



Curso de Front-End

Clase 01 (Adelanto)

¡Muchas gracias por habernos elegido!

Esperamos que puedan **disfrutar** y **aprovechar** este curso al máximo, y ojalá logremos transmitirles la **pasión** que nosotros sentimos por estos temas.

Si en algún momento del curso ven algo que les gustaría **mejorar**, por favor háganoslo saber lo antes posible. Además, al finalizar tendrán una **encuesta** de satisfacción anónima para hacernos llegar sus comentarios.



Presentaciones



Presentarnos

(30 segundos cada uno)

Nombre | Qué hago | Qué espero del curso



¿Conseguir trabajo? ¿Descubrir de qué se trata la programación? ¿Vincularse mejor con programadores?



Pedidos

Pedidos 🙏



- Respetar los **horarios** 🕒 de clase.
- ¡**Participen** mucho! Queremos oír la **voz** de todos. 🗣️
- Queremos que compartan su **pantalla** 🖥️. Queremos ver lo que hicieron, y si está mal, mejor. De los **errores** se aprende.
- La clase es mucho más **divertida** si todos participan. 🎉
- En la **interacción** con ustedes es donde podemos aportar más **valor**. Para tener una clase “unilateral” (dónde sólo habla el docente), probablemente les rinda más un curso de Udemy.



Software necesario



Software necesario

Antes de la primera clase, todos los alumnos deberán instalar:

- Google **Chrome** (google.com/chrome).
- Microsoft **Visual Studio Code** (code.visualstudio.com).
 - VSC es un editor de código abierto. Es como un editor de texto pero más avanzado.
 - Aprenderemos más sobre VSC dentro de un rato.

Además, recomendamos instalar **Microsoft Teams** para *desktop* y para *mobile* (ver las siguientes diapositivas).



Uso de Microsoft Teams



Uso de Microsoft Teams (1/4)

Desde la fundación de Hack Academy en el año 2016, hemos usado [Slack](#) como la principal herramienta de comunicación con los alumnos.

A partir del año 2020, hemos empezado a incorporar, de forma paulatina, el uso de [Microsoft Teams](#), ya que permite manejar en una misma herramienta todo lo referente a **mensajería**, intercambio de **materiales** y **videollamadas**.

Independientemente de la herramienta utilizada, es importante respetar algunas buenas prácticas para hacer la comunicación más eficiente, las cuales se listan en la siguiente diapositiva.





Uso de Microsoft Teams (2/4)

Buenas prácticas:

- Usar su **nombre completo** + **foto** de perfil.
- Escribir mensajes en los **canales** correspondientes.
- Compartir **código** con el formato adecuado (no como “texto plano”).
- Escribir las **consultas** en un **único mensaje**. Evitar escribir varios mensajes para una misma consulta. Ej: No escribir “*Hola*”, “*¿Cómo están?*”, “*Tengo una consulta*”, “*No logré hacer funcionar...*”, en 4 mensajes diferentes. Con esto se evitan las “cataratas” de notificaciones.



Uso de Microsoft Teams (3/4)

Buenas prácticas (continuación):

- Respetar los **hilos** (*threads*) de comunicación al responder un mensaje.
- **Evitar** hacer **consultas** a los **docentes vía mensajes privados**. 🙏 🙏 🙏
Por consultas administrativas, escribir a hola@ha.dev.
- No sientan vergüenza de usar el **canal de dudas**. Esto permite que cualquier persona pueda responder, ¡incluso otro alumno!
- Si bien Slack y Teams se pueden usar vía la web, para lograr una mejor experiencia, recomendamos descargar las **apps** para **desktop** y **mobile**.



Uso de Microsoft Teams (4/4)

Buenas prácticas (continuación):

- Las **notificaciones** de **Teams** las recibirán en su **casilla de correo @student.ha.dev** (que es una casilla de [Microsoft Outlook](#)).

👉 Recomendamos que configuren dicha casilla para que los mensajes entrantes se **redirijan** (*forwardeen*) automáticamente a su casilla personal (ej: a su casilla de Gmail o Hotmail). De esta forma no se perderán las notificaciones que hagamos por Teams.






