

Añadir portada institucional

Práctica de ejemplo: Google Clone

Referencias útiles:

HTML Tutorial de W3Schools

<https://www.w3schools.com/html/default.asp>

Documentación de Mozilla para desarrolladores

<https://developer.mozilla.org/es/docs/Web/HTML>

Documentación de Chrome DevTools

<https://developer.chrome.com/docs/devtools/>

Guía visual para CSS

<https://cssreference.io/>

Trucos para CSS

<https://css-tricks.com/>

Flexbox Froggy – Para aprender flex jugando

<https://flexboxfroggy.com/#es>

Grid Garden – Para aprender Grid jugando

<https://cssgridgarden.com/#es>

Metodología BEM que permite, entre otras cosas, colocar nombres más claros a clases y atributos

<https://getbem.com/introduction/>

Íconos

<https://iconos8.es/>

Íconos de google

<https://fonts.google.com/icons>

Alternativa para íconos

<https://www.flaticon.es/>

Alternativa para íconos

<https://fontawesome.com/icons?d=gallery>

Conceptos previos:

- HTML Semántico: Son etiquetas que ayudan a identificar en dónde estamos trabajando. Para maquetar nuestras páginas se pueden usar puros “div”; sin embargo, para que nuestra página sea mejor indexada por el navegador, se deben utilizar las etiquetas que ya tiene una función específica como “header”, “footer”, etc.
- HTML tiene dos tipos de etiquetas:
Contenedoras: Contiene otras etiquetas y ocupan un espacio. Ejemplos: header, nav, section, div.
De contenido: Contiene lo que el usuario va a ver (texto, imágenes, videos, etc.). Ejemplos: p, a, li, h1, img.
- En CSS existen tres tipos de layout: -Display Layout (el primero que salió)
-Display Grid -Display Flex (ambos salieron para solucionar el problema de ajuste de elementos.

Primero, revisar la maqueta de la página:

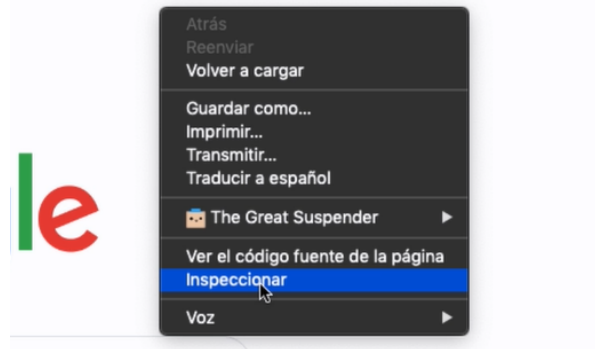
1. Observar el header
2. Se encuentra la sección principal con 3 elementos:
 1. Ícono
 2. Un input (un elemento en donde vamos a ingresar una entrada)
 3. Una serie de botones
3. Footer - Cuenta con algunos links que llevan a diferentes partes. Tiene dos secciones

Setup del proyecto en VS Code

1. Crear la carpeta en tu explorador de archivos con el nombre “P1 - Google Clone”
2. Abrir VS Code
3. Arrastrar la carpeta a VS Code
4. Crear el archivo “index.html”

5. Crear una carpeta llamada “css” y ahí crear el archivo “main.css”

Extra antes de empezar: Chrome Dev tools (en casi todos los navegadores existe). Sirve para muchas cosas, en este caso se usará para manipular el css en tiempo real. Para llegar al Dev Tools, en cualquier parte de una página se da click izquierdo y luego se selecciona “inspeccionar”. En esta práctica solamente usaremos la pestaña “Elements”.



