

CROQUIER

URL del proyecto: <https://croquier.netlify.app/index.html>

Memoria – Proyecto final

Diseño web – CEI

Florencia Mejliker

El proyecto tiene el objetivo poder plasmar todo lo estudiado en el modulo de diseño y programación web. En este caso elegí desarrollar una web de un artista que realiza ilustraciones, para poder mostrar un poco el arte y contar acerca de la historia del mismo.

El proyecto tiene 4 secciones. Comenzando por la página principal (index.html) la cual muestra un header con el titulo principal y un Nav que contiene las 3 secciones restantes.

Luego en el cuerpo de la página se encuentran 3 imágenes como fondo, que representan cada una de las secciones y que a la vez se puede navegar desde las mismas. Decidí agregarle un hover al pasar el cursor por encima para que se pueda diferenciar las 3 partes por separado.

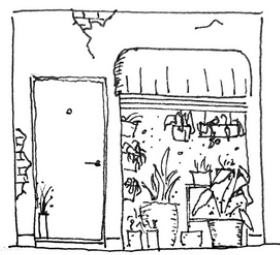
En el footer decidí poner dos enlaces a las redes sociales del artista, para que quienes entren puedan navegar y conocer un poco más al mismo.

Croquier

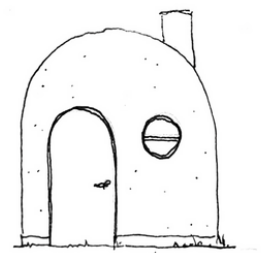
[Sobre croquier](#) [Catálogo](#) [Contacto](#)



sobre croquier



catálogo



contacto

Esta sección en el CSS fue lo primero que estilicé, las partes que compartía con las otras secciones las estilice de manera genera, como por ejemplo:

```
*{
  margin: 0;
  padding: 0;
}
body{
  height: 100vh;
  font-family: Lato, sans-serif;
  font-weight: 300;
  background-color: #ffffff;
}
header{
  padding: 20px;
  width: 100%;
  display: flex;
  justify-content: space-between;
  align-items: center;
  position: fixed;
  z-index: 1;
  background-color: white;
}
header h1 a{
  text-decoration: none;
  font-family: 'Cutive Mono', monospace;
  color: #333;
  font-size: 1.5em;
  letter-spacing: -4px;
}
```

```
nav ul{
  display: flex;
  justify-content: center;
  align-items: center;
  list-style: none;
  margin-right: 30px;
}
nav li{
  margin-right: 30px;
  flex-grow: 1;
}
nav li a{
  color: #333;
  text-decoration: none;
  display: block;
  margin-right: 10px;
  font-weight: 400;
  cursor: pointer;
}
```

Luego para poder diferenciar la estilizacion entre las diferentes secciones, decidí ponerle una clase a los body, por ejemplo en el caso del inicio:

```
<body class="inicio">
```

SECCION: SOBRE CROQUIER.

En esta sección quise contar un poco acerca de la historia de croquier en sí y del artista, entonces por un lado mantuve el header con el H1 y el nav y por otro lado agregue una section que contiene un h2, varios párrafos y las imágenes.

Decidí dividir el texto en diferentes párrafos para poder generar los saltos de pagina y que no quede todo el parrafo junto sin espacio.

En el CSS, la estilización del mismo esta identificado con la clase .contenido y .sobre_croquier. Esta ultima es la clase que decidí ponerle al body para poder identificarlo.

SECCION: CATALOGO.

En esta sección quise mostrar las diferentes ilustraciones que hace Croquier, entonces decidí crear una section con clase galería la cual va a tener como estilización el display:grid, lo cual genera que se arme esa grilla para poder mostrar todas las imágenes. Las imágenes del grid se agregan desde el HTML, ya que son imágenes del contenido de la pagina.

Las imágenes en el grid tienen un filtro de greyscale para que al pasar el mouse, con un hover, pueda cambiar el filtro y mostrar el color del mismo. De la misma manera realice un hover para que al pasar el cursor por una imagen se agrande la misma.

```
.catalogo .galeria img{
  filter: grayscale(80%);
  transition: filter 0.9s;
}
.catalogo .galeria a:hover img{
  filter: grayscale(0);
}
.catalogo .galeria img{
  transform: scale(1);
  transition: transform 0.9s;
}
.catalogo .galeria a:hover img{
  transform: scale(1.1);
}
```

Al hacer click en la imágenes se puede ver que aparece la modal, la misma muestra en tamaño grande la imagen pequeña del grid. Y desde ahí, desde la modal, se puede pasar para la derecha y para la izquierda e ir observando las diferentes imágenes que contiene la galería.

