

## HTML y CSS (Parte I)

### Introducción

El proyecto que se realizará en esta unidad es:

- Proyecto "Meet&Coffee"

En la primera unidad aprenderemos a construir un sitio web. Como ejercicio haremos un sitio llamado "**Meet & Coffee**", el cual será una página de reuniones y eventos.



Imagen 1: Proyecto Meet&Coffee

Con él aprenderemos las bases de la construcción de un sitio web, y algunas herramientas para realizar pruebas y solucionar errores comunes. Terminaremos la unidad con dos actividades para profundizar los conocimientos.

## ¿Qué es una página web? ¿Qué es un sitio web?

Antes que comencemos con todo, necesitamos definir qué es una página web y cuál es su diferencia con un sitio web.

Si buscamos en Google, posiblemente encontremos respuestas o muy complejas o muy simples. Así que, en esta instancia, una **página web** la definiremos como:

**Página web (web page):** Documento que puede ser visto a través de un navegador como Firefox, Google Chrome o Safari. Las páginas web están construidas en un lenguaje de etiquetas llamado HTML.

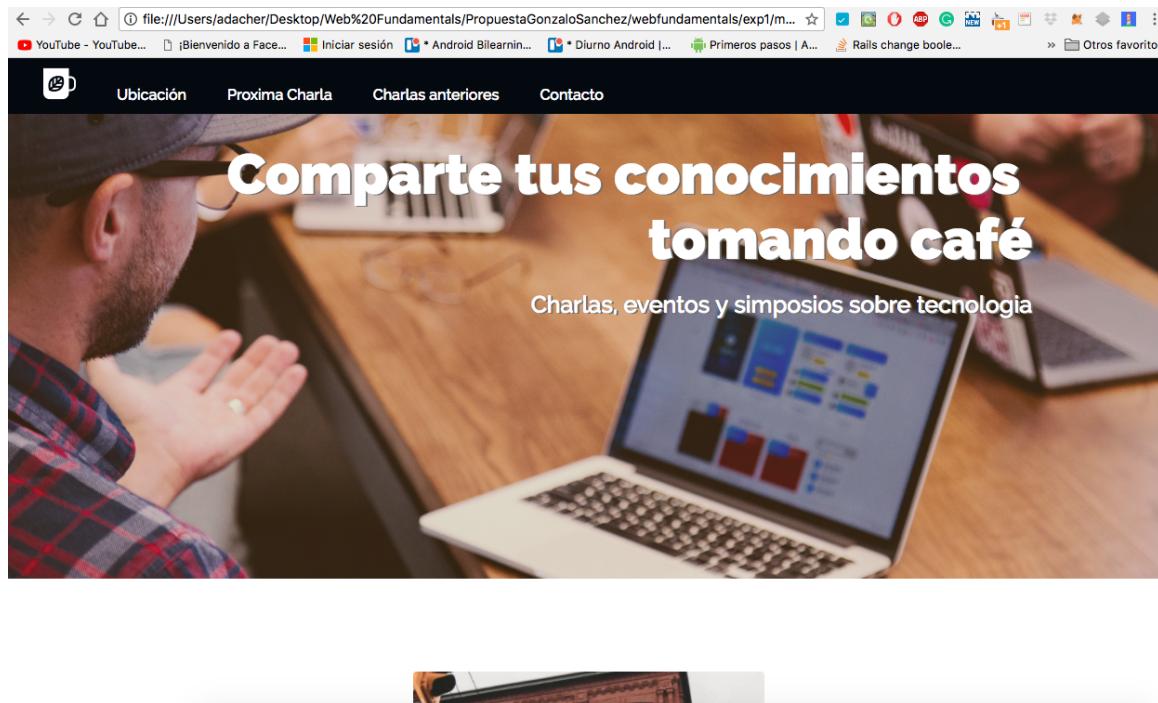


Imagen 2: Ejemplo página web

Si ya sabemos qué es una página web, podremos indicar que un **Website (sitio web)** es una colección de páginas (documentos), agrupadas e interconectadas en relación a un mismo tema.

Por ejemplo, esta página de Wikipedia contiene enlaces que la conectan a todos los diferentes artículos. Cada uno de esos artículos es una página en sí misma y todos los artículos en su conjunto son el sitio web de Wikipedia.

The screenshot shows the main page of the Spanish Wikipedia. The top navigation bar includes links for 'Portada', 'Portal de la comunidad', 'Actualidad', 'Cambios recientes', 'Páginas nuevas', 'Página aleatoria', 'Ayuda', 'Donaciones', 'Imprimir/exportar', 'Crear un libro', 'Descargar como PDF', 'Versión para imprimir', 'En otros proyectos', 'Wikimedia Commons', 'MediaWiki', 'Meta-Wiki', 'Wikispecies', 'Wikibooks', 'Wikidata', 'Wikinoticias', 'Wikiquote', 'Wikisource', 'Wikivisual', 'Wikiviajes', 'Wikcionario', 'Herramientas', 'Lo que enlaza aquí', 'Cambios en enlazadas', 'Subir archivo', 'Páginas especiales', 'Enlace permanente', and 'Informativa de la'. The main content area features a welcome message, a sidebar for participation and community, another for searches and consultations, and a central article about the Paleolithic era with two images of stone tools.

Imagen 3: Ejemplo página web - Wikipedia

Otro ejemplo, este video lo estás viendo desde una página web del sitio web de la plataforma empieza.

## Contenido y formato

En cada página web podemos identificar dos grandes áreas.

Por un lado, el contenido que ésta tiene, o sea, su texto, imágenes, videos, entre otros. Es decir, a grandes rasgos, su información.

Por otro lado, está el formato o estilo, que es el modo en el que se expresa esta información. Ya sea a través de los colores, distribución, tamaño de las imágenes, tipo de letra, entre muchas otras cosas.

