuando se visualice en el navegador. Para esto usaremos una etiqueta [<span>](https://www.w3schools.com/tags/tag_span.asp) y una etiqueta [<strong>](https://www.w3schools.com/tags/tag_strong.asp) para darle más importancia, las cuales nos permitirá ocultar el texto y aún así mostrar la imagen.

**2.** Incluyamos el HTML semántico del menú de redes sociales. Debe terminar más o menos así:

**<div id="redes">**

**<menu type="toolbar">**

**<li>**

**<a href="#" class="btnFacebook">**

**<span>**

**Facebook**

**</span>**

**</a>**

**</li>**

**<li>**

**<a href="#" class="btnFlickr">**

**<span>**

**Flickr**

**</span>**

**</a>**

**</li>**

**</menu>**

**</div>**

La etiqueta **<menu>** se comporta como una lista de elementos, lo cual requiere [<li>](https://www.w3schools.com/tags/tag_li.asp) (items de lista en español, o list items en inglés) para definir cada una de las opciones de las redes. Si se usan las etiquetas <span> dentro de las etiquetas **<a>**, se pueden ocultar los textos para mostrar sólo íconos.

**3.** Incluyamos el HTML semántico del menú de NAVEGACIÓN principal, para lo cual usaremos la etiqueta [<nav>](https://www.w3schools.com/tags/tag_nav.asp) . Debe terminar más o menos así:

**<div id="menu">**

**<h3 class="tituloMenuFlechaRight">**

**Menú**

**</h3>**

**<nav>**

**<ul>**

**<li>**

**<a href="#">**

**<span>**

**Work**

**</span>**

**</a>**

**</li>**

**<li>**

**<a href="#">**

**<span>**

**Personal**

**</span>**

**</a>**

**</li>**

**<li>**

**<a href="#">**

**<span>**

**Reel**

**</span>**

**</a>**

**</li>**

**<li>**

**<a href="#">**

**<span>**

**Making of**

**</span>**

**</a>**

**</li>**

**<li>**

**<a href="#">**

**<span>**

**Mentions**

**</span>**

**</a>**

La etiqueta **<h3>** nos ayudará a definir el texto que se muestra cuando el menú está cerrado. La usamos como título, para que los buscadores identifiquen su relevancia en el header.

La etiqueta **<nav>** puede contener sólo etiquetas de vínculos (**<a>**), o una lista de vínculos. Aquí también podemos usar los **<span>** para ocultar los textos si quisieramos mostrar sólo íconos.

**4.** Para crear menús desplegables, se crean listas de items anidadas, es decir, que dentro de un item de la lista principal, se crea otra lista con sus propios items. Para nuestro menú desplegable, ubicaremos las listas internas DEBAJO de la etiqueta **<a>** de cada item del menú principal:

**<div id="menu">**

**<h3 class="tituloMenuFlechaRight">**

**Menú**

**</h3>**

**<nav>**

**<ul>**

**<li>**

**<a href="#">**

**<span>**

**Work**

**</span>**

**</a>**

**<ul>**

**<li>**

**</li>**

**</ul>**

**</li>**

**<li>**

**<a href="#">**

**<span>**

**Personal**

**</span>**

**</a>**

**<ul>**

**<li>**

**</li>**

**</ul>**

**</li>**

**<!-- Desde aquí se insertan todos los demás -->**

**</ul>**

**</nav>**

**</div>**

Recuerde que cada item será un elemento en la lista de proyectos, así como se ve en el mockup (La primera imagen del laboratorio). En el código sólo se tiene un elemento **<li>**, pero deben ser seis en total uno por cada proyecto, tres arriba y tres abajo del botón del menú principal.

**5.** Para incluir los proyectos, dentro de cada <li> creamos dos elementos:

1. Una etiqueta [<img />](https://www.w3schools.com/html/html_images.asp), que será el fondo del submenu y sólo podrá verse cuando el mouse esté en sobre el item del proyecto.
2. Una etiqueta **<a>**, que tendrá la miniatura del proyecto y los textos del nombre del proyecto.

**<div id="menu">**

**<h3 class="tituloMenuFlechaRight">**

**Menú**

**</h3>**

**<nav>**

**<ul>**

**<li>**

**<a href="#">**

**<span>**

**Work**

**</span>**

**</a>**

**<ul>**

**<li>**

**<img   
src="uploads/med-crush-a.jpg"   
class="fondoItemMenu"/>**

**<a href="#">**

**<img src="uploads/pic-crush-a.jpg"/>**

**<span>**

**Nombre Proyecto A - Work**

**</span>**

**</a>**

**</li>**

**<li>**

**<img   
src="uploads/med-olimpo-a.jpg"   
class="fondoItemMenu"/>**

**<a href="#">**

**<img src="uploads/pic-olimpo-a.jpg"/>**

**<span>**

**Nombre Proyecto B - Work**

**</span>**

**</a>**

**</li>**

**<li>**

**<img   
src="uploads/med-powerade-a.jpg"   
class="fondoItemMenu"/>**

**<a href="#">**

**<img src="uploads/pic-powerade-a.jpg"/>**

**<span>**

**Nombre Proyecto C - Work**

**</span>**

**</a>**

**</li>**

**6.** Ahora incluimos los contenidos del **<footer>**. Primero ponemos el textos del "About" dentro del elemento id="infoAbout"

**<!-- INICIO About -->**

**<div id="infoAbout" class="infoFooter">**

**<h4>**

**<span>**

**about**

**</span>**

**</h4>**

**<p>**

**<strong>**

**Kiro VFX**

**</strong>**

**Texto de presentaci&oacute;n Kiro VFX - Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse vel mauris non lectus sodales suscipit. Morbi laoreet dapibus massa, eu luctus sem pellentesque a. Nunc dictum erat lobortis arcu luctus faucibus. Sed dignissim quam blandit, commodo nunc eget, tempor mauris. Integer a ante et massa commodo gravida. Fusce viverra ante convallis erat dictum hendrerit. In vitae gravida felis.**

**</p>**

**</div>**

**<!-- FIN About -->**

**7.** Ahora incluimos el formulario de contacto. Este código no está completo, así que hace falta incluir el codigo para hacerlo funcionar.