Clases por videollamada

Recomendaciones

Clases por videollamada (1/2)

Recomendaciones para mejorar la experiencia de las clases por videollamada:

-  Usar un **Monitor Externo** o TV, de tal forma que en una pantalla puedan seguir la clase y en la otra pantalla hacer sus ejercicios.
 En realidad, esto es sumamente útil más allá de las videollamadas. Casi todos los programadores trabajan con dos pantallas (¡o más!). Es de las mejores inversiones que pueden hacer, incluso antes de actualizar su PC. No deberían invertir demasiado, se pueden conseguir monitores nuevos y usados a muy buen precio.
-  Asegúrense de que su **webcam** y **micrófono** funcionen correctamente. No precisan nada sofisticado; en general, los que vienen integrados en una notebook son suficientes. A veces también es cómodo contar con unos auriculares o *headset*.



Clases por videollamada (2/2)

Recomendaciones (continuación):

- Cerrar programas de **Torrents** o aplicaciones que demanden mucho ancho de banda como Netflix o YouTube. Para medir la velocidad de su conexión a Internet sugerimos entrar a: <https://www.speedtest.net>.
- Desactivar las instalaciones automáticas del sistema operativo, como puede ser **Windows Update**.
- Reiniciar su **router** y/o **modem** un rato antes de cada clase. La forma más sencilla de hacer esto es “desenchufar y enchufar” el aparato. Esto puede ser útil para prevenir que la conexión a internet se tranque o se corte en medio de la clase.



¿Conocimientos previos?

*“El curso está hecho para personas **sin conocimientos previos de programación**”*


Se requiere manejo básico de **Internet** y del **Sistema Operativo** (Windows, Linux o Mac). Por ejemplo: mover archivos de lugar, cambiarle el nombre a un archivo, descomprimir una carpeta.

Saber **inglés**  es una **gran ventaja**, sobre todo a la hora de buscar trabajo en la Industria IT.



Tips generales

Tips generales (1/6)

- ¡Hagan **muchas preguntas!** (no sientan vergüenza).
- **Googleen** mucho. También consulten [Stack Overflow](#).
- Intenten **ayudarse** entre ustedes. Enseñar es una gran forma de aprender.
- No copien/peguen código salvo que entiendan o sepan lo que están copiando. Copiar código no está mal, pero sean criteriosos.
-  Cuidado con las actualizaciones automáticas del sistema operativo. Sobre todo en Windows. Eviten actualizar su equipo en medio de la clase.

Tips generales (2/6)



¡Sean pacientes!

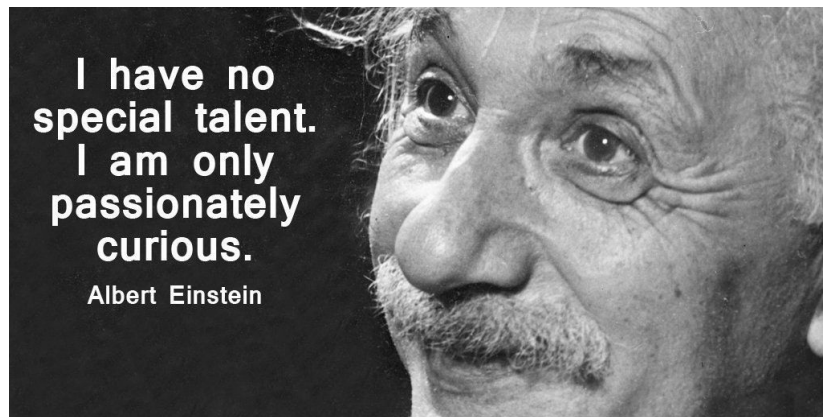


¡No hagan esto!

Para la mayoría no es fácil al principio.
Saber **manejar bien la frustración** es fundamental.

Tips generales (3/6)

¡Sean curiosos!



Para programar no hace falta ser un genio ni un erudito en matemáticas.
Más importante es tener **curiosidad** a la hora de **resolver problemas**.

Tips generales (4/6)



¡Practiquen mucho!



¿Cómo se aprende a tocar la guitarra?

Recuerden que no mandamos deberes, exámenes, pruebas, entregas, etc.

Pero eso no quiere decir que no practiquen por su cuenta. Lo ideal es conseguir un **proyecto propio** (un sitio web para ustedes o para alguien más). Es inevitable notar la diferencia entre los que practican y los que no.

