

Trabajo práctico autónomo.

Objetivo.

El objetivo del presente trabajo práctico es el armado de una página personal.

En esta primera parte se espera que el alumno pueda seguir una serie de pasos detallados en el presente requerimiento, que los guiará en el armado paso a paso de la página final.

Cada uno de los pasos incluye las aclaraciones teóricas que correspondan, por lo que, en cada uno, el alumno incorporará nuevos conocimientos teóricos de la mano de su correspondiente aplicación práctica.

Alcance.

Estas son las habilidades que se espera poner en práctica:

- Lectura e interpretación de un requerimiento técnico formal.
- Aplicación práctica de los conocimientos de HTML vistos en clase.
- Adquisición de los conocimientos prácticos del uso de Bootstrap.

Material de la práctica.

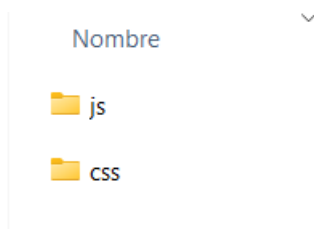
Para la realización de la presente práctica, será necesario el uso de una computadora, un navegador de internet y un editor de texto de los usados para programación.

Será necesario el uso de los archivos de Bootstrap descargados en la PC del alumno.

Paso 1: Carpeta de trabajo.

El primer paso siempre para comenzar un nuevo proyecto es armar la carpeta de trabajo con todas las herramientas que vayamos a necesitar. En nuestro caso, vamos a trabajar con Bootstrap, recordemos que en la carpeta de cada trabajo, deberemos incluir todo lo necesario para que la página web funcione correctamente, ya que en algún punto vamos a necesitar pasar todos los archivos de nuestro trabajo al equipo de nuestro cliente o deberemos subirlo a un hosting, que son repositorios para que nuestra página web este accesible a todo el mundo.

Para el presente trabajo, vamos a crear una carpeta llamada HTML y dentro de ella colocaremos las carpetas CSS y JS que previamente hemos descargado de la página de Bootstrap (<https://getbootstrap.com/>)



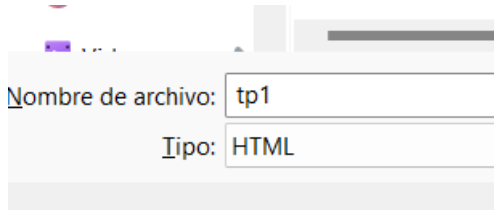
Paso 2: Página principal de nuestro proyecto.

Ya hemos visto como se crea y funciona un menú de navegación. Vimos como es la lógica, aprendimos relacionar la etiqueta `` con los diferentes archivos armando la ruta relativa de cada uno y colocándola como valor al atributo HREF.

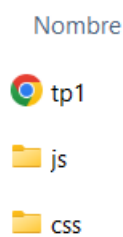
Para este trabajo en particular, vamos a crear una página web de un solo archivo, que contendrá varias secciones y el usuario podrá navegar entre cada una de ellas, usando un menú de navegación al que modificaremos para que solo se mueva dentro del archivo.

Por esto es por lo que solo vamos a crear una única página y lo vamos a hacer como vimos en clase, aplicando la estructura básica y obligatoria de toda página web.

Desde nuestro editor de texto, en mi caso Visual Studio Code, creamos un archivo nuevo y lo guardamos en nuestra carpeta de trabajo, recordando colocarle la extensión HTML.



Nuestra carpeta de trabajo debería verse así:



Lo que sigue, es comenzar a escribir en nuestro archivo, el código correspondiente, en nuestro caso comenzaremos con la estructura de la página.

```
<!doctype html>
<html lang="es">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Fundación TP1</title>

  </head>
  <body>

  </body>
</html>
```

El siguiente paso es agregar las dos etiquetas que necesitamos para que nuestra página pueda utilizar los archivos de Bootstrap que ya tenemos descargados y colocados en la carpeta de trabajo.

CSS.

Comencemos con el archivo de estilos, y para esto usaremos la etiqueta LINK que nos permite indicarle al navegador que debe dirigirse a la ruta que le indiquemos y cargar en memoria todo el código que encuentre en dicho archivo.