En el sitio web que construiremos, el contenido será la información sobre las charlas que se realizan en un espacio colaborativo de trabajo, todo acompañado de un rico café. Todo el contenido que agregaremos al sitio estará relacionado a este tema.

Luego, le daremos estilo a la página para que se vea tal cual a estas maquetas.

Meet & Coffee

Ubicación Próxima Charla Eventos anteriores Contacto

# Descubre lo último en tecnología bebiendo café

Charlas, eventos y simposios sobre tecnología



## ¿Donde nos juntamos?

Todos los martes y viernes, de 19:00 a 22:00 en We Work, Calle Baker 133, Providencia, Santiago.



**Próxima charla**  
**Big Data para Noobs**

Rafaela Valdés  
Data scientist, PhD Theoretical Physics, Carnegie Mellon University

En esta charla revisaremos las técnicas y tecnologías que hacen que sea económico hacer frente a los datos de una escala extrema.

### Eventos anteriores



**Simposio Vegan DB**

**Machine learning 101**

**Scrum sin scream**



Meet & coffee 2018. Todos los derechos reservados.

Imagen 4: Estilo de página

El resultado de esta combinación es lo que veremos en el navegador: el contenido expresado de la forma más atractiva posible para los usuarios de nuestra página.

## Analizando la página a construir

Muchas veces, cuando nos encarguen hacer un sitio web nos pasarán algo llamados maquetas. Las maquetas son imágenes del diseño del sitio, de cómo se debería ver. En esta experiencia trabajaremos en base a estas maquetas diseñadas anteriormente.

El objetivo de construir este sitio es que nos permitirá adquirir las habilidades básicas para el desarrollo web.

La maqueta la podemos ver en: [maqueta final](#)

## Desglosando la página en secciones

Antes de construir cualquier página web es bueno que analicemos su estructura. Para eso la vamos a separar en unidades llamadas **secciones**.

En el caso de la página que construiremos veremos que existen distintas secciones de información:

- Una barra de navegación
- Una sección principal, llamada **Hero section**
- Sección de lugar
- Sección de próxima charla
- Sección de eventos anteriores
- Sección final, de contacto, a veces llamada **footer**

Esto es importante porque dentro del código tenemos que diferenciar cada sección, para trabajarlas por separado.

## Herramientas necesarias

Para desarrollar nuestra primera página web, necesitaremos que nuestro computador cuente con:

1. *Un navegador web*: Un navegador web es un programa diseñado para acceder y navegar por la web y, a través de la interpretación de los archivos, visualizar las páginas.

En esta experiencia utilizaremos Google Chrome. Lo puedes descargar desde [acá](#).

2. *Un editor de texto*: Este es un programa que se instala en el computador y que ocuparemos para escribir el lenguaje que luego interpretará el navegador. Existen muchos editores y los más utilizados hasta ahora son Atom, Visual Studio Code y Sublime. En esta serie de videos utilizaremos Atom, que puedes descargar de [acá](#).

Hemos puesto a tu disposición las imágenes, texto e información que utilizaremos en la construcción de esta página web. Puedes visualizarlos desde las siguientes URL:

[guía de estilo](#)

[maqueta final](#)

# Nuestra primera página web

En este capítulo comenzaremos a crear nuestra primera página web. Para esto es muy importante que tengas instalado los programas Chrome y Atom. También es necesario que tengas el link de la página web que construiremos. Toda esta información está en el capítulo anterior.

## Empecemos con nuestra página

Existen diversas formas de empezar a crear una página web. Para empezar un proyecto nuevo comenzaremos por una carpeta.

1. Crearemos una nueva carpeta en nuestro escritorio, que llamaremos "meet&coffee".
2. Ahora vamos abrir la carpeta con nuestro editor de texto. Hay varias formas de lograr esto:
  - La primera y más simple es arrastrar la carpeta nueva sobre el ícono de Atom.
  - Otra forma, sería arrastrar la carpeta directamente sobre una pestaña abierta de Atom.
  - También podríamos abrir Atom, ir a "File / Open" y buscar la carpeta.

Sabremos que lo logramos si podemos ver nuestra carpeta al costado izquierdo de nuestro editor. Antes de seguir, asegúrate de poder ver la carpeta del proyecto en la columna izquierda. Ahora podremos generar nuestro primer archivo en Atom.

Para ello, vamos a hacer *clic* con el botón secundario sobre la carpeta generada y seleccionaremos **new file**. También podemos presionar la tecla **a** sobre la carpeta, con esto se abrirá una ventana donde podremos escribir el nombre del archivo. Lo llamaremos **index.html**. Una vez creado el archivo, nos aparecerá al costado izquierdo dentro de la carpeta de nuestro proyecto.

La página principal de nuestro sitio se debe llamar **index.html** porque es una convención que subtiende que es el archivo índice que inicia un sitio web.

El que sea **.html**, dice que es un archivo que se interpretará como HTML, por lo tanto podrá ser leído por el navegador.

Si necesitamos cambiarle el nombre, podremos hacer click con el botón secundario (click derecho) sobre el archivo y seleccionar rename en la barra de navegación, donde podremos escribir el nuevo nombre directamente sobre el archivo para renombrarlo.

Cada vez que realicemos un cambio en el archivo, nuestro editor, Atom, nos avisará dejando un punto de color al lado del nombre del archivo, el cual será el punto de la vergüenza indicándonos que los cambios no están guardados.

Para guardar, debemos presionar **cmd + s** en Mac y **ctrl + s** en Windows o Linux. También podemos hacerlo sin utilizar atajos dentro del menú, haciendo clic sobre file y luego sobre guardar (o save).

Estaremos guardando constantemente, así que es importante recordar este atajo.

Una vez que esté guardado nuestro archivo, ya podremos abrirlo en nuestro navegador.