Tips generales (5/6)

Por las dudas lo repetimos...

¡Practiquen mucho!

👉 Sugerimos que practiquen en sus casas aprox. **6 horas semanales**.



¡No falten a clase!

Es difícil re-engancharse si faltan, aunque se hagan repasos.

👉 Los **certificados de asistencia** se dan a los alumnos que lo soliciten (son la minoría) y que además hayan asistido a un **90%** de las clases.



Uso del teclado



Uso del teclado (1/4)

A la hora de programar, es necesario escribir caracteres especiales como:

< > \ / () [] | { } # \$ & * : ; + = " ' ` - _

Por lo tanto, es fundamental identificar estos caracteres con rapidez.

La mejor recomendación que les podemos hacer es que configuren su teclado de tal manera que **cuando presionen un carácter en el teclado físico, dicho carácter aparezca en la pantalla.** 🙌 Por favor no intenten recordar nada de memoria.



Al presionar esta tecla, debería aparecer un “punto y coma” en su pantalla.



Uso del teclado (2/4)

Si su teclado físico tiene las teclas en inglés 🇺🇸...



Generalmente
porque no tiene eñe.

...recomendamos configurarlo como **U.S. International** 🙌

Ver instrucciones en la siguiente diapositiva.



Uso del teclado (3/4)

Para los que tengan **teclado físico en inglés** 🇺🇸 recomendamos:

- Configurado en su sistema operativo como **U.S. International** o Estados Unidos Internacional. 👉 [Ver video de configuración](#) en Windows 10.
- Borrar otros teclados de inglés/español que hayan configurado anteriormente, ya que la configuración **U.S. International** sirve tanto para escribir en inglés como en español. 🇪🇸 🇬🇧 🇪🇸 🇬🇧 🇪🇸

Gracias a esta configuración **no tendrán que estar cambiando el idioma** ni acordarse de memoria donde está cada carácter en el teclado.

Uso del teclado (4/4)

- **Escribir rápido** con el teclado les ayudará mucho.
Pruébense aquí: <http://www.typingtest.com>.
- Acostúmbrense a los **atajos** (shortcuts) del teclado:
 - CTRL + C = Copiar.
 - CTRL + X = Cortar.
 - CTRL + V = Pegar.
 - CTRL + S = Guardar.
 - CTRL + Z = Deshacer.
 - CTRL + T = Abrir nueva pestaña (tab).
 - CTRL + W = Cierra la pestaña actual.
 - F5 = Recargar Página.

En Mac: sustituir CTRL por CMD. Para recargar la página usar: CMD+R.




Vamos a empezar a usar el mouse
cada vez menos.



Objetivos del curso

Objetivos del curso

- Introducción a la **maquetación** (usando HTML y CSS).
- Introducción a la **programación** (usando JavaScript).
- Maquetar un sitio como [éste](#).
- Aprender muchos **tips** para seguir avanzando por tu cuenta.
- Generar **buenos hábitos** de trabajo a la hora de maquetar y programar.
Ej: buscar en Google / Stack Overflow, escribir código de forma prolija (siguiendo buenas prácticas), etc.
- Poder responder: ¿es esto para mí?

 **Importante:** este es un curso introductorio y por lo tanto no tiene como objetivo la inserción laboral inmediata del egresado.

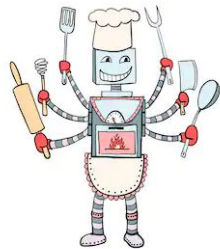
¿Qué es programar?

¿Qué es programar? (1/3)

Programar es “*crear un conjunto de instrucciones que serán ejecutadas por una computadora*”.

Por el momento, las computadoras son máquinas bastante “tontas”. Hay que decirles, con lujo de detalles (y **sin ambigüedades**), lo que queremos que hagan.

Analogía: Escribir un programa es como escribir una **receta de cocina** para una persona que nunca cocinó algo en su vida. Imagínense que en lugar de una persona es un robot. No alcanza con especificar los ingredientes y las cantidades. Es necesario explicar con precisión cada uno de los pasos que hay que realizar. Ejemplo: “*Girar la cuchara 90 veces en sentido horario a una velocidad de 300 rpm. Si quedan grumos, girar en sentido anti-horario a 120 rpm hasta que desaparezcan, etc*”. A la persona (o al robot) no le deben quedar dudas sobre cómo proceder.