**<!-- INICIO Contacto -->**

**<div id="infoContact" class="infoFooter">**

**<h4>**

**<span>**

**contact**

**</span>**

**</h4>**

**<form>**

**<label for="campAsunto">Asunto:</label>**

**<input type="text" id="campAsunto"  
value="Escriba el asunto del mensaje" />**

**<label for="campMensaje">Mensaje</label>**

**<input type="text" id="campMensaje" />**

**<input type="submit" value="Enviar"/>**

**</form>**

**</div>**

**<!-- FIN Contacto -->**

## Trabajemos la maquetación

#### Vincular la hoja de estilos

Cree un archivo nuevo en el editor de código, guárdelo en la carpeta "css" con el nombre styles y defina su extensión como .CSS (styles.css) y luégo vincúlelo al HTML usando la etiqueta [<link>](https://www.w3schools.com/tags/tag_link.asp) dentro del **<head>** del documento.

Graphical user interface, text

Description automatically generated with medium confidence

**<!DOCTYPE html>**

**<html>**

**<head>**

**<title>Kiro - VFx Artist</title>**

**<link rel="stylesheet" type="text/css" href="css/style.css"/>**

**</head>**

**<body>**

**</body>**

**</html>**

**1.** Dentro del archivo styles.css copie y pegue el siguiente código:

**/\* ----------- INICIO genericos \*/**

**html{**

**width: 100%;**

**height: 100%;**

**}**

**body{**

**margin: 0px;**

**background-color: #000;**

**color: #fff;**

**font-family: "Trebuchet MS", Arial, Helvetica, sans-serif;**

**background-image: url(../images/fondwebpage.png);**

**width: 100%;**

**height: 100%;**

**}**

**header,**

**section,**

**menu,**

**nav,**

**footer,**

**hgroup{**

**margin: 0px;**

**padding: 0px;**

**}**

**a{**

**text-decoration: none;**

**}**

**.ocultar{**

**display:none;**

**}**

**/\* ----------- FIN genericos \*/**

El código anterior asigna una altura y un ancho a la etiqueta **<html>** con las propiedades ["width" y "height"](https://www.w3schools.com/css/css_dimension.asp), luego para la etiqueta **<body>** hace lo mismo (al final de las propiedades).

La propiedad ["margin"](https://www.w3schools.com/css/css_margin.asp) define el espaciado entre elementos, pero para la etiqueta **<body>** define la separación de los contenidos respecto del marco del navegador. La propiedad ["background-color"](https://www.w3schools.com/css/css_background.asp) establece el color de fondo, debe revisar [esta página en W3Schools](https://www.w3schools.com/css/css3_colors.asp), los posibles sistemas de color que puede usar, aunque principalmente se utilizan los hexadecimales. La propiedad ["color"](https://www.w3schools.com/cssref/pr_text_color.asp) establece el color del texto y utiliza los mismos sistemas de color. La propiedad ["background-image"](https://www.w3schools.com/css/css_background.asp) establece que el elemento tendrá una imagen de fondo que se repetirá indefinidamente horizontal y verticalmente. La propiedad ["font-family"](https://www.w3schools.com/css/css_font.asp) define la fuente que se utilizará para presentar los textos.

Luego de las propiedades del **<body>**, asigna a varias etiquetas las propiedades ["margin"](https://www.w3schools.com/css/css_margin.asp) y ["padding"](https://www.w3schools.com/css/css_padding.asp) con valor de 0 y usando el [sistema de medida](https://www.w3schools.com/cssref/css_units.asp) en pixeles. Al hacer esto, los elementos no tienen márgenes interna ni externa.

Para la etiqueta **<a>** se establece sólo la propiedad ["text-decoration"](https://www.w3schools.com/cssref/pr_text_text-decoration.asp), que elimina el subrayado de los vínculos en el documento.

Por último, un [selector tipo "class"](https://www.w3schools.com/cssref/css_selectors.asp) que utiliza la propiedad "display" para ocultar los elementos de la pantalla.

**2.** Debajo de los estílos genéricos, vamos a crear otro grupo de estilos. Dentro de los comentarios asignaremos las propiedades iniciales de wrapper, header, main y footer.

**/\* ----------- FIN genericos \*/**

**/\* ----------- INICIO Diagramacion y contenidos \*/**

**#wrapper{**

**width: 100%;**

**height: 100%;**

**}**

**#header{**

**background-color: #f00;**

**}**

**#contenidos{**

**background-color: #0f0;**

**width: 100%;**

**height: 100%;**

**}**

**#footer{**

**background-color: #00f;**

**position: fixed;**

**bottom: 0px;**

**width: 100%;**

**z-index: 110;**

**}**

**/\* ----------- FIN Diagramacion y contenidos \*/**

Sólo para poder hacerlos visibles, asignaremos un color de fondo a los elementos, así podremos verlos en el navegador. Más adelante cambiaremos esos valores.