Si no lo guardamos, solo veremos un archivo vacío dentro del navegador. Probemos ingresando texto en nuestro archivo y guardándolo.

Hola esta es mi primera página web

Ahora probemos como se ve en en navegador, para eso abriremos el archivo con Chrome y podremos ver el texto que escribimos.

Si no guardas el archivo, al recargar el navegador no verás los cambios realizados. De igual forma para recargar la página del navegador, puedes hacer clic el ícono de recargar o utilizando el atajo `cmd + r` en Mac y `ctrl + r` en Linux o Windows.

Para ver los cambios debemos actualizar la página.

¡Felicitaciones! acabamos de crear nuestro primer documento, pero OJO, aun no estamos ocupando código HTML.

# Crear páginas con HTML

## Introducción a HTML

En este capítulo empezaremos a trabajar con HTML.

### ¿Qué es HTML?

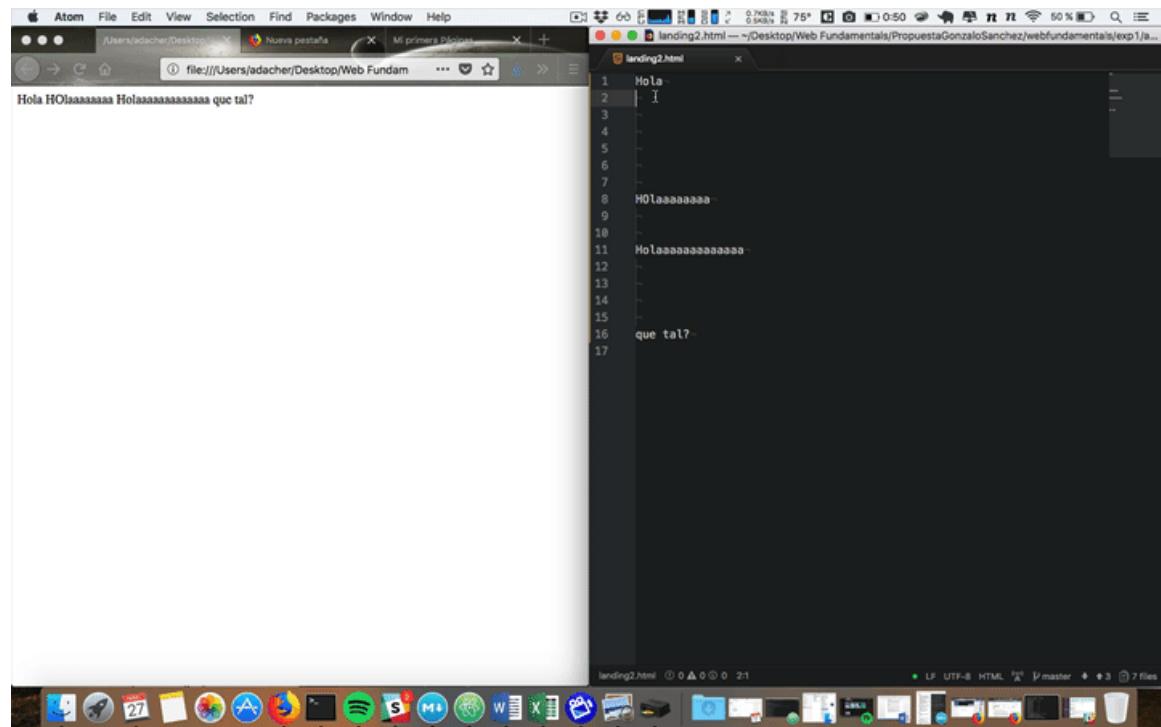
Del inglés **HyperText Markup Language**, podemos traducir HTML como un lenguaje de marcas de hipertexto, que sirve para la elaboración de páginas web.

Su principal estructura es a base de etiquetas, por lo tanto, la clave es conocer qué son, cómo se escriben y cómo se ocupan.

```
<p> Hola </p> <!-- esto es una etiqueta! -->
```

### Las etiquetas

Las etiquetas son los elementos con los que damos formato y estructura a un archivo HTML.



The screenshot shows a Mac desktop environment. On the left, a browser window displays the text "Hola HOaaaaaaaa Holaaaaaaaaaaaa que tal?". On the right, a code editor window titled "landing2.html" shows the corresponding HTML code:

```
1 Hola
2
3
4
5
6
7
8 Holaaaaaaaaa-
9
10
11 Holaaaaaaaaaaaa-
12
13
14
15
16 que tal?-
```

The code editor interface includes tabs for "landing2.html", file statistics (0 0 0 2:1), and repository information (LF, UTF-8, HTML, master, + 3, 7 files). The dock at the bottom contains icons for various Mac applications like Finder, Mail, and Safari.

Imagen 5: Etiquetas

Existen muchísimas etiquetas y cada una de ellas apuntan a diferentes tipos de elementos. Además, cada una tiene un formato determinado por defecto (que veremos más adelante).

Por lo general la estructura, o sintaxis, de una etiqueta es la siguiente:

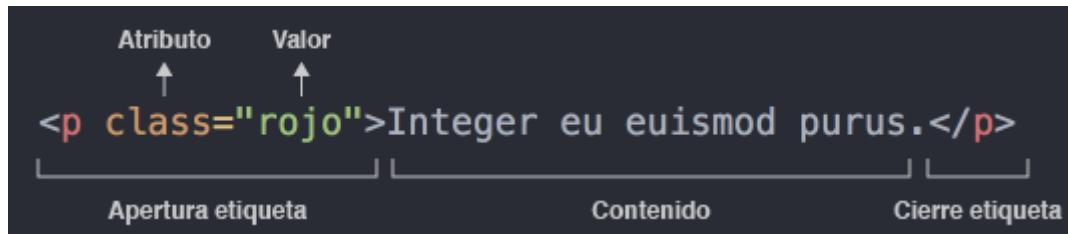


Imagen 6: Estructura etiquetas

Existen muchas etiquetas y éstas pueden estar una dentro de otras, como lo que veremos a continuación ejemplificado en la estructura base de un documento HTML.