¿Qué es programar? (2/3)

Así como los humanos disponen de distintos lenguajes (idiomas) para comunicarse entre sí, a la hora de darle instrucciones a una máquina (PC) también es necesario elegir un **lenguaje de programación**. Algunos ejemplos son: JavaScript, Java, PHP, Python, C++ y Ruby.

Tanto los lenguajes “humanos” (**lenguajes naturales**) como los lenguajes de programación disponen de **sintaxis** y **semántica**.

La ventaja de los lenguajes de programación es que permiten dar instrucciones **sin ambigüedades**. En cambio, los lenguajes naturales son ambiguos.

¿Qué es programar? (3/3)

Tipos de lenguajes de programación:

- **Código de máquina / Código binario**

- Es el menor nivel de abstracción. Consiste en en 1's y 0's.

```
0110001100
1011010110
1111011110
```

- **Lenguaje Assembly**

- Representación simbólica de código de máquina.

- **Lenguajes compilados**

- Lenguajes de alto nivel que necesitan ser compilados a código de máquina.
- Ej: C, C++, Go, Fortran, Pascal, Java.



- **Lenguajes interpretados**

- Lenguajes de alto nivel que no necesitan ser compilados. Los programas residen en la memoria RAM.
- Ej: JavaScript, Python, PHP, Ruby.

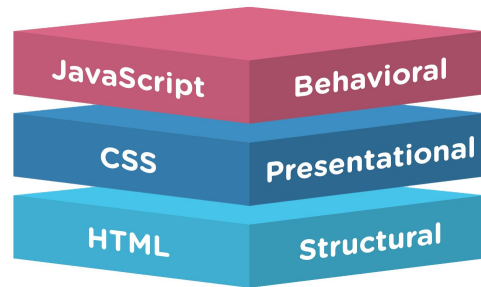
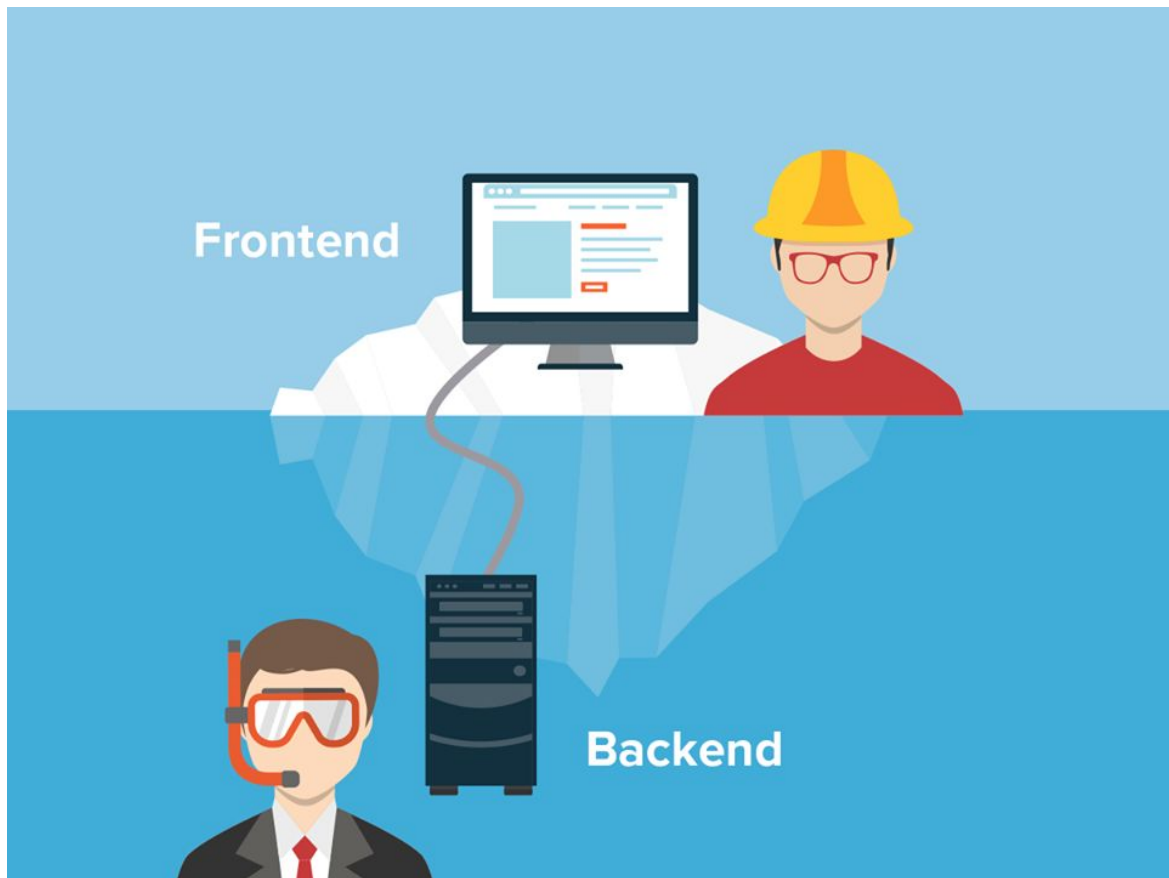


