Sprint 1

**Creación de la primera web**

Ahora que estamos un poquito más familiarizados con *html* es hora de empezar el proyecto. Recuerda crear una carpeta nueva con el nombre del proyecto, luego ábrela en Visual Studio Code.

​

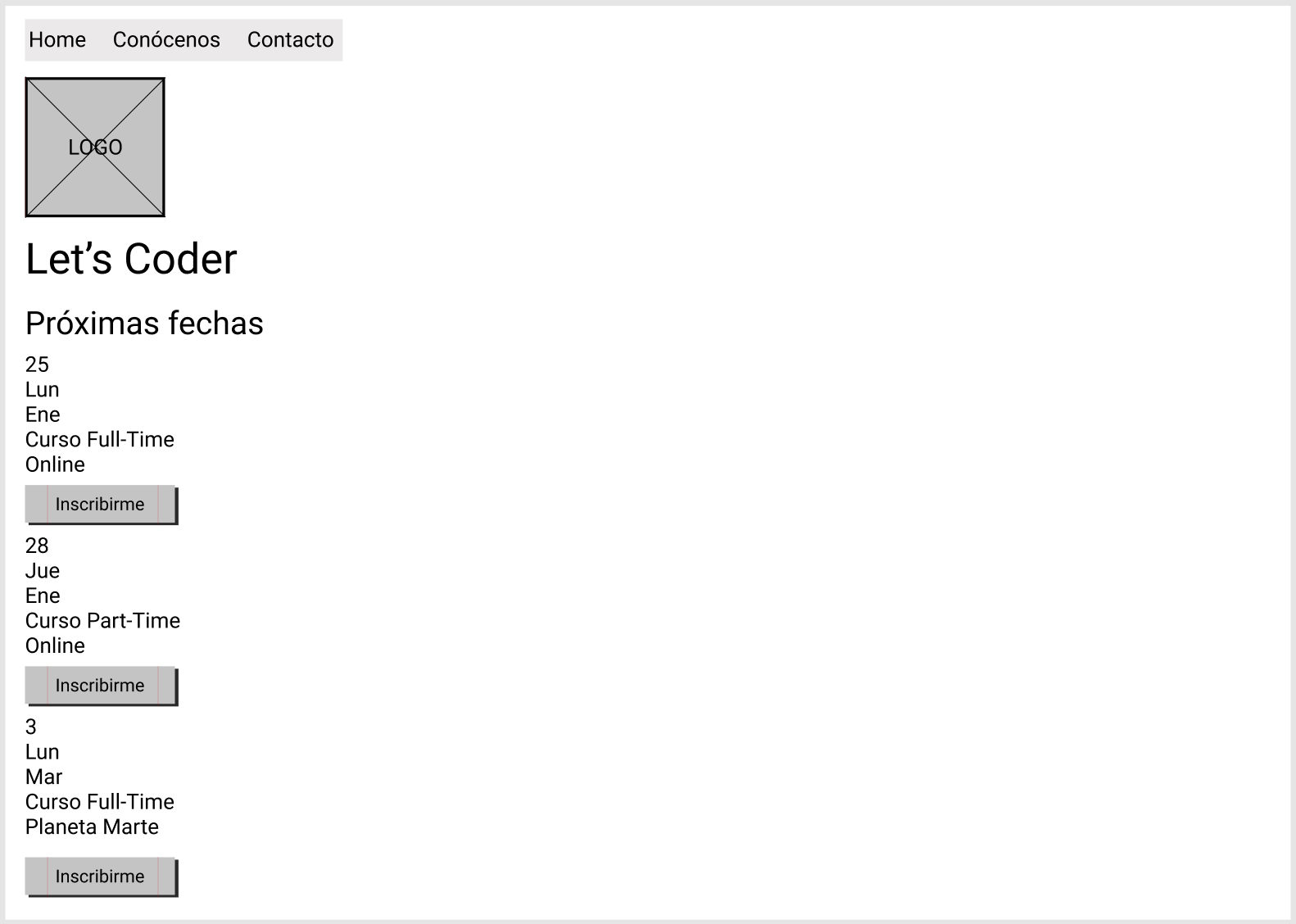
**Tarea 1: Crea home.html**

Una vez la carpeta del proyecto esté abierta en VSC, crea un nuevo archivo llamado **home.html**. Empieza por el "esqueleto" del un *html* i empieza a trabajar en el tag <body>.

**TIP:** VSC viene con un emmet pre instalado, así que conoce algunos atajos dependiendo de la extensión del archivo. Si el archivo es *.html* y escribes html:5 y luego la tecla *tab* creará la estructura automáticamente.

En caso que lo hayas olvidado, para ver tu código en tiempo real guarda tu trabajo y arranca **Live Server.**

Antes de pensar sobre el estilo y la disposición de los elementos, debes añadir todo el contenido de la página. No te preocupes por cómo se ve, simplemente intenta organizar los *tags* debidamente. Ahora mismo debería verse tal que así:



Puedes descargarte las imágenes aquí:

[imágenes proyecto](https://firebasestorage.googleapis.com/v0/b/gitbook-28427.appspot.com/o/assets%2F-MR_LfkA570QIKDFynHv%2F-MRqOYrX9vop4fTBSy-G%2F-MRqkXjGsF4i-s5Febqd%2Fimages.zip?alt=media&token=74aa15f0-2a86-4905-a703-6f9ea182909d" \t "_blank)

[images.zip - 2MB](https://firebasestorage.googleapis.com/v0/b/gitbook-28427.appspot.com/o/assets%2F-MR_LfkA570QIKDFynHv%2F-MRqOYrX9vop4fTBSy-G%2F-MRqkXjGsF4i-s5Febqd%2Fimages.zip?alt=media&token=74aa15f0-2a86-4905-a703-6f9ea182909d" \t "_blank)

Debe verse más o menos como el sketch anterior, simplemente unos cuantos tags en el archivo *html.* No le des importancia a que el texto de la barra de navegación no lleve a ningún lado, ya lo convertirás en links cuando crees más páginas. Lo mismo para los botones, ahora no sirven.