Las propiedades del **<footer>** incluyen ["position" con valor "fixed"](https://www.w3schools.com/cssref/pr_class_position.asp), para que su posición se defina respecto del navegador, y al asignarla en conjunto con la propiedad "bottom" le definimos el valor de la posición desde extremo inferior de la ventana. Las dos propiedades combinadas, dejarán el footer fijo en el inferior de la página.

Debido a que se mantiene en la parte inferior de la ventana, podría ocurrir que algunos elementos se ubiquen por encima del footer, para poder controlar la superposición de los elementos, cuando una etiqueta tiene asignada la propiedad position, se usa la propiedad ["z-index"](https://www.w3schools.com/cssref/pr_pos_z-index.asp) para controlar el orden (delante o detrás de otros).

**3.** Ahora se incluyen las propiedades para la identidad corporativa. Por favor copie el siguiente código y péguelo entre el selector del #header y el selector #contenidos:

**#identidad{**

**width: 70px;**

**height: 130px;**

**position: absolute;**

**right: 0px;**

**top: 50%;**

**margin-top: -65px;**

**background-color: #FFF;**

**border-radius: 10px 0px 0px 10px;**

**background-image: url('../images/identidad.png');**

**background-size: contain;**

**background-repeat: no-repeat;**

**background-position: center;**

**transition: all 1s;**

**}**

**#identidad:hover{**

**width: 130px;**

**}**

**#identidad h1{**

**margin:0px;**

**width: 100%;**

**height: 130px;**

**}**

**#identidad h1 a{**

**display: block;**

**width: 100%;**

**height: 130px;**

**}**

**#identidad h1 a span,**

**#identidad h1 a strong{**

**display: none;**

**}**

Para la identidad, asignaremos un alto y ancho en pixeles que luego usaremos de referencia para otras propiedades. La posición de la identidad se logra usando la propiedad "position" con valor absolute (para ubicarla respecto de la etiqueta body) y las propiedades ["right"](https://www.w3schools.com/cssref/pr_pos_right.asp) y ["top"](https://www.w3schools.com/cssref/pr_pos_top.asp) con valores bien planeados. Para este caso, queremos que quede al extremo de la derecha asignando el valor en 0 pixeles y queremos que se ubique centrada verticalmente para lo cual asignaremos un valor de 50%, que tomará la altura de la ventana del navegador (y por ende, también en cualquier pantalla) y la divide en dos para establecer la posición en pixeles.

Pero, la medida en pixeles para "top" se toma de la borde superior del elemento al que se asigna. Eso significa que ese borde superior es el que se ubica a la mitad de la altura de la pantalla, y eso no es lo que queremos. Buscamos que se vea centrada verticalmente.

Para solucionarlo, asignaremos a la margen superior "margin-top" un valor de margen en negativo, lo que en lugar de desplazar el objeto para separarlo, lo desplaza para acercarlo o sobreponerlo. Para centrarlo verticalmente, la margen superior se asignará como la mitad negativa de la altura del contenedor de la identidad, es decir: si mide 130 pixeles de altura, la margen negativa será -65px;

Las esquinas redondeadas se crean con la propiedad ["border-radius"](https://www.w3schools.com/cssref/css3_pr_border-radius.asp), el orden de los valores, define cual es el valor asignado a cada esquina.

La imagen de fondo o "background-image" puede tener asignada una dimensión específica o relativa, usando la propiedad ["background-size"](https://www.w3schools.com/cssref/css3_pr_background-size.asp) con valor "contain" especificaremos que su tamaño siempre se ajuste a las medidas del elemento y que la imagen en sí, siempre se vea completa.

Para cambiar la medida del ancho de la identidad (de 70px a 130px) al pasar el mouse por encima, usaremos la [pseudoclase (:hover)](https://www.w3schools.com/css/css_pseudo_classes.asp) pero para que ocurra el cambio de forma animada, usaremos dentro de #identidad la propiedad ["transition"](https://www.w3schools.com/css/css3_transitions.asp) la cual permite asignar la propiedad específica que podrá ser animada o, como en este caso usando el valor "all", todas las propiedades que cambien su valor, se mostrará el cambio de forma animada.

Las propiedades asignadas para las etiquetas **<h1>** y **<a>** ya las hemos explicado más arriba, así que debe ser fácil de entender por que fueron asignados.

Para ocultar los textos que están dentro de la etiqueta **<span>** y **<strong>** usaremos nuevamente la propiedad "display" con el valor "none".

**4.** Luego de los estilos para la identidad e inmediatamente antes del selector #contenidos, ponga los siguientes estilos correspondientes al menú de redes sociales:

**#redes{**

**position: absolute;**

**right: 0px;**

**top: 65%;**

**width: 70px;**

**margin-left: 62px;**

**border-bottom: 3px solid #666;**

**}**

**#redes li{**

**margin: 0px;**

**padding: 0px;**

**list-style: none;**

**List-style-type: none;**

**float: left;**

**}**

**#redes li a{**

**display: block;**

**width: 24px;**

**height: 24px;**

**margin-top: -10px;**

**}**

**.btnFacebook{**

**background-image: url("../images/redes.png");**

**background-position: left;**

**margin-left: -62px;**

**}**

**.btnFacebook:hover{**

**background-image: url("../images/redes-over.png");**

**}**

**.btnFlickr{**

**background-image: url("../images/redes.png");**

**background-position: -33px;**

**margin-left: -28px;**

**}**

Para #redes, la única propiedad nueva es ["border-botom"](https://www.w3schools.com/cssref/pr_border.asp) que permite asignar al mismo tiempo y siempre en ese orden: El grosor de la línea, el estilo de la línea y el color de la línea. Todas, al borde inferior del elemento.