Iniciando la práctica:

1. **Crear la estructura básica** de etiquetas en html. Tip: en VS Code existe un shortcut que crea automáticamente la estructura al presionar “!”. En las primeras veces es importante reconocer cómo hacer comentarios y los cierres de etiquetas. *Investigar qué es “DOCTYPE” y qué significan todos los “metas” que están en la etiqueta “body”:*

```
<!DOCTYPE html> <!--Esto le dice al navegador que estamos hablando de HTML5 -->
<html lang="en"> <!--Es el contenedor html -->
<head> <!--Estructura no visible en el proyecto pero que se necesita en el navegador para
manejar dependencias y ciertos metas importantes -->
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body> <!-- Aquí irá todo el contenido que el usuario podrá ver-->

</body>
</html>
```

2. **Enlazar con el archivo css.** Con el fin de que el navegador pueda reconocer los estilos que estaremos trabajando, es importante que el archivo html esté

ligado con el archivo de estilos css. Ligar nuestro archivo css al archivo html agregando la siguiente línea en la etiqueta “head”, debajo de la etiqueta “title”:

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

3. Agregar la etiqueta de html semántico “header”. Asimismo, añadir la etiqueta “nav” que es una sección de navegación que contiene elementos que llevan a otras páginas que utilizarán los elementos “ul” (unordered list) con sus respectivos “li” (list item). Tip: presiona “li*4” para que en VS Code se generen 4 elementos de la etiqueta “li”.

```
<body> <!-- Aquí irá todo el contenido que el usuario podrá ver-->
  <header>
    <nav>
      <ul>
        <li></li>
        <li></li>
        <li></li>
        <li></li>
      </ul>
    </nav>
  </header>
</body>
```

4. Dentro de los elementos “li”, se utilizará la etiqueta “a” (anchor) para crear un enlace a otras páginas de internet, archivos o ubicaciones dentro de la misma página, direcciones de correo, o cualquier otra URL. Esta etiqueta a menudo lleva “href” en donde va una URL o un fragmento de URL al cual apunta el enlace. El elemento de “Photo” se va a actualizar en un futuro ya que se tenga la foto del usuario, este ícono saldrá de la página real de google.

```
<li>
  <a href="">Gmail</a>
</li>
<li>
  <a href="">Imágenes</a>
</li>
<li>
  <a href="">Icon</a>
</li>
```

```
<li>
  <a href="">Photo</a>
</li>
```

5. Para ver estos elementos en el navegador “en tiempo real”, las versiones más actualizadas de VS Code cuentan con una opción llamada “Open with Live Server”. Al presionar click derecho en cualquier parte del código en VS Code, aparece generalmente como las últimas opciones. Esta opción simula un servidor local de tal manera que por cada cambio que le hagamos a nuestros archivos, en el navegador no hay necesidad de refrescar la página para observar dichos cambios. Revisar el [Tip 1](#).

Aplicando estilos

6. *Repasar la importancia del uso de HTML semántico para un uso más sencillo para CSS. Repasar los conceptos de los dos tipos de etiquetas y recalcar su importancia para, nuevamente, tener un uso más sencillo para CSS.*

Estilos al body. En el archivo main.css se añade estilo a la etiqueta “body”. Cuando se inician los navegadores por defecto dan un margin/padding que podríamos no necesitar, por lo que se requiere quitárselos. Cambiamos el tipo de fuente y su tamaño. Si se requiriera quitar el margin/padding de toda la página, revisar [Tip 2](#).

```
body {
  margin: 0; /*Que no haya margin y padding que da el navegador default para que
nosotros lo manipulemos más adelante*/
  padding: 0;
  font-size: 13 px;
  font-family: Arial, Helvetica, sans-serif;
}
```

7. **Estilos al header.** Agregarle estilo al header para darle una altura específica y que ocupe el ancho de toda la pantalla. Px significa pixeles:

```
header {
  width: 100%; /*Para que el header ocupe todo el ancho de la pantalla*/
  height: 60px;
}
```

8. **Estilos al nav.** Para darle estilo al nav, el html nos ayuda ya que podemos ser muy específicos para evitar problemas de estilos entre diferentes elementos de HTML ya que en una página web extensa pueden existir repetidos elementos múltiples veces. Se utilizará el display flex que ayuda a ajustar el contenido de tal manera que

se pueda justificar/alinear a la derecha. Practicar con Flexbox Froggy para tener una idea más clara de cómo funciona flex.

```
/*Busca una etiqueta nav que esté dentro de la etiqueta header*/
header nav {
  display: flex;
  justify-content: flex-end;
}
```