Más nivel de abstracción





Front-End



Estructura de una página web

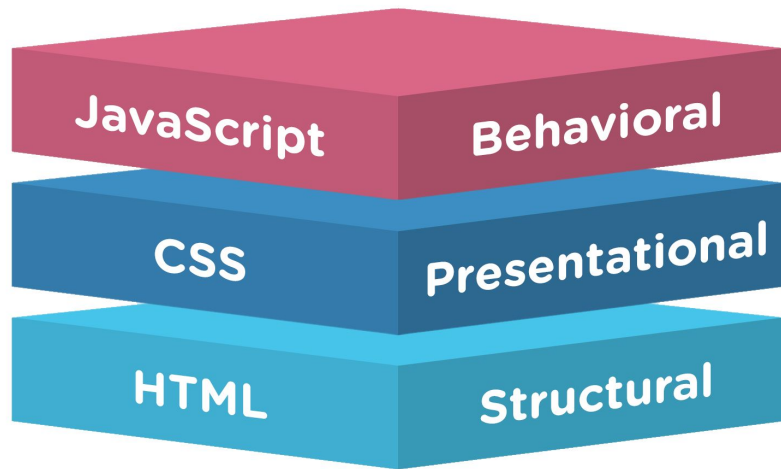
Contenido: Texto, imágenes, videos.

+ **HTML**: Estructura + Semántica

+ **CSS**: Presentación + Diseño

+ **JS**: Interacción

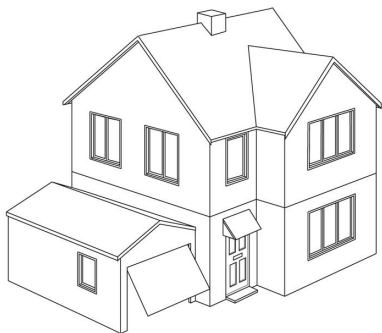
= Página web



Nota: Esta es la estructura básica de una página desde el punto de vista de un desarrollador Front-End. Un desarrollador Back-End también consideraría, por ejemplo, el guardado (persistencia) de los datos.

Estructura de una página web

Analogía con la construcción de una casa:



HTML

Qué



CSS

Cómo



JavaScript

Interacciones

HTML → Qué

CSS → Cómo



HTML



HTML: “HyperText Markup Language”

Qué es:

- Es el lenguaje nativo de los browsers (creado en 1993).
- Se utiliza para **describir la estructura y contenido** de las páginas web.
- Está compuesto por **etiquetas** (*tags*).
- Se escribe en un archivo de texto con extensión `.htm` o `.html` (sobre todo la segunda opción).

Qué no es:

- No es un lenguaje de programación (es un lenguaje de marcado o etiquetas).

HTML – Documentación

- Documentación en **Mozilla Developer Network** (muy completa):
<https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5> (Inglés)
<https://developer.mozilla.org/es/docs/HTML/HTML5> (Español)
- Documentación en **W3 Schools** (más básica; más simple):
<http://www.w3schools.com/html/default.asp>

IMPORTANTE

Empiecen a acostumbrarse a consultar la **documentación** de cada tecnología.
En general, suelen ser mejores las versiones en **inglés** 🇬🇧.