## Estructura base de un documento HTML

La estructura básica de un archivo HTML consiste en una etiqueta cabeza ( `<head>` ) y un cuerpo ( `<body>` ). El `<head>` contiene toda la información que es para el **navegador** y el `<body>` contiene todo el contenido que es para el **usuario**.

### Ejemplo:

```
<!DOCTYPE html>
<html>
<head>
  <!-- Aquí va la información para el navegador -->
</head>
<body>
  <!-- Aquí va el contenido para el usuario -->
</body>
</html>
```

La primera etiqueta `<!DOCTYPE html>` es la que le indica al navegador cómo debe interpretar el resto del documento (el doctype especificado es de HTML5, el cuál es el estándar de hoy).

La etiqueta `<html>` especifica que todo lo que está dentro de ella deberá ser interpretado como HTML.

## Ayuda para generar la estructura

Puede resultar un poco engorroso tener que escribir una y otra vez la estructura HTML, por lo mismo, en Atom existe algo llamado el **autocomplete**. Primero debemos asegurarnos que el tipo de código que estamos escribiendo sea HTML, y eso se ve en la esquina inferior derecha del editor. Luego, simplemente tenemos que escribir `html` y apretar la tecla `Tab`.

## Material adicional

Existen diversos tipos de lenguajes de etiquetas. Algunos sirven para contenido y construcción de páginas web como Markdown. Otros son multipropósito como XML y algunos para escribir ecuaciones matemáticas como Latex.

## Trabajando en el head (título, codificación y metadatos)

Ahora que ya hemos creado la carpeta principal de nuestro proyecto, con el archivo index dentro de ella y además hemos escrito la estructura principal de archivo HTML, llegó la hora de agregar el contenido. Para ello debemos abrir la [guía de estilo](#) y la [maqueta final](#), ya que en ellas se detallan las imágenes, colores y contenido que utilizaremos para la construcción de nuestro sitio y cómo debería verse al final.

Pero vamos por partes. Como vimos anteriormente, la etiqueta `<head>` especifica el contenido que se le entregará al navegador y es necesaria para mostrar correctamente la página. Dentro de ella están contenidos los metadatos para la página. Por lo general, podemos encontrar: el título (`<title>`), la codificación (`<meta charset="utf-8">`), el favicon (`<link rel="shortcut icon">`) y otros metadatos (`<meta>`).

También podemos encontrar los estilos (`<style>` o `<link rel="stylesheet">`) y los scripts (`<script>`), pero esos temas los veremos más adelante.

### Título:

La etiqueta `title` muestra el título de la página en el tab, además tiene atribuciones semánticas para el navegador. Por lo tanto, como nuestro sitio se llama "Meet & Coffee", lo escribiremos así

```
<title>Meet & Coffee</title>
```



Imagen 7: Título

### Codificación:

La etiqueta `<meta>` con el atributo `charset` especifica la codificación de los caracteres del documento. En este caso usaremos **utf-8** para poder codificar símbolos y caracteres latinos. `<meta charset="utf-8">`

Esta etiqueta es muy importante, ya que si no la usamos, la letra como la "Ñ" y las letras con tilde no se codificarían bien. Algo tan simple como este texto: `Ñ á é í ó ú` se vería así:

Ã‘ Ã¡ Ã© Ã© Ã³ Ãº

Imagen 8: Tildes

## Favicon:

Para poner el pequeño ícono que aparece en la pestaña del navegador se tiene que tener una imagen pequeña (formato jpg, png, gif o bmp) y cuadrada. Dentro de los documentos de nuestros assets tenemos este png llamado favicon. Lo descargaremos y lo pondremos dentro de la carpeta del proyecto. Lo llamamos de la siguiente forma: `<link rel="shortcut icon" type="image/png" href="favicon.png" />`

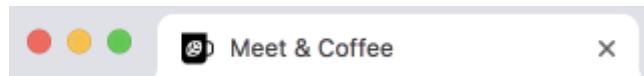


Imagen 9: Favicon

## Otros metadatos:

Además de lo que ya hemos aprendido, existen varios tipos de metadatos que se pueden incluir en el head.

Existen de autor `<meta name="author" content="Francisca Medina Concha">`, de descripción `<meta name="description" content="Comparte tus conocimientos tomando café">`, de palabras clave `<meta name="keywords" content="charlas, eventos, simposios, tecnología, co-work">` y más.

En este [link de w3schools](#) están detallados para que tengas mayor información.

## Resultado:

```
<!DOCTYPE html>
<html>
<head>
    <title>Meet & Coffee</title>
    <meta charset="utf-8">
    <link rel="shortcut icon" type="image/png" href="favicon.png">
    <meta name="author" content="Francisca Medina Concha">
    <meta name="description" content="Comparte tus conocimientos tomando café">
    <meta name="keywords" content="charlas, eventos, simposios, tecnología, co-work">
</head>
<body>
    <!-- Aquí va el contenido de la página web -->
</body>
</html>
```

## Trabajando en el body (textos)

Ya trabajamos con el head, ahora nos toca trabajar sobre el body, ahí donde le debemos añadir el contenido específico para el usuario. En la [guía de estilo](#) y la [maqueta final](#) podremos sacar el contenido necesario.

### Párrafos

En el `body` de nuestro `index.html` escribiremos las dos primeras frases.

```
<!DOCTYPE html>
<html>
<head>
  <title>Meet & Coffee</title>
  <meta charset="utf-8">
  <link rel="shortcut icon" type="image/png" href="favicon.png">
  <meta name="author" content="Francisca Medina Concha">
  <meta name="description" content="Comparte tus conocimientos tomando café">
  <meta name="keywords" content="charlas, eventos, simposios, tecnología, co-work">
</head>
<body>
  Descubre lo último en tecnología bebiendo café

  Charlas, eventos y simposios sobre tecnología
</body>
</html>
```

Guardaremos el archivo, recargaremos nuestro navegador y podremos observar que el texto se escribió, pero quedó todo junto.