9. **Editar una lista en HTML.** Modificar el 100% de la etiqueta de lista no ordenada (ul). Para ello, se necesita agregar un identificador a esta etiqueta. Se recomienda leer sobre la **metodología BEM**, que es una arquitectura que permite, entre otras cosas, colocar nombres más claros a las clases y atributos. **Importante:** las clases son genéricas, mientras que los id, únicos. Colocamos clases a elementos que se repetirán a lo largo del código, es decir, tendrán la misma clase. Los id no se pueden repetir.

```
<nav>
  <ul class="nav-right-section">
    <li>
```

10. **Editar una lista en HTML.** Usar la metodología BEM para ser específicos al llamar a las clases y asegurar que no haya otra estructura parecida. Se invoca la clase como: la etiqueta header que contenga un nav con una clase llamada nav-right-section (la clase se invoca con un punto).
- width: tiene un ancho específico para que no tenga el 100% como el header
 - height: como ya se definió una altura a su padre (header) y no puede ser mayor, se coloca automática, para que tome la altura de su padre
 - display: para posicionar los elementos
 - list-style: se quitan los bullets, ya que no se necesita
 - justify-content: se centra el contenido para que no se quede únicamente a la derecha
 - align-items: se alinean los elementos para no mantenerlos en la parte de arriba

```
header nav .nav-right-section{
  width: 250px;
  height: auto;
  display: flex;
  list-style: none; /* Para quitar los puntos de la lista */
  justify-content: center;
  align-items: center;
}
```

11. Margin. Ahora se tienen que separar los elementos con el margin. Se especifica el elemento en cuestión. La etiqueta nav que tenga una clase llamada nav-right-section y que contenga una etiqueta a.

margin-right: se despega un poco más del extremo de la pantalla

color: se usa una cifra hexadecimal de color negro

```
nav .nav-right-section a {  
  margin-right: 10px;  
  color: #000000;  
}
```

12. Eliminar el subrayado. Falta eliminar el subrayado que tiene el texto. Como no se requiere en ninguna etiqueta ancla, se aplican los cambios a la etiqueta de manera general en la parte de arriba. Se llama directamente.

text-decoration: para quitar el subrayado.

cursor: pointer: para agregar la imagen de una mano al posicionar el cursor sobre el texto.

Se recomienda poner este código por debajo de body para dar a entender que es un ajuste global.

```
a {  
  text-decoration: none;  
  cursor: pointer;  
}
```

13. Estilos al ícono de navegación. Aplicar una clase a la etiqueta en que se encuentra el ícono y eliminar el texto para que no entre en conflicto con el ícono.

```
<li class="menu-icon">  
  <a href=""></a>  
</li>
```

14. Aplicar los estilos. En el archivo css se llama específicamente a nuestra clase.

Se inserta la imagen como background (fondo) con el atributo url. Es importante pegar la url entre comillas

En algunos casos, si la imagen insertada es muy pequeña en relación con el contenedor, se repetirá para rellenar el espacio. background-repeat: no-repeat lo evita

background-position: center para centrar el ícono

background-size: contain ajusta la imagen al tamaño del contenedor, evitando el punto antes mencionado por si es muy pequeña