Luego, para todos los elementos **<li>** dentro de #redes (#redes li) se eliminan las viñetas (figuras geométricas que identifican visual mente a cada elemento de la lista) con la propiedad ["list-style"](https://www.w3schools.com/cssref/pr_list-style.asp) o la propiedad ["list-style-type"](https://www.w3schools.com/cssref/pr_list-style-type.asp) se utilizan propiedades similares, excepto por la propiedad ["float"](https://www.w3schools.com/cssref/pr_class_float.asp) en este caso, con valor "left"; la cual permite ubicar los elementos uno al lado del otro, de izquierda a derecha (si la propiedad tuviera asignado el valor right, los elementos se organizarían de derecha a izquierda).

Como dentro de cada elemento **<li>** tenemos un vínculo **<a>**, ahora debe darle propiedades. La propiedad "display" tiene asignado en estos estilos un valor "block". Con este valor, la propiedad permite presentar un elemento en la pantalla, pero con una característica específica: sólo cuando un elemento tiene el valor "block" tiene la capacidad de utilizar los valores asignados a las propiedades "width" y "height". PERO, en todos los navegadores las etiquetas **<a>** tienen por defecto el valor "inline" para la propiedad "display" es decir que no recibe ni alto ni ancho y por ende, no se puede manipular sus dimensiones.

Cada **<a>** tiene asignado un class que le define el tipo ícono o identidad de la red social a la que representa (.btnFacebook y .btnFlickr) las marcas para estos botones se asignan como imagenes de fondo, pero en lugar de ser imagenes independientes, usamos la estrategia de ["sprites"](https://www.w3schools.com/css/css_image_sprites.asp) en la que un solo archivo de imagen tiene los recursos visuales para cada elemento html con ícono o botón. Del mismo modo, al usar la pseudoclase ":hover" se utiliza el método de "sprites" para mostrar diferente el botón cuando el mouse pasa por encima de él.

## La interacción del menú

**1.** Los estilos del menú principal de navegación los trabajaremos en cinco partes. La primera parte será para cuadrar el título "menú", el cual permitirá abrir el menú cuando se le dé click. Copie y pegue estos estilos debajo de los estilos de #redes y justo arriba del selector #contenidos

**#menu{**

**height: 36px;**

**position: absolute;**

**left: 0px;**

**top: 50%;**

**margin-top: -18px;**

**padding-right: 10px;**

**border-radius: 0px 5px 5px 0px;**

**background-color: #222;**

**z-index: 10;**

**}**

**#menu h3{**

**margin: 0px;**

**width: 40px;**

**height: 26px;**

**padding: 8px 10px 0px 20px;**

**background-position: right;**

**background-repeat: no-repeat;**

**float: left;**

**font-size: 13px;**

**cursor: pointer;**

**}**

**h3.tituloMenuFlechaRight{**

**background-image:url("../images/arrow-tit-menu-1.png");**

**}**

**h3.tituloMenuFlechaLeft{**

**background-image:url("../images/arrow-tit-menu-2.png");**

**}**

El menú también fue pensado para está centrado verticalmente, así que se utiliza la misma estrategia de asignar "position" con valor "absolute" y "top" con valor "50%", luego una margen superior negativa igual a la mitad de la altura del elemento. Y las esquinas redondeadas sólo serán para las del extremo derecho del #menu.

El texto "Menú", al estar dentro de una etiqueta **<h3>** tendrá una margen superior e inferior que el navegador le asigna por defecto, para eliminarla, se debe sobreescribir ese valor asignando el valor de "0px".

Asignarle a este elemento un padding derecho de 20px permitirá que la imagen de fondo se muestre siempre separada del texto.

La propiedad "cursor" define el ícono que el mouse debe mostrar cuando esté sobre el elemento, existen varias versiones de ícono para el mouse, pero para que el usuario identifique que es un elemento clieckeable, se utiliza el valor "pointer" (que corresponde a la mano con un dedo apuntando).

Finalmente, usaremos dos class diferentes para mostrar la flecha de fondo del título: una flecha para cuando esté cerrado (que apunta a la derecha) y una para cuando esté abierto (que apunta a la izquierda). Estos class serán asignados al **<h3>** según el estado del menú cuando se haga click alternadamente.

**2.** Ahora, la segunda parte, que corresponde al menú principal y su distribución horizontal. Inmediatamente después del "h3.tituloMenuFlechaLeft" pegue el siguiente código:

**h3.tituloMenuFlechaLeft{**

**background-image:url("../images/arrow-tit-menu-2.png");**

**}**

**#menu ul{**

**height: 36px;**

**width: 755px;**

**margin: 0px;**

**padding: 0px;**

**margin-left: 100px;**

**list-style: none;**

**list-style-type: none;**

**}**

**#menu ul li{**

**width: 150px;**

**float: left;**

**margin: 0px;**

**padding: 0px;**

**height: 20px;**

**margin: 8px 0px;**

**border-left: 1px dotted #FFF;**

**position: relative;**

**}**

**#menu ul li:first-child{**

**border-left: none;**

**}**

**#menu ul li a{**

**display: block;**

**height: 28px;**

**width: 160px;**

**margin: -8px 0px;**

**text-align: center;**

**padding-top: 8px;**

**font-size: 14px;**

**color:#fff;**

**}**

Como la lista de opciones del menú (el elemento **<ul>**) no tiene la propiedad float, no se ubicará al lado derecho del texto "Menú". En lugar de eso, se ubicará detrás de él. Pero la situación es que sí lo necesitamos a lado derecho... pero si usamos "float" para hacerlo, tendremos problemas al momento de tratar de abrir el menú, pues se verá un salto de la lista en lugar de una animación fluida. Si está detrás del texto, ahora usaremos la margen izquierda para desplazar la lista al lado derecho del título.