Una vez que se clickea fuera de la modal, la misma desaparece y volvemos a observar la galería.

El trabajo de la modal lo realice con html, css y JS. En este ultimo lo que hice fue que al hacer click en una imagen, a la modal se le agregue la clase visible y que de esa manera se incluyan todos los puntos que tenía en el CSS esa clase. La modal en todo momento esta con display:none, es decir que no se ve en ningún momento a no ser que la llamemos.

```
miniaturas.forEach(function(miniatura, indice){
  rutasImagenes.push(miniatura.getAttribute("href"));
  miniatura.addEventListener("click", function(evento){
    evento.preventDefault();
    imagenActual = indice
    imgModal.setAttribute("src", rutasImagenes[imagenActual]);
    modal.classList.add("visible");
  })
})
```

En principio le estoy diciendo que por cada imagen pequeña, primero guarde en la memoria de la variable rutasImagenes las rutas de las imágenes para un acceso mas optimo. luego le digo que al hacer click en una miniatura, me devuelva un callback, en el que primero prevenga el evento por default y luego le voy a decir que la imagen que se vea sea igual al indice, osea a la miniatura que toca en ese momento. También le voy a decir que modifique el src por la ruta que toca en cada bucle. y luego que le ponga la clase "visible" para que se pueda ver finalmente la modal

Luego para que al apretar la modal, la misma desaparezca, hice este bloque de código

```
if(modal){  
    modal.addEventListener("click", function(){  
        modal.classList.remove("visible");  
    });  
};
```

En la modal, al pasar para la derecha y para la izquierda, se puede observar que no se termina nunca, no se frena una vez que llega a la primera o ultima imagen, esto lo trabajo en JS de la siguiente manera

```
flechas.forEach(function(flecha, indice){  
    flecha.addEventListener("click", function(evento){  
        evento.preventDefault();  
        evento.stopPropagation();  
  
        if(indice == 0){  
            imagenActual = imagenActual > 0 ? imagenActual - 1 : rutasImagenes.length - 1;  
        }else{  
            imagenActual = imagenActual < rutasImagenes.length - 1 ? imagenActual + 1 : 0;  
        }  
        imgModal.setAttribute("src", rutasImagenes[imagenActual]);  
    });  
});
```

En el fichero de JS se puede ver el comentario que hace referencia a todo el proceso que se realizó para la modal.

SECCION: CONTACTO.

En esta ultima sección quise simular un formulario de contacto, para que el usuario pueda hacer una consulta o mismo contactarse con el artista. El mismo es un formulario plano, no llega a ningún lugar al clicar "enviar". El formulario cuenta con 4 inputs de tipo texto, que tienen su placeholder para ser de ayuda al usuario y un quinto de tipo submit.

RESPONSIVE

La pagina web de Croquier es completamente responsive, decidí generar tantos @media screen como la pagina y como cada sección me pedía.

Algo para destacar del método Responsive, es que a las pantallas de hasta 700 px se genero un nav diferente al que se ve en las pantallas grandes.

Se puede observar que arriba a la derecha hay un menú hamburguesa, donde al hacer click aparece el menú con una transition y desde ahí se puede navegar a las diferentes secciones.

Para poder hacer esto, utilicé html y css, enconjunto con JS.

```
const navegacion = document.querySelector("nav");
const botones = document.querySelectorAll(".abrir,.cerrar");

function toggleNavegacion(evento){
  evento.preventDefault();
  navegacion.classList.toggle("desplegado");
};

botones.forEach(function(boton){
  boton.addEventListener("click", toggleNavegacion);
});
```

Donde genero una función para que se agregue o se elimine, según corresponda, la clase ".desplegado". Entonces le digo que al apretar el botón de abrir (menú hamburguesa) o cerrar (x dentro del menú desplegado) se genere esa función que cree.

Entrega: 22/07/22

Florencia Mejliker

flormejliker@alumnos.cei.com