Es importante recordar que en general, el navegador no nos dará un mensaje de error por nuestros errores en el maquetado, por lo que deberemos prestar mucha atención a lo que vamos escribiendo para no encontrarnos mas adelante que nuestro trabajo no se ve como esperamos y tengamos que hacer una búsqueda de errores que nos podría llevar mucho tiempo.

Colocamos dentro del HEAD la siguiente etiqueta:

```
<link href="css/bootstrap.min.css" rel="stylesheet">
```

Dentro del atributo HREF ponemos la ruta relativa para que el navegador vaya desde el archivo actual (el archivo que estamos codificando) hasta el archivo de estilos, que se encuentra dentro de la carpeta CSS.

El atributo REL es para indicar la relación de este archivo con el tipo de archivos de hojas de estilo, esto ayuda al navegador a tener mas datos acerca de lo que debe hacer.

JS.

De manera similar, agregaremos la etiqueta correspondiente para que el navegador, también busque y cargue en memoria las funciones de JavaScript que necesitamos para nuestro proyecto.

Para los archivos de JS la etiqueta que usaremos será SCRIPT, que le indica al navegador que el contenido del archivo a cargar es código.

Colocamos esto en nuestro trabajo:

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

Al atributo SRC (por SOURCE, fuente u origen en inglés) le colocamos la ruta relativa para que el navegador encuentre, desde el archivo actual, el archivo de JS, siguiendo el mismo criterio que aprendimos en clase para el armado de rutas relativas, le indicamos que, desde nuestra posición, busque dentro de la carpeta JS al archivo que necesitamos.

Una buena práctica es colocar dentro de nuestro trabajo, todos los comentarios que nos sean de utilidad.

Así debería quedar nuestro código hasta el momento:

```
<!doctype html>
<html lang="es">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Fundación TP1</title>
    <!-- Archivo de estilos -->
    <link href="css/bootstrap.min.css" rel="stylesheet">
  </head>
</head>
<body>
  <!-- Funciones de JS -->
  <script src="js/bootstrap.bundle.min.js"></script>
</body>
</html>
```

De manera opcional para cada uno, podemos verificar muy fácilmente si hasta acá no hemos cometido ningún error, colocando algún elemento de Bootstrap en nuestra página y viendo desde el navegador si se ve correctamente.

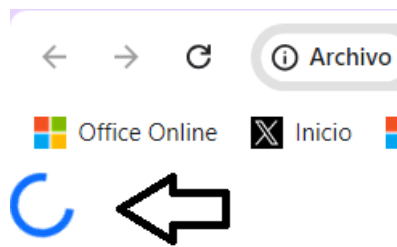
Vamos a colocar el siguiente elemento dentro del BODY de nuestra web:

```
<div class="spinner-border text-primary" role="status">
  <span class="visually-hidden">Hola mundo...</span>
</div>
```

Si todo está correcto, deberemos ver desde nuestro navegador una ruedita azul girando en la esquina superior de nuestra pantalla.

Para poder ver nuestro trabajo desde el navegador simplemente hacemos doble CLICK sobre nuestro archivo de trabajo, y se abrirá automáticamente el navegador que tengamos como predeterminado en nuestra PC.

Así deberíamos verlo:



Si está todo bien, procedemos a eliminar el último código que agregamos (el del elemento giratorio de Bootstrap que usamos para verificar).

Paso 3: Menú de navegación.

Vamos a armar nuestro menú de navegación, armado para que se mueva dentro de las secciones de nuestra página.

Para ir dándole forma a nuestro proyecto, vamos a optar por diseñar una página personal, que nos sirva de plantilla para futuros desarrollos o para una página personal más completa.

Para comenzar, vamos a definir cuales serán las secciones de nuestra página personal y diseñaremos el menú en base a eso.

Las secciones que formarán nuestra página personal serán estas:

- Inicio.
- Trabajos.
- Acerca de.
- Contacto.

Con las secciones ya decididas, comenzamos a maquetar nuestro menú de navegación utilizando la etiqueta NAV como vimos en clase y algunas clases de estilo de Bootstrap.