Para mostrar el menú horizontal, tomamos el valor del "width" de cada item de la lista, y lo multiplicamos por el número de items (5 itmes multiplicado por 150px = 750px). Pero como los items de la lista tienen un borde izquierdo de 1 pixel, eso les agrega un pixel más a la medida del ancho, lo que significa que en lugar de 150px, realmente miden 151px de ancho cada uno. [En esta página del W3school](https://www.w3schools.com/css/css_boxmodel.asp) puede encontrar más información sobre el modelo de caja de los elementos HTML.

Cuando establezcamos la programación para abrir y cerrar el menú, intercambiaremos el valor del "width" alternadamente entre el ancho que muestra las opciones horizontales y 0px.

A cada item del lista del menú principal le definiremos la medida del ancho de cada botón. Con la propiedad "float" los organizaremos horizontalmente. Por último, para mostrar dónde inicia y dónde termina una opción, usaremos una línea punteada blanca al lado izquierdo de cada una.

Pero, la primera opción no necesita separarse de una anterior (la primera no tiene opción a la izquierda), así que se debe quitar la línea punteada UNICAMENTE a la primera opción del menú. Para lograrlo usaremos el [pseudoelemento ":first-child"](https://www.w3schools.com/css/css_pseudo_elements.asp) con el valor "none" asignado a la misma propiedad "border-left" para sobre escribirla.

**3.** Para poder mostrar el menú cerrado, crearemos un class nuevo que se incluirá por programación javascript al elemento **<ul>**. Igualmente para mostrarlo abierto de nuevo.

**#menu ul{**

**height: 36px;**

**width: px;**

**margin: 0px;**

**padding: 0px;**

**margin-left: 100px;**

**list-style: none;**

**list-style-type: none;**

**overflow: hidden;**

**transition: width 2s;**

**}**

**#menu ul.abierto{**

**width: 755px;**

**}**

**#menu ul.abierto:hover{**

**overflow: visible;**

**}**

Pero al cambiar el valor del "width" por 0px para que esté cerrado, las opciones del submenú y las miniaturas de los proyectos serán visibles por fuera de la lista, organizados uno debajo de otro. Para evitar que sean visibles se utiliza la propiedad ["overflow" con valor "hidden"](https://www.w3schools.com/cssref/pr_pos_overflow.asp), que ocultan cualquier elemento que sea más grande que la etiqueta al que se le asigna esta propiedad.

Así que primero agregaremos una propiedad "overflow" con valor "hidden" al **<ul>** dentro de #menu.

Por otro lado, al estar abierto, los submenú de proyectos serán más grandes (su altura será animada verticalmente y quedarían ocultos) así que debe cambiarse el valor del "overflow" a "visible" para que los elementos de mayor dimensión sean visibles por arriba y abajo del menú horizontal.

**4.** La tercera parte será asignar la función que despliega o colapsa el menú con el evento click del mouse.

Descargue la referencia rápida de jQuery para iniciar con los mínimos conocimientos que le permitirán sacar provecho de la interacción o la animación con la librería de jQuery.

Para empezar, en su editor de código cree un archivo javascript y guárdelo en la carpeta "js" con el nombre "menu.js".

Puede usar [jQuery](https://www.w3schools.com/jquery/jquery_intro.asp) descargando la librería o conectarla desde un [CDN](https://es.wikipedia.org/wiki/Red_de_distribución_de_contenidos), en [esta página de w3Schools](https://www.w3schools.com/jquery/jquery_get_started.asp) se explica cómo hacer cualquiera de las dos.Por último, vincule su archivo debajo de la librería de jQuery al HTML usando la etiqueta **<script>** de la siguiente forma:

**<!DOCTYPE html>**

**<html>**

**<head>**

**<title>Kiro - VFx Artist</title>**

**<link rel="stylesheet" type="text/css" href="css/style.css"/>**

**<script type="text/javascript" src="js/jquery-2.1.4.min.js">  
</script> <!-- Si la descargó se vincula así -->**

**<script type="text/javascript" src="js/menu.js"></script>**

**</head>**

**<body>**

**. . .**

**. . .**

**. . .**

**</body>**

**</html>**

Dentro del archivo "menu.js" vamos a incluir la función de jQuery para que ejecute los comandos cuando la página termine de cargar, para lograrlo usaremos la sintaxis de jQuery y nos apoyamos en el evento "ready" para el documento. Dele una mirada a estos conceptos en [esta página del w3schools](https://www.w3schools.com/jquery/jquery_syntax.asp). Al terminar de leerlo, copie y pegue el siguiente código:

// JavaScript Document

**$(document).ready(function ($) {**

**// Los métodos de jQuery van aquí...**

**});**

Dentro de la función del evento "ready", primero crearemos una variable de control, que nos permita conocer el estado actual del menú (true para cuando esté abierto o false para cuando esté cerrado)

// JavaScript Document

**$(document).ready(function ($) {**

**var abiertomenu = false;**

**});**

Debajo de la variable, vamos a crear una función que invocaremos cada vez que se haga click en el título "menú". El parámetro "event" será a través del cual se pasará el objeto de datos del listener:

// JavaScript Document

**$(document).ready(function ($) {**

**var abiertomenu = false;**

**function abrirmenu(event) {**

**}**

**});**

Debajo de la función creamos un trigger, que corresponde al listener del evento "click" e invoca la función "abrirmenu":

// JavaScript Document

**$(document).ready(function ($) {**

**var abiertomenu = false;**

**function abrirmenu(event) {**

**}**

**$("#menu h3").on("click", abrirmenu);**

**});**

Dentro de la función "abrirmenu" incluimos una condición que evalúe el valor de verdad de la variable de control:

// JavaScript Document

**$(document).ready(function ($) {**

**var abiertomenu = false;**

**function abrirmenu(event) {**

**if (abiertomenu) {**

**} else {**

**}**

**}**

**$("#menu h3").on("click", abrirmenu);**

**});**

Dentro del "if", evaluaremos si la variable de control posee un valor "true". Si la evaluación es falsa, vamos a agregarle a la lista del menú principal el class "abierto" que define la dimensión de su ancho en 755px, pero si es verdadera debemos retirársela. Luego, controlamos el valor de la variable "abiertomenu" estableciendo el valor contrario al que entra en la condición.