Eso es porque no le hemos indicado que cada texto será un párrafo independiente.

1. Empecemos asigándole la etiqueta `<p>`, que indica párrafo a la primera frase: `<p>Descubre lo último en tecnología bebiendo café</p>`
2. Luego etiquetaremos el segundo párrafo, `<p>Charlas, eventos y simposios sobre tecnología</p>`
3. Guardaremos el archivo, recargaremos nuestro navegador y podremos observar como ya podemos escribir hacia abajo.

Todo lo que esté al interior de la marca `p` formará parte del párrafo.

## Auto-completado con Lorem Ipsum

Muchas veces nos tocará hacer un sitio web pero nuestros clientes no tendrán el contenido definitivo. Cuando nos suceda eso podemos llenar con texto ficticio. Lorem Ipsum es un autocompletado de texto que se usa habitualmente en diseño gráfico o demostraciones de tipografías. Nosotros la utilizaremos para añadir texto de prueba en nuestras páginas de forma rápida.

Cuando añadimos texto en nuestros archivos, existe la opción de auto-completar texto utilizando Lorem ipsum.

Crearemos un nuevo párrafo para probarlo, luego insertaremos `lorem` y autocompletaremos con la tecla *Tab*.

```
<p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</p>
```

También puedes extraer texto Lorem Ipsum desde [lipsum.com](http://lipsum.com)

En nuestro caso, no necesitamos de texto ficticio ya que tenemos todo el contenido a nuestra disposición en la maqueta y la guía de estilo, pero es muy importante considerar este recurso del Lorem Ipsum para futuros desarrollos.

## Encabezados

Una página web no solo tiene párrafos, también tiene headings (o encabezados), que son textos de mayor jerarquía que los párrafos.

Los encabezados son etiquetas que se escriben con la letra h y la sigue un número del 1 al 6.

`<h1>` , `<h2>` , `<h3>` , `<h4>` , `<h5>` y `<h6>` .

El `<h1>` es el encabezado con mayor jerarquía, o sea, el más importante. Mientras va aumentando el número del encabezado va disminuyendo su jerarquía, siendo el `<h6>` el menos relevante. La jerarquía de los encabezados se ve reflejado tanto en su estilo por defecto, como en su semántica.

Entonces, en nuestro index.html, vamos a cambiar nuestros párrafos por encabezados.

- El primer párrafo lo transformaremos en un `<h1>` , ya que es el de mayor tamaño y mayor relevancia.

```
<h1>Descubre lo último en tecnología bebiendo café</h1>
```

- Y cambiaremos el segundo párrafo por un `<h2>` :

```
<h2>Charlas, eventos y simposios sobre tecnología</h2>.
```

- Vamos a guardar nuestra página y la reabriremos en nuestro navegador.

Ahora podemos ver las notables diferencias entre el `<h1>` y el `<h2>`.

Añadiremos, entonces, el resto de los textos con las siguientes etiquetas.

- Texto ubicación:

```
<h2>¿Dónde nos juntamos?</h2>
<p>Todos los martes y viernes, de 19:00 a 22:00 en We Work, Calle Baker 133, Providencia, Santiago.
</p>
```

- Texto próxima charla:

```
<h2>Próxima charla</h2>
<h3>Big Data para Noobs</h3>
<h4>Rafaela Valdés</h4>
<h5>Data scientis, phD Theoriccal Physics, Carneige Mellon University</h5>
<p>En esta charla revisaremos las técnicas y tecnologías que hacen que sea económico hacer frente a los datos de una escala extrema.</p>
```

- Texto eventos

```
<h2>Eventos anteriores</h2>
<h3>Simposio Vegan DB</h3>
<h3>Machine learning 101</h3>
<h3>Scrum sin scream</h3>
```

Guardaremos el archivo y recargaremos, ahora podemos ver los encabezados y párrafos generados en el navegador.

Si has seguido todas las indicaciones tu página debería verse así:

```
<!DOCTYPE html>
<html>
<head>
  <title>Meet & Coffee</title>
  <meta charset="utf-8">
  <link rel="shortcut icon" type="image/png" href="favicon.png">
  <meta name="author" content="Francisca Medina Concha">
  <meta name="description" content="Comparte tus conocimientos tomando café">
  <meta name="keywords" content="charlas, eventos, simposios, tecnología, co-work">
</head>
<body>
  <h1>Descubre lo último en tecnología bebiendo café</h1>
  <h2>Charlas, eventos y simposios sobre tecnología</h2>
```

<h2>¿Dónde nos juntamos?</h2>

<p>Todos los martes y viernes, de 19:00 a 22:00 en We Work, Calle Baker 133, Providencia, Santiago.</p>

<h2>Próxima charla</h2>

<h3>Big Data para Noobs</h3>

<h4>Rafaela Valdés</h4>

<h5>Data scientis, phD Theoriccal Physics, Carneige Mellon University</h5>

<p>En esta charla revisaremos las técnicas y tecnologías que hacen que sea económico hacer frente a los datos de una escala extrema.</p>

<h2>Eventos anteriores</h2>

<h3>Simposio Vegan DB</h3>

<h3>Machine learning 101</h3>

<h3>Scrum sin scream</h3>

</body>

</html>

## Trabajando en el body (imágenes)

Ahora, que ya hemos escrito el contenido de nuestra página, vamos a poner las imágenes. (Recuerda que nuestras imágenes están almacenadas en esta carpeta de **assets**).