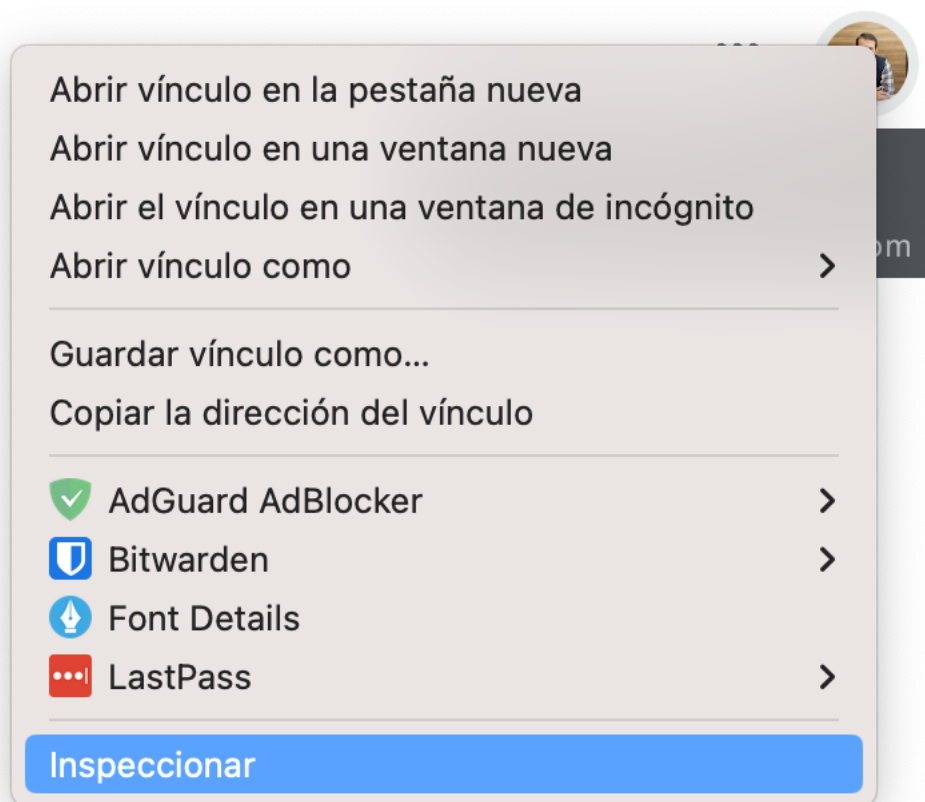
Se define su altura y ancho en 25 píxeles

Se debería tener el siguiente resultado:

Gmail Imágenes Photo

```
nav .nav-right-section .menu-icon{  
  background-image: url("https://static.thenounproject.com/png/756729-200.png");  
  background-repeat: no-repeat;  
  background-position: center;  
  background-size: contain;  
  width: 25px;  
  height: 25px;  
}
```

- 15. Estilos a la imagen de perfil.** Se debe obtener el enlace de la imagen de perfil de Google en google.com. Se da clic derecho en la foto y luego clic en “Inspeccionar”.



- 16. Pegar el enlace de la consola.** En la consola se posiciona justo en la url de la foto, se copia ese enlace y en el archivo HTML se cambia el contenido de “Photos” de la barra de navegación y se inserta una etiqueta img en su lugar para luego pegar el enlace en el atributo src. El atributo alt sirve para ponerle descripción a la imagen

con fines de accesibilidad. Probablemente la URL sea demasiado larga para que se vea completa en la pantalla de tu editor, revisa el [Tip 3](#) de Word Warp (ajuste de palabra). Si la foto es demasiado grande, en los últimos caracteres se debe cambiar 64 por 32.



```
<li>
  <a href=""></a>
</li>
```

- 17. Detalles en el estilo de la imagen de perfil.** Se llama a la etiqueta `img` dentro de la clase `.nav-right-section` de la etiqueta `nav`. Se aplica un `border-radius` del 50% para que sea completamente redondo. También se aplica un `margin-left` de 10 píxeles para que no esté tan pegada al ícono de menú.

```
nav .nav-right-section img{
  border-radius: 50%;
  margin-left: 10px;
}
```

El header debe tener un estilo similar a este:



- 18. Creando la estructura.** En la etiqueta `main` se coloca todo el contenido que es principal en el proyecto. Se puede considerar como todo lo que esté por debajo del

header y antes del footer y es esencial para mejorar nuestro posicionamiento considerando el SEO. En esencia, es el cuerpo de la página. Se generan tres secciones: una para el logo, otra para la barra de búsqueda y el último para los botones de “buscar” y “me siento con suerte”. Como tenemos varias secciones, es importante que se distingan con clases.