Depués, fuera de la condición, pero dentro de la función incluimos el método ["toggleClass"](https://api.jquery.com/toggleclass/) que permite agregar o quitar un class a un elemento alternadamente. En este caso, nos permite cambiarle la imagen de fondo al título "Menu" para que apunte a la dirección opuesta con cada click.

// JavaScript Document

**$(document).ready(function ($) {**

**var abiertomenu = false;**

**function abrirmenu(event) {**

**if (abiertomenu) {**

**$("#menu nav>ul").removeClass("abierto");**

**abiertomenu = false;**

**} else {**

**$("#menu nav>ul").addClass("abierto");**

**abiertomenu = true;**

**}**

**$("#menu h3").toggleClass("tituloMenuFlechaLeft");**

**}**

**$("#menu h3").on("click", abrirmenu);**

**});**

**5.** Ya podemos eliminar los colores de fondo. Por favor elimine la propiedad "background-color" o cambie su valor para los siguientes elementos:

**/\* . . . \*/**

**#header{**

**background-color: #f00; /\* eliminar la propiedad \*/**

**}**

**/\* . . . \*/**

**#contenidos{**

**background-color: #0f0; /\* eliminar la propiedad \*/**

**width: 100%;**

**height: 100%;**

**}**

**#footer{**

**background-color: #00f; /\* eliminar la propiedad \*/**

**}**

**6.** La cuarta parte es establecer las propiedades del submenú con las miniaturas del proyecto dispuestas verticalmente. Copie y pegue el siguiente código justo encima del elemento #contenidos:

**#menu ul li ul{**

**width: 210px !important;**

**height: 26px;**

**box-sizing: border-box;**

**position:absolute;**

**top: -28px;**

**left: -25px;**

**z-index: -1;**

**display: block;**

**opacity: 0;**

**margin: 0px;**

**padding: 10px 3px;**

**border-radius: 10px;**

**overflow: hidden !important;**

**background: linear-gradient(to top, #000000, #222222);**

**border: 1px solid #444;**

**transition: all 1s;**

**}**

La lista del submenú de proyectos debe tener un ancho en pixeles, pero cuando asignamos la medida de 755px o de 0px al **<ul>** dentro de #menu, el segundo **<ul>** (el de adentro del **<li>**) también la recibe y se abre o se cierra. Para que la propiedad de un elemento no sea cambiada o sobre escrita se le asigna "important!" al valor. Así que, sin importar el ancho que se asigne genérico al **<ul>** exterior, el interior mantendrá la medida de 200px. De paso, la altura inicial del submenú será de 26px para que pueda ser animada.

Como en el código HTML la lista **<ul>** se encuentra debajo de la etiqueta **<a>**, la visualización por defecto será ubicarla DEBAJO del botón. Pero el submenú debería arrancar desde el borde superior de la barra de menú, así que es importante DESPLAZAR el submenú durante su visualización desde debajo del botón hasta el borde superior. Para lograrlo, usaremos la propiedad "position" con el valor "absolute" lo que permite moverlo de posición, respecto de su ubicación por defecto cambiando los valores de "top" y "left". Para este caso, "top" con valor negativo de -28px (casi los mismos de la altura del menú horizontal).

La propiedad "z-index" con valor "-1" indica que el **<ul>** se ubicará por detrás del botón del menú.

La propiedad "opacity" se utiliza para definir la transparencia de un elemento. Para dejarlo completamente transparente se usa "0", para no tener ningún nivel de transparencia se utiliza "1" y los decimales entre 0 y 1 corresponderán a los porcentajes entre 0% y 100%.

Desde la versión CSS 3.0 la propiedad "background" puede crear degradados por medio de funciones ["gradient"](https://www.w3schools.com/css/css3_gradients.asp) que se crean durante la visualización del HTML en el navegador.

Ahora, por favor copie y pegue el siguiente código debajo del selector "#menu ul li ul":

**#menu ul li ul{**

**. . . .**

**}**

**#menu ul li:hover ul{**

**display:block;**

**height: 575px;**

**top: -275px;**

**opacity: 1;**

**}**

**#menu ul li ul li{**

**border: none;**

**display: block;**

**float: none;**

**margin:0px 0px 6px 0px;**

**background-color: #000;**

**border: 1px solid #666;**

**padding: 3px;**

**width: auto;**

**height: auto;**

**opacity: 1;**

**transition: all 0.3s;**

**}**

Para desplegar el submenú al pasar el mouse sobre el botón del menú principal, usamos un selector que será la combinación del :hover (pseudoclase) y sus elementos internos. Esto normalmente significa que de todos los elementos después del ":hover" sólo el último recibirá las propiedades descritos dentro de las llaves "{ ... }". Las que para este caso, definen que cambia su posición vertical restando 300 pixeles a la ptopiedad "top", la altura y cambiando la transparencia. Si tenemos en cuenta que el **<ul>** inicialmente tiene asignada la propiedad "transition", todas las propiedades cambiarán animadas en 1 segundo.