Ahora, antes de nuestro `h1` colocaremos la etiqueta que corresponde a una imagen de la siguiente forma:

```
<img src="" alt="">
```

La etiqueta `<img>`, a diferencia de los párrafos y encabezados, no tiene un cierre. Es sólo la etiqueta y sus parámetros.

Podemos identificar, de momento, dos parámetros fundamentales.

- `src` : Que significa origen.
- `alt` : Que significa texto alternativo.

## Imágenes desde Internet

Bueno, como primera prueba vamos a poner una imagen desde internet, que llamaremos desde nuestra etiqueta `<img>`. iremos a [Google](#) y buscaremos café.

Buscamos la que más nos guste, la abrimos y con click derecho le damos a "**Copiar dirección de imagen**". Esto nos copiará un link directo a la foto.

Si lo colocamos en el navegador veremos esto:

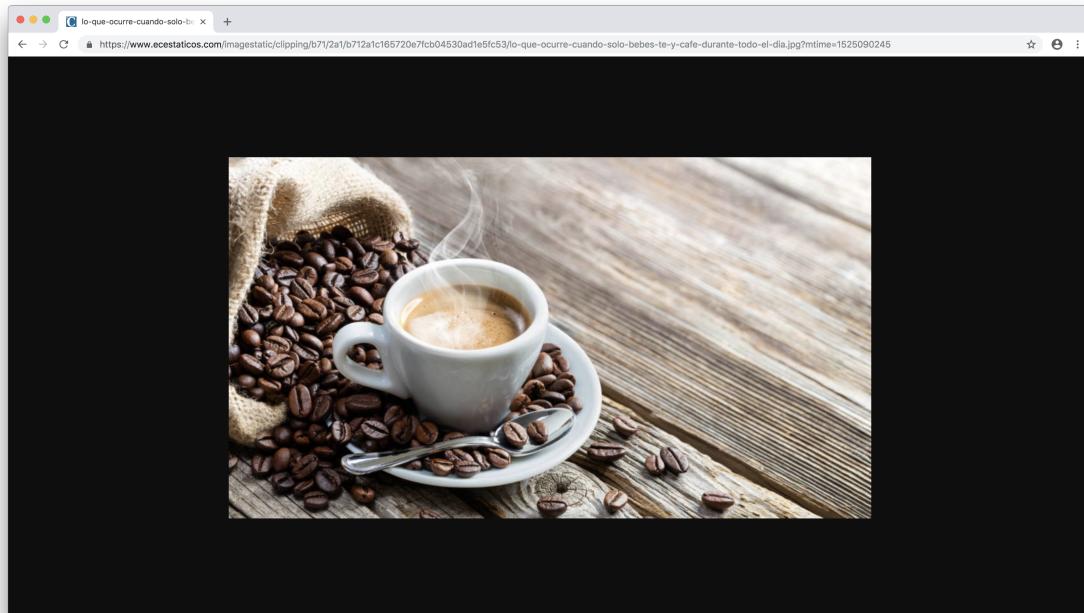


Imagen 10: Agregar imagen desde internet

Ahora, dentro del parámetro `src`, le daremos como valor el link de la foto que hemos copiado.

```

```

Vamos a meternos a nuestro navegador y veremos que la imagen ha sido incluida, del porte que venía.

## Agregando imágenes a un subdirectorio

Como ya tenemos las imágenes que utilizaremos en nuestro sitio en los **assets** vamos a borrar esta fotografía que no corresponde.

Además, al mostrar una imagen desde Internet nos arriesgamos a que ésta pueda ser borrada, por lo tanto, es mucho mejor tener las imágenes dentro de la carpeta de nuestro proyecto, así que vamos a descargar las imágenes del proyecto.

Como buena práctica, no dejaremos las imágenes dentro de la raíz de la carpeta, sino crearemos subdirectorios. ¿Por qué? Porque a medida que el proyecto va creciendo puede llegar a ser difícil mantener el orden, especialmente si trabajamos todos los archivos en la raíz del proyecto, por lo tanto, es importante clasificarlos.

En Atom, vamos a generar una nueva carpeta llamada **assets** al interior de la carpeta raíz del proyecto. A continuación, dentro de la carpeta **assets** recién creada, crearemos una nueva carpeta llamada **img**, donde ingresaremos todas las imágenes.

Una vez que tengamos las imágenes dentro de la carpeta correspondiente, escribiremos en nuestro archivo `<img src="" alt="">` y esta vez, en el atributo `src` le daremos el siguiente origen `assets/img/bg-hero.png`. Esto debería quedar así:

```

```

Vamos a actualizar el navegador y podremos ver que se ha desplegado la imagen.

Ahora, vamos a colocar las imágenes que restan, según cómo están dentro de nuestra [maqueta final](#).

Ingresa todas las imágenes en tu proyecto, excepto `coffee-cup.svg`, que será el logo de nuestra barra de navegación que añadiremos después. Será un buen trabajo para practicar.

Si realizaste todo bien, tu página debería verse de esta forma.