Como puedes ver, hay muchos elementos distintos: hipervínculos, imágenes, títulos, párrafos... intenta utilizar el tag correcto en cada ocasión. Puedes echarle un vistazo a esta [Cheat Sheet](https://drive.google.com/file/d/1NFSf7eipIfwcfnlFjO3g6kmGe1GIzopW/view) si necesitas ayuda.

Aquí están los sketches de las otras páginas, que tendrás que hacer a continuación.

[Let's Coder - sketches](https://firebasestorage.googleapis.com/v0/b/gitbook-28427.appspot.com/o/assets%2F-MR_LfkA570QIKDFynHv%2F-MRqOYrX9vop4fTBSy-G%2F-MRqj8cc6-a_fjw5yMSz%2Fhtml%20sketches.zip?alt=media&token=0b874a2a-2b7d-4a79-b4fe-88d642a1e770" \t "_blank)

[html sketches.zip - 254KB](https://firebasestorage.googleapis.com/v0/b/gitbook-28427.appspot.com/o/assets%2F-MR_LfkA570QIKDFynHv%2F-MRqOYrX9vop4fTBSy-G%2F-MRqj8cc6-a_fjw5yMSz%2Fhtml%20sketches.zip?alt=media&token=0b874a2a-2b7d-4a79-b4fe-88d642a1e770" \t "_blank)

​

**Tarea 2: crear conocenos.html y contacto.html**

Siguiendo los pasos de la tarea anterior, crea dos archivos más: **conocenos.html** y **contacto.html.**

Ya tienes los sketches para estas dos páginas nuevas, con sus correspondientes imágenes

​

**Tarea 3: Añade los links**

Ahora que tenemos la estructura y contenido, es hora de vincular todas tus páginas entre si. Para ello, utiliza el tag <a>. Hazlo dentro de la navbar <nav> , así cuando hagas click a *home* irá a *home.html*, y lo mismo para las demás. Esto hará que tu web sea un poco más dinámica.

​

**Tarea 4: Valida tu código**

Existen validadores online donde puedes ver si hay algún error o advertencia en tu código, tales como [el HTML validator](https://validator.w3.org/#validate_by_input). Puedes copiar y pegar tu código para ver si está escrito como debería. Revisa todos los errores y advertencias que te salgan.

A partir de ahora, crea el hábito de validar tu código regularmente, ya que te evitará dolores de cabeza resolver pequeños problemas en vez de intentar resolverlos todos a la vez al final.

​

**Añadir estilos y mejoras visuales**

Hasta ahora hemos añadido el contenido. Pero como colocamos todo en su sitio? Pues bien, es hora de aprender **CSS**.

Puede que hayas visto diferentes manera de dar estilo a una página web, desde atributos de estilo dentro de los tags hasta el tag <style> dentro del <head> donde puedes escribir css. Desde ahora vamos a utilizar lo que es más común y preferible, el **archivo .css externo**.

​

**Tarea 1: Añade estilo a tu proyecto actual**

Dentro de la carpeta del proyecto, crea una carpeta que se llame *styles* y dentro el archivo *main.css* (o *style.css*). Una vez lo tengas, conectalo a tus html y sigue estos wireframes para la organización de los elementos

[Let's Coder - wireframes](https://firebasestorage.googleapis.com/v0/b/gitbook-28427.appspot.com/o/assets%2F-MR_LfkA570QIKDFynHv%2F-MRqOYrX9vop4fTBSy-G%2F-MRqjA-njytXrobMKQL0%2Fhtml%20wireframes.zip?alt=media&token=d6cde824-b6ca-4085-8a89-7fc7578e5d60" \t "_blank)

[html wireframes.zip - 288KB](https://firebasestorage.googleapis.com/v0/b/gitbook-28427.appspot.com/o/assets%2F-MR_LfkA570QIKDFynHv%2F-MRqOYrX9vop4fTBSy-G%2F-MRqjA-njytXrobMKQL0%2Fhtml%20wireframes.zip?alt=media&token=d6cde824-b6ca-4085-8a89-7fc7578e5d60" \t "_blank)

Si tienes la misma estructura en todos los html con las mismas clases, te será mucho más fácil tener un estilo homogéneo entre todas las páginas.

Antes de saltar a la siguiente tarea, valida tu trabajo con el [CSS validator](https://jigsaw.w3.org/css-validator/#validate_by_input) y comenta tu trabajo con el mentor.

​

**Crea tu propio estilo**

Ahora ya hemos visto la base tanto de *html* como de *css*. Has podido aplicar el estilo de un wireframe a la perfección, que es lo que a menudo se te va a pedir si en el proyecto hay un diseñador gráfico. Aquí no tenemos, así que es una gran oportunidad para ver tu gran talento artístico!

​

**Tarea 1: Un nuevo archivo**

No debes borrar todo el css que ya has hecho. Crea otro archivo *css* y no olvides hacer el link en cada uno de los archivos *html*.

Para esta tarea, y a partir de ahora, puedes hacer cambios estructurales en el *html* para que se ajuste lo máximo posible a tu diseño. El contenido debe permanecer igual.

Una vez hayas acabado, contrástalo con tu mentor para que te de inputs sobre tu diseño. Tened en cuenta que no vamos a juzgar el diseño en sí, no somos diseñadores. Aunque la totalidad de la página web debe homogénea y no romper con los estándares.