Luego cambian las propiedades para los items de lista del submenú:

Primero, re-seteamos los bordes de los **<li>** del submenú, sobreescribiendo el valor de la propiedad a "none". También debemos invalidar la propiedad "float" que heredó de los items del menú principal, sobreescribiendo su valor a "none". Y por último, redefinir las márgenes del item.

Por último, hay que separar el tercero item respecto del cuarto del submenú, para dejar un espacio entre los dos que permita ver el botón del menú principal sobre el que está el mouse ubicado:

**#menu ul li ul li:nth-child(3){**

**margin-bottom: 60px;**

**}**

**7.** Ahora la quinta parte: las propiedades de los elementos internos del item de lista del submenú. Primero las imágenes de fondo, que solo se mostrarán cuando pase el mouse sobre la miniatura del proyecto:

**#menu ul li ul li img{**

**margin-left: -7px;**

**position: absolute;**

**}**

**#menu ul li ul li img.fondoItemMenu{**

**z-index: -20;**

**display: none;**

**transition: opacity 1s;**

**opacity: 0;**

**}**

**#menu ul li ul li:nth-child(1) .fondoItemMenu{**

**margin-top: -15px;**

**}**

**#menu ul li ul li:nth-child(2) .fondoItemMenu{**

**margin-top: -97px;**

**}**

**#menu ul li ul li:nth-child(3) .fondoItemMenu{**

**margin-top: -180px;**

**}**

**#menu ul li ul li:nth-child(4) .fondoItemMenu{**

**margin-top: -317px;**

**}**

**#menu ul li ul li:nth-child(5) .fondoItemMenu{**

**margin-top: -400px;**

**}**

**#menu ul li ul li:nth-child(6) .fondoItemMenu{**

**margin-top: -483px;**

**}**

La propiedad ["nth-child" es un pseudoelemento](https://www.w3schools.com/cssref/sel_nth-child.asp) y se utiliza para asignar propiedades a etiquetas que se encuentran dentro de otras identificándolas por su posición y orden. Se identifican usando el número dentro de los paréntesis, asignando el número del elemento específico al que se quieren dar las propiedades. Para este caso, la propiedad que queremos sobreescribir es la margen superior, para que el elemento pueda ubicarse alineada con el marco del submenú, y como cada uno de los items se encuentra dispuesto verticalmente, entre más abajo en la lista, mayor el número negativo para que se ubique más arriba en el diseño.

**8.** Ahora los botones con las miniaturas de los proyectos:

**#menu ul li ul li a{**

**display:block;**

**position: relative;**

**z-index: 10;**

**height: auto !important;**

**width: auto !important;**

**margin:0px !important;**

**padding: 0px !important;**

**}**

**#menu ul li ul li a img{**

**float:none;**

**border: none;**

**margin:0px !important;**

**position:static;**

**}**

**#menu ul li ul:hover li{**

**opacity: 0.1;**

**}**

**#menu ul li ul li:hover{**

**opacity: 1;**

**}**

**#menu ul li ul:hover li:hover img.fondoItemMenu{**

**display: block;**

**opacity: 1;**

**}**

Como ocurrió anteriormente, la etiqueta **<a>** de la lista del submenú primero recibe las propiedades que fueron asignadas a la etiqueta **<a>** del menú de navegación principal. Así que primero debemos sobreescribir las propiedades pero para que no se restauren, usamos "important!". Igual pasa con las propiedades de la etiqueta **<img />** que contiene por dentro, también deben ser sobreescritas y redefinidas... sobre todo la propiedad "position" que ahora debemos cambiar a "static", que es el valor que tienen todos los elementos por defecto para dicha propiedad.

Luego, se crea un selector para que al ubicar el mouse sobre el submenú, los items de los proyectos se muestran con transparencia del 10%. Pero esa misma transparencia se sobreescribe con el siguiente selector que le cambia el valor a 1 cuando el mouse está sobre un item específico de la lista de proyectos del submenú.

El último selector combina las dos pseudoclases ":hover" (sobre el submenú Y sobre el item de la lista) para que la imagen que está dentro del item, pero fuera de la etiqueta **<a>** se haga visible con "display:block" y cambie la transparencia a 1.

Como podrá notar, las propiedades cambian su apariencia de forma animada debido a la propiedad "transition: all 1s" en dónde el tiempo especificado es 1 segundo para animar cualquier cambio.

## Estilos del footer

**1.** Ahora vamos a darle estilos al footer. Vamos a organizar las dos secciones del contenido distribuidas horizontalmente y con la mitad del ancho de la página.Como los **<div>** dentro del footer tienen el class ".infoFooter" a ambos se les puede asginar el mismo "width" del 49% lo que deja un 1% restante para usar como margen. con "float" se organizarán horizontalmente.La justificación de los textos dentro de los títulos **<h4>** permitirá ubicar los hacia el eje central de la página, de acuerdo al diseño. También se sobre escribe las márgenes que el navegador les dá por defecto. Por último, queremos que los contenido de cada sección del **<footer>** estén ocultos, de modo que al hacer click sobre los títulos podamos desplegarlos. Para que sea evidente que son clickeables asignaremos "pointer" a la propiedad "cursor".