- 19. Sección 1.** En la primera sección se añade una clase, en este caso *main-logo*. Se agrega una etiqueta *img* para el logo.
- Sección 2.** En la segunda sección se añade una clase. En este caso *main-input* (por ser el contenedor principal)
- Se distingue que hay tres elementos: un ícono de lupa, un input para ingresar texto, y un botón para comando por voz
- Se crea una etiqueta *div* cuya clase se llamará *main-input-container* (porque es el contenedor que lleva el input)
- Se crea un *span* para colocar un ícono y le añadimos una clase. En este caso *search-icon* (por la lupa). *Span* es similar al *div* con la diferencia que se enfoca en una sola línea para luego ponerle un estilo, en cambio *div* se enfoca en todo un bloque.
- Se crea un *input*
- Se crea una etiqueta *a* para ligar el ícono de micrófono a una acción y se le añade una clase. En este caso *micro-icon*
- Sección 3.** En la tercera sección se añade una clase. En este caso *main-buttons*
- Se necesitan dos contenedores, por lo que se crean dos *div*
- Se añade una etiqueta *button* en cada una y se escribe el texto que lleva cada una

```
<!-- Considerando el SEO, se usa main para lo que va después del header y antes de footer -->
<main>
  <!-- Sección del logo -->
  <section class="main-logo">
    <img src="" alt="">
  </section>
  <!-- Sección del buscador -->
  <section class="main-input">
    <div class="main-input-container">
```

```

        <!-- Logo de lupa -->
        <span class="search-icon">

        </span>
        <!-- Es donde el usuario escribirá lo que desea buscar -->
        <input type="text">
        <!-- En el micrófono se usa anchor para simular el comportamiento de un
enlace -->
        <a class="micro-icon" href=""></a>
    </div>
</section>
<!-- Sección de botones -->
<section class="main-buttons">
    <div><button>Buscar con Google</button></div>
    <div><button>Voy a tener suerte</button></div>
</section>
</main>

```

20. Para darle estilo al main. Luego de llamar a la etiqueta main, lo primero es separarla un poco de header: se añade un margin-top de 150 píxeles y se centra el contenido con un text-align: center.

```

main {
    margin-top: 150px;
    text-align: center;
}

```

21. Agregando la imagen. En src se agrega el siguiente enlace:

```

<!-- Sección del logo -->
<section class="main-logo">
    
</section>

```

22. Para ponerle estilo a la sección “Logo de Google”. Se llama a la clase desde la etiqueta padre para ser específicos: main .main-logo.
Se define un width de 530 píxeles, que serán las dimensiones que tendrá el logo.
Se define un margin: 0 auto. Cuando agregamos un ancho a una etiqueta, pero aún hay espacio sobrante y queremos que se posicione de manera automática, aplicamos auto. Esto hace que divida el espacio sobrante entre dos y coloca a los lados. Así nos aseguramos que se mantenga siempre en el centro
Se define un margin-bottom de 35 píxeles

```

main .main-logo{

```

```
width: 530px;
margin: 0 auto;
margin-bottom: 35px;
}
```

23. Dimensiones exactas. La imagen que se añadió ya tiene las dimensiones necesarias; sin embargo, la manera correcta de que se pueda ajustar el tamaño desde el estilo sería así:

```
/* La imagen ya tiene las dimensiones necesarias, pero si no fuera así, este estilo le da
el tamaño adecuado */
main .main-logo img{
  height: 92px;
  width: 272px;
}
```

24. Darle estilo al *main-input*. Se llama a la clase main. *main-input*. Se ajusta el width a 530 píxeles. Se coloca el margin: 0 auto para que se posicione en la mitad. margin-bottom: 35px para que la sección del input empuje la sección de botones a la parte de abajo.

```
main .main-input {
  width: 530px;
  margin: 0 auto;
  margin-bottom: 35px;
}
```

25. Darle estilo al *main-input-container*. Se llama la clase que contiene a los íconos *main .main-input-container*. Hay que recordar que se creó un contenedor dentro de un contenedor para manejar correctamente sus estilos. Se ajusta un **width: 525px** para que no tenga el mismo tamaño que su padre. Se coloca un **border-radius: 100px** para redondear los extremos. Se genera un borde de un pixel con el color del borde del input original de Google con **border: 1px solid #dfe1e5**. ([Utilizar el tip para el color picker](#)). Se coloca **display: flex** para que los elementos se posicionen de manera lineal. Se coloca **justify-content: center** para situarlos en el centro. Se alinean los elementos al centro con **align-items: center**.

```
main .main-input-container {
  width: 525px;
  border-radius: 100px;
  border: 1px solid #dfe1e5;
  display: flex;
  justify-content: center;
```

```
align-items: center;
}
```