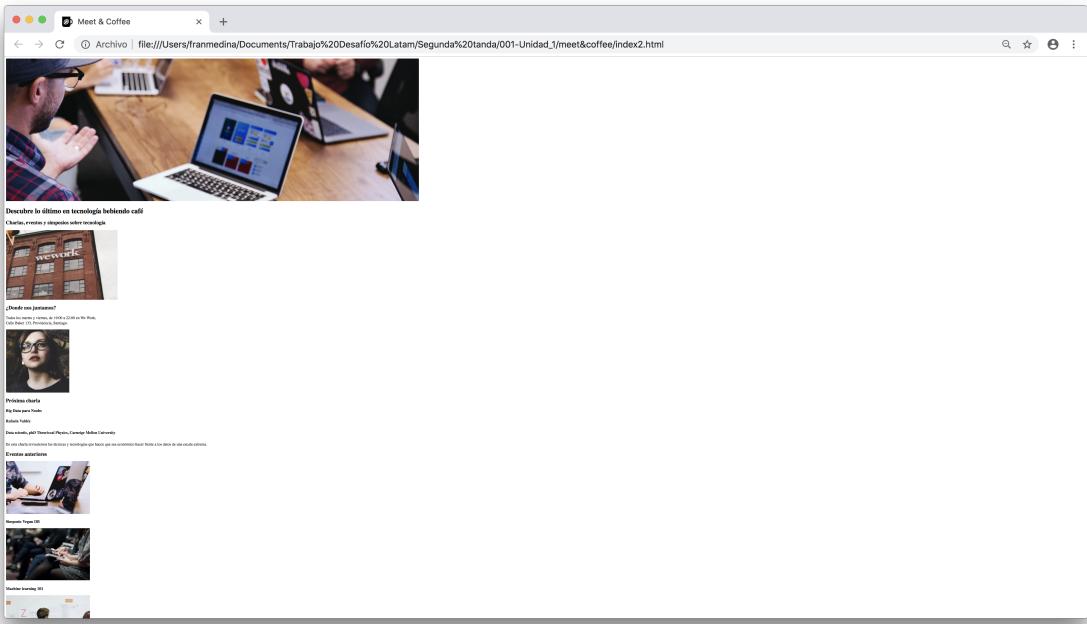


Imagen 11: Visualizar imágenes en página web

## Texto alternativo

Pero nos falta algo. El parámetro `alt`, significa texto alternativo y es muy importante, ya que los motores de búsqueda codificarán tu imagen a través de ese texto. También, si la imagen no se encuentra, mostrará el texto en vez de ella.

Por lo tanto, deberemos atribuirle un texto a cada una de las imágenes.

## Resultado

```
<!DOCTYPE html>
<html>
<head>
    <title>Meet & Coffee</title>
    <meta charset="utf-8">
    <link rel="shortcut icon" type="image/png" href="favicon.png">
    <meta name="author" content="Francisca Medina Concha">
    <meta name="description" content="Comparte tus conocimientos tomando café">
    <meta name="keywords" content="charlas, eventos, simposios, tecnología, co-work">
</head>
<body>
    
    <h1>Descubre lo último en tecnología bebiendo café</h1>
    <h2>Charlas, eventos y simposios sobre tecnología</h2>

    
    <h2>¿Dónde nos juntamos?</h2>
    <p>Todos los martes y viernes, de 19:00 a 22:00 en We Work, Calle Baker 133, Providencia, Santiago.</p>
```

```

<h2>Próxima charla</h2>
<h3>Big Data para Noobs</h3>
<h4>Rafaela Valdéz</h4>
<h5>Data scientis, phD Theoriccal Physics, Carneige Mellon University</h5>
<p>En esta charla revisaremos las técnicas y tecnologías que hacen que sea económico hacer frente a los datos de una escala extrema.</p>
```

```
<h2>Eventos anteriores</h2>

<h3>Simposio Vegan DB</h3>

<h3>Machine learning 101</h3>

<h3>Scrum sin scream</h3>
</body>
</html>
```

## Trabajando en el body (listas)

Ya hemos agregado la mayor parte del contenido de nuestra página index, pero aún nos queda conocer más etiquetas.

Ahora conoceremos lo que son las listas. Para ello conoceremos las etiquetas `<ul>`, `<ol>` y `<li>`.

- `ul` significa **unordered list** y la usaremos cuando tengamos que hacer un listado de elementos sin un orden particular.
- `ol` significa **ordered list** y como su nombre lo indica, lo utilizaremos cuando necesitemos tener una lista de elementos ordenados o enumerados.
- `li` significa **list item** y es un elemento de la lista de cualquiera de los dos tipos, ordenada o desordenada.

Hagamos un ejercicio rápido, independiente del proyecto que estamos trabajando. De momento, crearemos un nuevo documento dentro de la carpeta del proyecto de tipo html, y le daremos la estructura base. Ahí haremos una lista no ordenada y una lista ordenada.

Generaremos una lista no ordenada de las cosas que debemos comprar en el supermercado y una lista ordenada con nuestros animales favoritos.

```
<!DOCTYPE html>
<html>
<head>
    <title>Listas no-ordenadas y listas ordenadas</title>
    <meta charset="utf-8">
</head>
<body>
    <h1>Listas no-ordenadas y listas ordenadas</h1>

    <h2>Lista del supermercado</h2>
    <ul>
        <li>Lechuga</li>
        <li>Palmitos</li>
        <li>Tomate</li>
        <li>Espinaca</li>
    </ul>

    <h2>Mis animales favoritos</h2>
    <ol>
        <li>Gato</li>
        <li>Perro</li>
        <li>Conejo</li>
        <li>Ratón</li>
    </ol>
</body>
</html>
```

Recargaremos y podemos observar que en la lista no-ordenada no se jerarquizan los elementos y se le antepone una viñeta redonda y sólida, y en la lista ordenada se jerarquizan los elementos a través de números.

## Trabajando en el body (links)

Ahora, siguiendo en el archivo del ejercicio anterior de las listas, veremos una etiqueta muy importante, la etiqueta para los enlaces o links.

Los links nos permiten enlazar nuestra página a:

- Otros sitios, diferentes al nuestro.
- Otras páginas de nuestro sitio.
- Archivos.
- Secciones dentro de la misma página.
- Entre otros.

La etiqueta para los enlaces es: `<a>`, básicamente se escribe así:

```
<a href="link">Texto a mostrar</a>
```

Al interior de `href=" "` debe ir la ruta a la cuál seremos dirigidos. Y entre los símbolos `><` debería ir el contenido mostrado para hacer click.