**#footer{**

**. . .**

**. . .**

**}**

**#footer .infoFooter{**

**width: 49%;**

**float: left;**

**}**

**#infoAbout{**

**margin-right: 1%;**

**background-color: #000;**

**}**

**#infoContact{**

**margin-left: 1%;**

**background-color: #000;**

**}**

**#infoAbout h4{**

**color: #999;**

**margin: 5px;**

**text-align: right;**

**cursor: pointer;**

**}**

**#infoContact h4{**

**color: #999;**

**margin: 5px;**

**text-align: left;**

**cursor: pointer;**

**}**

Luego de tener los dos contenedores dispuestos como en el diseño, debemos crear la flecha roja al lado de cada título. Confirme que sabe cuáles revisando en el mockup. Para las flechas, usaremos los psuedo-elementos [::before](https://www.w3schools.com/cssref/sel_before.asp) y [::after](https://www.w3schools.com/cssref/sel_after.asp), las cuales permiten incluir contenido dentro de una etiqueta ANTES o DESPUÉS del contenido que ya tiene, en otras palabras, al inicio o al final del elemento.

Usando la propiedad "content" podemos definir el contenido (que puede ser texto, imágenes o los valores de los atributos que posee esa misma etiqueta). El psuedo-elemento de la etiqueta recibe muchas (pero NO TODAS) de las propiedades CSS y aquellas que se incluyan dentro de las llaves, serán asignadas al contenido que esté dentro de la propiedad "content". Por ejemplo, en este caso, no incluimos un texto específico, sino la cadena de texto vacía, por ende, no mostrará nada de contenido, pero recibe la propiedad "display" que en este caso usamos con el valor "inline-block", pero también podría ser "block".

Al usar "::after" el pseudo-elemento se incluye después del texto, y "inline-block" le permite estar en el mismo renglón, comportándose como un caracter de texto... pero que se ve afectado por "width" y "height" que nos permite visualizarlo como un cuadrado de color rojo.

si asignamos las propiedades de borde a un elemento usando "border", luego podemos cambiar alguno de los valores para alguno de los atributos específicos. Por ejemplo, aquí estamos usando la propiedad "border-top-color" para sobre escribir el color de la línea y pasarlo de rojo a transparente.

Ahora que nuestro cuadrado rojo ya no tiene color en el border superior ni el derecho, se ve como un vértice o ángulo, para que se visualice como una flecha, solo hace falta rotarlo 45 grados y lo dejará apuntando hacia la izquierda.

**#footer{**

**. . .**

**. . .**

**}**

**#infoAbout{**

**. . .**

**}**

**#infoContact{**

**. . .**

**}**

**#infoAbout h4{**

**. . .**

**}**

**#infoAbout h4 span::after{**

**content: "";**

**display: inline-block;**

**width: 6px;**

**height: 6px;**

**border: 1px solid #F00;**

**border-top-color: transparent;**

**border-right-color: transparent;**

**transform: rotate(45deg);**

**}**

**#infoAbout p{**

**display: none;**

**margin: 20px;**

**font-size: 11px;**

**}**

**#infoAbout p strong{**

**display: block;**

**text-align: right;**

**margin-bottom: 10px;**

**}**

Ahora, de la misma forma que lo hicimos arriba para el título de la presentación de la página, le damos valores similares a los elementos del formulario de contacto. Esto lo hará ver similar, pero con sutiles diferencias: la flecha se encuentra a la izquierda por usar el pesudo-elemento "::before"... y la misma ahora apunta en la otra dirección debido a que se rota -45 grados... el borde de color transparente es el izquierdo y no el derecho.

**#infoContact h4{**

**. . .**

**}**

**#infoContact h4 span::before{**

**content: "";**

**display: inline-block;**

**width: 6px;**

**height: 6px;**

**border: 1px solid #F00;**

**border-top-color: transparent;**

**border-left-color: transparent;**

**transform: rotate(-45deg);**

**}**

**#infoContact form{**

**display: none;**

**margin: 20px;**

**}**

**#infoContact \*{**

**margin-left: 20px;**

**}**

## Abrir y cerrar la información del footer

Para abrir y cerrar la información del footer, vamos a crear dentro del archivo de javascript una función que sirva para ambos títulos por igual.

**1.** Primero, debajo de la función "abrirmenu(event)" pero fuera de ella, vamos a crear una nueva función llamada "abririnfofooter"

// JavaScript Document

**$(document).ready(function ($) {**

**var abiertomenu = false;**

**function abrirmenu(event) {**

**. . .**

**}**

**function abririnfofooter(event){**

**$(this).next().slideToggle();**

**}**

**$("#menu h3").on("click", abrirmenu);**

**});**

Dentro de la función "abririnfofooter" vamos a trabajar con el objeto que nos entrega el parámetro ["event"](https://www.w3schools.com/jquery/jquery_events.asp). Si usamos "$(this)" le estamos definiendo a jQuery que los siguientes métodos aplican específicamente al elemento que llegó por el parámetro "event", que fue el mismo que disparó el evento.

Luego, usamos el método [".next()"](https://www.w3schools.com/jquery/traversing_next.asp) para que use de referencia el elemento "$(this)" para [encontrar el elemento HTML que sigue en el código](https://www.w3schools.com/jquery/jquery_traversing.asp) (debajo de él).

**2.** Por último, debajo del trigger que abre el menú, creamos un trigger que se asigne a **cualquier** elemento **<h4>** dentro del "#footer", el listener al evento click para que invoque la función "abririnfofooter".

// JavaScript Document

**$(document).ready(function ($) {**

**var abiertomenu = false;**

**function abrirmenu(event) {**

**. . .**

**}**

**function abririnfofooter(event){**

**$(this).next().slideToggle();**

**}**

**$("#menu h3").on("click", abrirmenu);**

**$("#footer h4").on("click", abririnfofooter);**

**});**

### Por ahora, aquí termina este taller.

Espero que fuera sencillo desarrollarlo y aprendieran mucho con cada paso.

Para terminar el home con todas sus funcionalidades, debe desarrollar la siguiente parte.