Para que nuestra página completa quede debidamente centrada, comenzamos por agregar al BODY, la clase CONTAINER de la siguiente manera:

```
<body class="container">
```

A continuación, comenzamos el maquetado del menú.
Lo colocaremos dentro del BODY, a continuación de la etiqueta SCRIPT que utilizamos para cargar el archivo de JS.
Escribimos lo siguiente:

```
<!-- Menú de navegación -->
<nav class="navbar navbar-expand-lg bg-secondary text-uppercase fixed-top">
  <div class="container">

    </div>
  </nav>
```

A la etiqueta NAV que conocemos, le agregamos las siguientes clases de estilo de Bootstrap:

navbar:

Esta clase se utiliza para definir una barra de navegación. Es el contenedor principal que agrupa todos los elementos de la barra.

navbar-expand-lg:

Esta clase hace que la barra de navegación sea responsive, es decir, que se adapte a diferentes tamaños de pantalla. En este caso, la barra se expandirá completamente en pantallas grandes (lg) y se colapsará en pantallas más pequeñas.

bg-secondary:

Aplica un fondo de color secundario a la barra de navegación. Bootstrap tiene una paleta de colores predefinida, y “secondary” es uno de esos colores, una especie de gris.

text-uppercase:

Convierte todo el texto dentro de la barra de navegación a mayúsculas.

fixed-top:

Fija la barra de navegación en la parte superior de la página, de modo que permanezca visible mientras se desplaza el contenido, esto resulta práctico para los usuarios ya que, si llegan hasta el fondo de nuestra página, no tienen que volver hasta el inicio para usar el menú de navegación.

Luego, dentro de la etiqueta NAV colocaremos un contenedor DIV para acomodar todo el contenido y que se vea lo más ordenado posible.

Para movernos dentro de nuestra página, tenemos que hacer lo siguiente.

- 1- A cada sección de nuestra web deberemos asignarle un nombre, mediante el atributo ID.
- 2- Dentro del menú, a cada enlace de la etiqueta `` en lugar de colocarle un nombre de archivo, le colocaremos el valor de cada ID de cada sección, anteponiendo un símbolo #.

Agreguemos la primer etiqueta a nuestro menú, será la que nos permita volver desde cualquier lugar, hasta el inicio:

```
<!-- Menú de navegación -->
<nav class="navbar navbar-expand-lg bg-secondary text-uppercase fixed-top">
  <div class="container">
    <a class="navbar-brand" href="#inicio">Inicio</a>
  </div>
</nav>
```

Como podemos ver, la etiqueta `<a>` que agregamos dentro del DIV, nos dirigirá, cuando pulsemos sobre ella, a la sección llamada “inicio”.

¿Dónde está la sección inicio?

Todavía no la hemos definido, y como queremos que sea el principio de nuestra página web, daremos este nombre al BODY, así, cada vez que alguien presione la opción INICIO del menú, el navegador le mostrará el comienzo de nuestra página.

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

Nuestro menú será RESPONSIVE, es decir que se adaptará a diferentes tamaños de pantalla.

Cuando la pantalla sea pequeña, todas las opciones deberán contraerse y colocarse dentro de un único acceso, para que sea perfectamente visible, por ejemplo, desde un teléfono móvil.

Ese único acceso lo vamos a maquetar con forma de botón, y le colocaremos el texto “MENU” de la siguiente manera.

Escribimos lo que sigue debajo de la etiqueta `<a>` del menú:

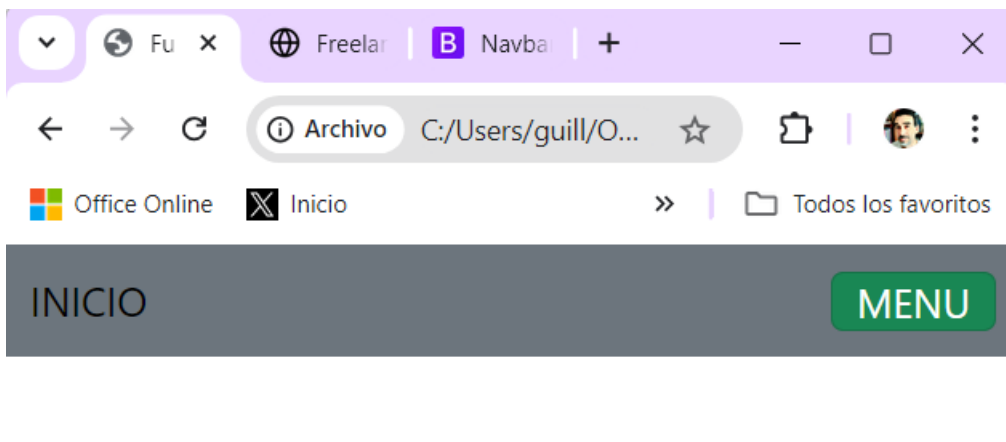
```
<button class="navbar-toggler text-uppercase bg-success text-white rounded" type="button" data-bs-toggle="collapse"
  data-bs-target="#navbarResponsive" aria-controls="navbarResponsive" aria-expanded="false"
  aria-label="Toggle navigation">
  Menu
</button>
```