26. Redimensionando el input. Se llama una clase ***main-input*** que llegue a la etiqueta input con ***main .main-input input***. Se ajusta el ***width: 450px*** y el ***height: 40px*** para que sea mas pequeño que su contendor padre. Se quita el borde permanente con ***boder: none*** y el que se genera al hacer clic sobre el input con ***outline: none***. [Utilizar el tip para conocer los diversos tipos de bordes.](#)

```
main .main-input input {
  width: 450px;
  height: 40px;
  border: none;
  outline: none;
}
```

27. Añadir el efecto hover. Se llama la etiqueta contenedora del input con el pseudo-elemento hover. Es decir, los diseños con hover que se apliquen solo se mostrarán cuando se coloque el cursor encima del elemento. Con ***box-shadow: 0 1px 6px 0 #20212447*** se agrega una sombra paralela con un color en tono oscuro. Con ***border-color: #dfele500*** se añade un borde más oscuro, es decir, resalta más como el input original.

```
main .main-input .search-icon {
  background-image:
url('https://upload.wikimedia.org/wikipedia/commons/thumb/7/7e/Vector_search_icon.svg/945px-Vector_search_icon.svg.png');
  background-repeat: no-repeat;
  background-position: center;
  background-size: contain;
  width: 18px;
  height: 18px;
}
```

28. Agregar ícono de lupa. Primero, se llama la clase ***.search-icon*** desde la clase ***.main input***. Se añade un ***background-image*** e insertamos la url del ícono entre comillas. Enlace:
"https://upload.wikimedia.org/wikipedia/commons/thumb/7/7e/Vector_search_icon.svg/945px-Vector_search_icon.svg.png".
background-repeat: no-repeat para evitar que el navegador repita la imagen por rellenar espacio.

https://upload.wikimedia.org/wikipedia/commons/thumb/e/e8/Google_mic.svg/716px-Google_mic.svg.png
background-position: center para centrarlo.
background-size: contain para que tome el tamaño del contenedor padre.
Se añade un width: 18px y un height: 18px para redimensionar el ícono.

```
main .main-input .search-icon {  
  background-image:  
url('https://upload.wikimedia.org/wikipedia/commons/thumb/7/7e/Vector_search_icon.svg/945px-Vector_search_icon.svg.png');  
  background-repeat: no-repeat;  
  background-position: center;  
  background-size: contain;  
  width: 18px;  
  height: 18px;  
}
```

29. Para agregar el ícono de micrófono. Se llama la clase **.micro-icon** desde la clase .main input. Se copia el código de arriba y se cambia la url del ícono de micrófono. Enlace:

["https://upload.wikimedia.org/wikipedia/commons/thumb/e/e8/Google_mic.svg/716px-Google_mic.svg.png"](https://upload.wikimedia.org/wikipedia/commons/thumb/e/e8/Google_mic.svg/716px-Google_mic.svg.png).

Se añade el **cursor: pointer** para que el cursor adopte la forma de la manita al colocarlo sobre el ícono.

```
main .main-input .micro-icon {  
  background-image:  
url('https://upload.wikimedia.org/wikipedia/commons/thumb/e/e8/Google_mic.svg/716px-Google_mic.svg.png');  
  background-repeat: no-repeat;  
  background-position: center;  
  background-size: contain;  
  width: 18px;  
  height: 18px;  
  cursor: pointer;  
}
```

30. **Aplicar los estilos de los botones.** Se llama la clase que contiene los botones dentro de la etiqueta main. Se ajusta el width: 530px. Se ajusta el margin: 0 auto para que se posicione siempre en el centro.

```
main .main-buttons {  
  width: 530px;  
  margin: 0 auto;
```

```
}
```