### Links a otros sitios

Escribiremos lo siguiente:

```
<a href="https://www.w3schools.com/">Aprende más sobre desarrollo web en W3Schools</a>
```

Esto creará un link que nos redirigirá al sitio enlazado.

Pero, muchas veces queremos que los links se abran en una nueva pestaña, sobre todo cuando la página a la cual redirige es de un sitio distinto al nuestro. Esto se consigue añadiendo otro atributo a la etiqueta `<a>` llamado `target`. Y para que siempre se abra en una nueva pestaña debe llevar el valor `_blank`.

```
<a href="https://www.w3schools.com/" target="_blank">Aprende más sobre desarrollo web en  
W3Schools</a>
```

## Links a páginas dentro de nuestro sitio

Como creamos este archivo dentro de la carpeta de nuestro proyecto, no nos será complicado crear un enlace que nos redirija al `index.html`.

De la misma forma que hicimos el link a W3Schools, haremos un link a la página principal nuestro proyecto. Como estamos trabajando dentro de una mismo sitio, no será necesario añadir el atributo `target="_blank"` que abre en una nueva pestaña.

```
<a href="index.html" target="_blank">Link a la página principal de nuestro proyecto</a>
```

Recuerden que el valor de `href` es la ruta al archivo.

## Imágenes con links

No siempre el contenido mostrado para hacer clic es texto, a veces puede ser una imagen.

Haremos un link a Google con su logo clickeable.

- Escribiremos la etiqueta `<a>`, con su atributo `href` dirigiendo a [Google](#), como es un enlace a un sitio externo le daremos el atributo `target="_blank"`.
- Luego en el mismo Google buscamos el logo de Google y generamos la etiqueta `<img>` con sus atributos `src` y `alt`, dentro del contenido de la etiqueta `<a>`.

```
<a href="https://www.google.cl" target="_blank"></a>
```

Y listo, hemos conocido los enlaces que nos servirán siempre en nuestros desarrollos.

## Trabajando en el body (menú)

Ya hemos conocido las listas y los links, estos elementos nos servirán para añadir lo que nos falta de contenido en nuestra página index: el menú.

Volveremos a nuestro archivo `index.html`.

Esta página index, al ser de tipo "**Landing Page**" tiene un scroll hacia abajo. Cada link del menú representa una sección de la página.

Comencemos a utilizar estos elementos previamente aprendidos.

- Vamos a añadir una lista no-ordenada, con 5 ítems, al principio de la etiqueta `<body>`:

```
<ul>
  <li></li>
  <li></li>
  <li></li>
  <li></li>
  <li></li>
</ul>
```

- Dentro de los list item vamos a colocar la etiqueta `<a>`, de momento, como valor del parámetro `href` colocaremos el signo `#`.
- Finalmente, como contenido de la etiqueta pondremos uno a uno los nombres de las secciones.

Como vemos en nuestra [maqueta final](#), el primer elemento del menú es el ícono del café, de momento lo dejaremos como texto.

```
<ul>
  <li><a href="#">ícono</a></li>
  <li><a href="#">Ubicación</a></li>
  <li><a href="#">Próxima Charla</a></li>
  <li><a href="#">Eventos anteriores</a></li>
  <li><a href="#">Contacto</a></li>
</ul>
```

- Recargaremos el navegador para observar que se generó nuestra lista con los elementos mencionados.

Al agregarle de valor al atributo `href` el símbolo de `#` eso hace que al clickearlo recargue la página. De momento lo dejaremos así.

## Resultado:

```
<!DOCTYPE html>
<html>
<head>
    <title>Meet & Coffee</title>
    <meta charset="utf-8">
    <link rel="shortcut icon" type="image/png" href="favicon.png">
    <meta name="author" content="Francisca Medina Concha">
    <meta name="description" content="Comparte tus conocimientos tomando café">
    <meta name="keywords" content="charlas, eventos, simposios, tecnología, co-work">
</head>
<body>
    <ul>
        <li><a href="#">Ícono</a></li>
        <li><a href="#">Ubicación</a></li>
        <li><a href="#">Próxima Charla</a></li>
        <li><a href="#">Eventos anteriores</a></li>
        <li><a href="#">Contacto</a></li>
    </ul>

    
    <h1>Descubre lo último en tecnología bebiendo café</h1>
    <h2>Charlas, eventos y simposios sobre tecnología</h2>

    
    <h2>¿Dónde nos juntamos?</h2>
    <p>Todos los martes y viernes, de 19:00 a 22:00 en We Work, Calle Baker 133, Providencia, Santiago.</p>

    
    <h2>Próxima charla</h2>
    <h3>Big Data para Noobs</h3>
    <h4>Rafaela Valdés</h4>
    <h5>Data scientist, PhD Theoretical Physics, Carnegie Mellon University</h5>
    <p>En esta charla revisaremos las técnicas y tecnologías que hacen que sea económico hacer frente a los datos de una escala extrema.</p>

    <h2>Eventos anteriores</h2>
    
    <h3>Simposio Vegan DB</h3>
    
    <h3>Machine learning 101</h3>
    
    <h3>Scrum sin scream</h3>
</body>
</html>
```

Ya hemos agregado todo el contenido de nuestra página `index.html`. Revisemos lo que hemos creado.

Ya sé que te preguntarás, ¿Por qué se ve así, sin ningún diseño?

Es porque lo que hemos hecho ha sido entregarle el contenido a la página. Aún nos queda ver temas de estilo y estructura, que lo veremos en la unidad a continuación.

De momento te invito a realizar una pequeña actividad que consiste en armar tu CV en html. Recuerda utilizar la estructura básica de html ( `<head>` y `<body>` ) y todas las etiquetas que hemos conocido:

Patra el `<head>` :

- Título
- Codificación
- Favicon
- Metadatos

Para el `<body>`

- Párrafos
- Encabezados
- Imágenes
- Listas (ordenadas y no-ordenadas)
- Links