Todos los atributos que le colocamos a la apertura de la etiqueta BUTTON, se pueden escribir en una misma línea, solo para hacer mas legible el apunte, lo coloqué en diferentes renglones.

Todos esos atributos de Bootstrap son necesarios para que el botón funcione, podremos analizar que hace cada uno mas adelante cuando veamos CSS.

¿Cómo podemos ver el resultado de lo que acabamos de escribir?

Guardemos los cambios, vayamos hasta el navegador y lo actualizamos para que tome los cambios, luego achicamos el tamaño de la pantalla y veremos como al hacerla mas pequeña, aparece el botón MENU:



Si volvemos a restaurar el tamaño completo de la pantalla, veremos que el botón desaparece, porque con la pantalla a su máximo tamaño, ya tenemos lugar para que todas las opciones del menú se vean bien.

Debajo del botón que agregamos recién, vamos a colocar un nuevo DIV, que es el que contendrá todas las opciones y usando algunas clases de Bootstrap, lo vamos a programar para que cuando la pantalla sea mas pequeña, todas las opciones se oculten y se ven solo cuando presionamos el botón anteriormente maquetado.

Escribimos lo siguiente:

```
<div class="collapse navbar-collapse" id="navbarResponsive">
  <ul class="navbar-nav ms-auto">
    <li class="nav-item mx-0 mx-lg-1"><a class="nav-link py-3 px-0 px-lg-3 rounded" href="#trabajos">Trabajos</a></li>
    <li class="nav-item mx-0 mx-lg-1"><a class="nav-link py-3 px-0 px-lg-3 rounded" href="#acerca_de">Acerca de</a></li>
    <li class="nav-item mx-0 mx-lg-1"><a class="nav-link py-3 px-0 px-lg-3 rounded" href="#contacto">Contacto</a></li>
  </ul>
</div>
```

El DIV que agregamos tiene las clases COLLAPSE que son las que le dan la habilidad de contraerse cuando detectan una pantalla pequeña.

Los elementos de la lista, las etiquetas LI, llevan clases que adaptan el tamaño de la letra para verse bien en diferentes tamaños de pantalla.

Por último, las etiquetas A, llevan clases que acomodan el texto de cada enlace dentro de la barra principal del menú para que se ven alineadas.

Así se ve nuestro menú normal:



Así se ve nuestro menú en una pantalla reducida:



Y si presionamos el botón MENU, veremos lo siguiente:



Tenemos varios aspectos de apariencia para modificar.

Lo habitual en este tipo de tareas es armar el maquetado, colocar cada elemento en su lugar y probar el funcionamiento, luego sobre el final cambiaremos los aspectos visuales que necesitemos, por ejemplo, los colores del menú.

Vamos a hacer algunos pequeños cambios visuales.

Modifiquemos la etiqueta NAV eliminando la clase BG-SECONDARY y la vamos a reemplazar por las clases NAVBAR-DAR y BG-DARK, quedando el código como sigue:

```
<nav class="navbar navbar-expand-lg navbar-dark bg-dark text-uppercase fixed-top">
```

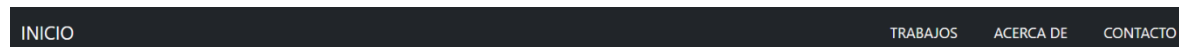
Ahora la barra del menú tiene fondo negro y letras blancas, lo que lo hace un poco más amigable de ver:



También podemos resaltar un poco más, visualmente hablando, las opciones de la barra, agregando a cada etiqueta <A>, la clase ACTIVE, quedando de la siguiente forma:

```
<li class="nav-item mx-0 mx-lg-1"><a class="nav-link active py-3 px-0 px-lg-3 rounded" href="#trabajos">Trabajos</a></li>
<li class="nav-item mx-0 mx-lg-1"><a class="nav-link active py-3 px-0 px-lg-3 rounded" href="#acerca_de">Acerca de</a></li>
<li class="nav-item mx-0 mx-lg-1"><a class="nav-link active py-3 px-0 px-lg-3 rounded" href="#contacto">Contacto</a></li>
```

Ahora las opciones de la barra se verán un poco mejor:



Paso 4: Maquetamos el Inicio

Vamos a agregar algunos elementos en la parte del inicio, para ir dándole forma, al menos a lo que queremos colocar, en este punto trabajaremos con lo aprendido mas el agregado de lo que nos gustaría colocar en nuestra página personal.

Como nuestro objetivo entre otras cosas es que los motores de búsqueda encuentren nuestra página, vamos a continuar realizando nuestro maquetado lo más formal posible.

Utilizaremos una nueva etiqueta, HEADER, pero antes veamos un poco de teoría al respecto, ¿para que sirve la etiqueta HEADER?

Se utiliza para definir un encabezado para un documento o una sección. Algunos de sus principales usos:

1. **Encabezado de página:** Puede contener el título de la página, un logotipo, un menú de navegación, y otros elementos introductorios o de navegación.
2. **Encabezado de sección:** También se puede usar dentro de una sección específica de la página, como un artículo o una sección de contenido, para proporcionar un encabezado a esa parte específica.

3. **Estructura semántica:** Ayuda a mejorar la accesibilidad y la optimización para motores de búsqueda (SEO) al proporcionar una estructura clara y semántica al contenido.
4. **Elementos comunes:** Dentro de un <header>, es común encontrar elementos como <h1>, <nav>, , y otros elementos que ayudan a identificar y navegar el contenido

Ahora que ya la conocemos, vamos a utilizar para maquetar nuestra sección principal.

Escribimos lo siguiente:

```
<!-- Sección principal -->
<header class="bg-primary text-white text-center">
  <div class="container d-flex align-items-center flex-column">

  </div>
</header>
<!-- Fin de Sección principal -->
```

Vamos a necesitar algunas imágenes, y una regla de oro para el diseño web es verificar bien que las imágenes que usemos sean de libre uso, para no tener ningún problema de mal uso de material que no es nuestro.

Existen muchas páginas que nos dejan descargar imágenes de uso libre en internet o una opción interesante es pedirle a cualquier inteligencia artificial que nos diseñe una para uso libre.

Estas imágenes las voy a dejar en la plataforma para que las puedan usar si no quieren crear las suyas.

Vamos a guardar todas nuestras imágenes dentro de una carpeta que deberemos crear en nuestro proyecto, la llamaremos IMG.



Vamos a crear tres contenedores dentro de nuestro DIV fila así:

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

Y ahora le vamos a asignar a cada uno, un número de columnas, recordando que solo tenemos 12 columnas por fila para repartir.

Podemos asignarlas de esta manera para que el contenedor del centro sea el más ancho porque será el que contenga la imagen:

```
<div class="row">  
  <div class="col-3"> </div>  
  <div class="col-6"> </div>  
  <div class="col-3"> </div>  
</div>
```

De las 12 columnas disponibles en cada fila, le asignamos 3 al primer contenedor, 6 al segundo y otras 3 al último.

Cada uno de estos 3 contenedores, estará uno junto al otro alineado dentro de nuestro contenedor fila.

Vamos a utilizar el contenedor del centro, el mas ancho, para colocar nuestra imagen, que ya la tengo descargada dentro de la carpeta IMG y le puse de nombre AVATAR.

Para esto usaremos la etiqueta IMG y agregaremos al inicio unos saltos de línea para terminar de alinearla.

Así debería quedar el código completo de nuestra sección principal hasta ahora, para poder mostrar la imagen.

Para esta sección en particular utilicé una imagen con formato SVG que se verá bien en cualquier tamaño de pantalla.

```
<!-- Sección principal -->
<header class="text-white text-center">
  <div class="container d-flex align-items-center flex-column">
    <br><br><br><br>
    <div class="row">
      <div class="col-3"> </div>
      <div class="col-6">
        
      </div>
      <div class="col-3"> </div>
    </div>
  </div>
</header>
<!-- Fin de Sección principal -->
```

Debajo de nuestra imagen, colocaremos un texto de buen tamaño para colocar nuestro nombre.