31. Definir la posición de los botones. Se llama la clase que contiene los botones que contenga una etiqueta div con main .main-buttons div. Se ajusta el display en inline-block. Esto es porque por defecto el navegador le asigna a las cajas display: block, lo que hace que esté una encima de otra. Al usar inline-block las ponemos una a lado de otra. Por esto se le asigna la misma propiedad a las dos cajas.

```
main .main-buttons div {  
  display: inline-block;  
}
```

32. Darle diseño de caja a los botones. Se llama la clase que contiene los botones que contenga la etiqueta button con main .main-buttons button. Se le asigna una altura con height: 36px. Se ajusta el color de fondo con background-color: #f2f2f2. Se cambia el borde para que no se desplace al colocar el cursor encima con border: 1px solid #f2f2f2. Se cambia el tamaño de fuente con font-size: 14px. Cambiamos el color de la fuente con color: #5f6368. Se redondean los bordes con border-radius: 5px. Se añade el espacio interno a los lados con padding: 0 15px. Se añade una separación entre los botones con margin-right: 15px.

```
main .main-buttons button {  
  height: 36px;  
  background-color: #f2f2f2;  
  border: 0;  
  font-size: 14px;  
  color: #5f6368;  
  border-radius: 5px;  
  padding: 0 15px;  
  margin-right: 15px;  
}
```

33. Poner el efecto hover en los botones. Se necesita que al pasar el cursor sobre los botones, cambie el color del texto y se cree una sombra alrededor de la caja. Para ello:

Se llama la clase que contiene los botones con el pseudo-elemento hover. Así: main .main-buttons button:hover.

Se genera un borde sólido con border: 1px solid #c6c6c6.

Se genera una sombra con box-shadow: 0 1px 1px rgba(0, 0, 0, 0.1).

Se cambia el color del texto con color: #222.

Se añade un color de fondo con background-color: #f8f8f8.

Se agrega cursor: pointer para que el ícono del cursor cambie a una manita al posicionarlo sobre los botones.


```
main .main-buttons button:hover {  
  border: 1px solid #c6c6c6;  
  box-shadow: 0 1px 1px #000001;  
  color: #222;  
}
```

Tips:

1. En caso de que VS Code no tenga esa opción, instalar la extensión en la pestaña homónima dentro de VS Code o en el siguiente enlace:
<https://marketplace.visualstudio.com/items?itemName=ritwickdey.LiveServer>
Otra opción sería darle doble click al archivo index.html desde nuestro explorador de archivos.
2. Si se requiere aplicar un estilo a toda la página, usar el * como selector universal.
3. Si en VS Code se desea ver texto o url demasiado largo, en la pestaña View de VS Code se activa la función Word Warp.
4. Para utilizar una herramienta que permita obtener el color hexadecimal exacto en una página web, nos vamos a la opción de “inspeccionar” del navegador. Una vez ahí, se selecciona cualquier estilo que tenga un color, hacer clic a ese color y después hacer clic al ícono de “Cuentagotas” o “Gotero” para que se active la función y con darle clic al elemento en cuestión, ya obtenemos nuestro color en hexadecimal.



Usualmente a esto se le conoce como “Color Picker” y existen diversas herramientas web/desktop que ayudan a lograr el objetivo.

5. Lista de propiedades que puedes usar para distintos estilos de bordes. La propiedad *border-style* especifica qué tipo de borde mostrar. Se permiten los siguientes valores:
dotted - Define un borde punteado
dashed - Define un borde punteado
solid - Define un borde sólido
double - Define un doble borde
groove- Define un borde acanalado en 3D. El efecto depende del valor del color del borde.

ridge- Define un borde ondulado en 3D. El efecto depende del valor del color del borde.

inset- Define un borde insertado en 3D. El efecto depende del valor del color del borde.

outset- Define un borde de inicio 3D. El efecto depende del valor del color del borde.

none - Define sin borde

hidden - Define un borde oculto

La propiedad *border-style* puede tener de uno a cuatro valores (para el borde superior, el borde derecho, el borde inferior y el borde izquierdo).