Para ello, vamos a maquetar una nueva fila, debajo de la que hicimos para la imagen y la vamos a dividir en 3 contenedores cada uno con la misma cantidad de columnas que usamos para la imagen:

```
<!-- Sección principal -->
<header class="text-white text-center">
  <div class="container d-flex align-items-center flex-column">
    <br><br><br><br>
    <div class="row">
      <div class="col-3"> </div>
      <div class="col-6">
        
      </div>
      <div class="col-3"> </div>
    </div>
    <!-- Fila para colocar nuestro nombre -->
    <div class="row">
      <div class="col-3"></div>
      <div class="col-6"></div>
      <div class="col-3"></div>
    </div>
  </div>
</header>
<!-- Fin de Sección principal -->
```

En el contenedor columna mas ancho, al igual que hicimos con la imagen, colocaremos nuestro nombre:

```
<!-- Fila para colocar nuestro nombre -->
<div class="row">
  <div class="col-3"></div>
  <div class="col-6">
    <h1 class="fst-italic text-center">Nombre y apellido.</h1>
  </div>
  <div class="col-3"></div>
</div>
```

Si probamos nuestra página, notaremos que lo que acabamos de escribir no se ve, y eso es porque inicialmente colocamos todo el texto de nuestra sección en color blanco.

Lo hice de esta manera por si queríamos mas adelante agregar un color de fondo, pero por ahora vamos a dejarlo sin color de fondo.

Para que nuestro texto sea visible, lo cambiaremos de blanco a negro, en la etiqueta HEADER.

```
<!-- Sección principal -->
<header class="text-black text-center">
```

Ahora sí, guardamos los cambios y actualizamos nuestra página para ver el estado de nuestro trabajo hasta el momento:



Nombre y apellido.

Vamos a agregar ahora, dos elementos más, una línea divisora y otro texto con nuestras habilidades o cualquier descripción que crean conveniente. Ambos elementos los vamos a colocar alineados de la misma manera que lo hicimos con la imagen y con nuestro nombre.

Es decir, volveremos a maquetar una fila que la dividiremos en tres partes y lo mismo haremos para ambos elementos, la línea y nuestra descripción.

Siempre dentro del HEADER y a continuación de la fila que maquetamos para nuestro nombre y apellido escribiremos lo siguiente:

```
<!-- Fila para colocar una barra divisora -->
<div class="row">
  <div class="col-3"></div>
  <div class="col-6">

  </div>
  <div class="col-3"></div>
</div>

<!-- Fila para colocar una descripción nuestra -->
<div class="row">
  <div class="col-3"></div>
  <div class="col-6">

  </div>
  <div class="col-3"></div>
</div>
```

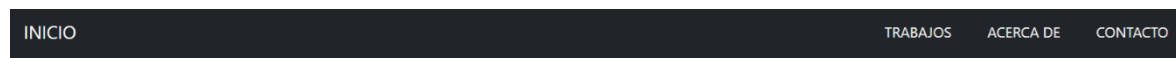
Luego, desde la página de Bootstrap, voy a elegir una barra de división con efecto de movimiento, la coloco dentro del DIV más ancho.

```
<!-- Fila para colocar una barra divisora -->
<div class="row">
  <div class="col-3"></div>
  <div class="col-6">
    <div class="progress" role="progressbar" aria-label="Animated striped example" aria-valuenow="75">
      <div class="progress-bar progress-bar-striped progress-bar-animated" style="width: 100%"></div>
    </div>
  </div>
  <div class="col-3"></div>
</div>
```

Por último, hacemos lo mismo para agregar una breve descripción, podemos utilizar el mismo tipo de texto que usamos para el nombre, pero de menor tamaño.

Hasta acá llegamos con la primer parte de nuestra página web personal. Antes de seguir avanzando y agregar las secciones que faltan, necesitamos incluir algunos conocimientos mas que veremos en clase.

Si todo salió bien, nuestra página web debería verse de la siguiente manera:



Nombre y apellido.

Desarrollador Web - Programador Full Stack.