



[Instituto tecnológico]

Apuntes de clase:

Introducción al desarrollo web

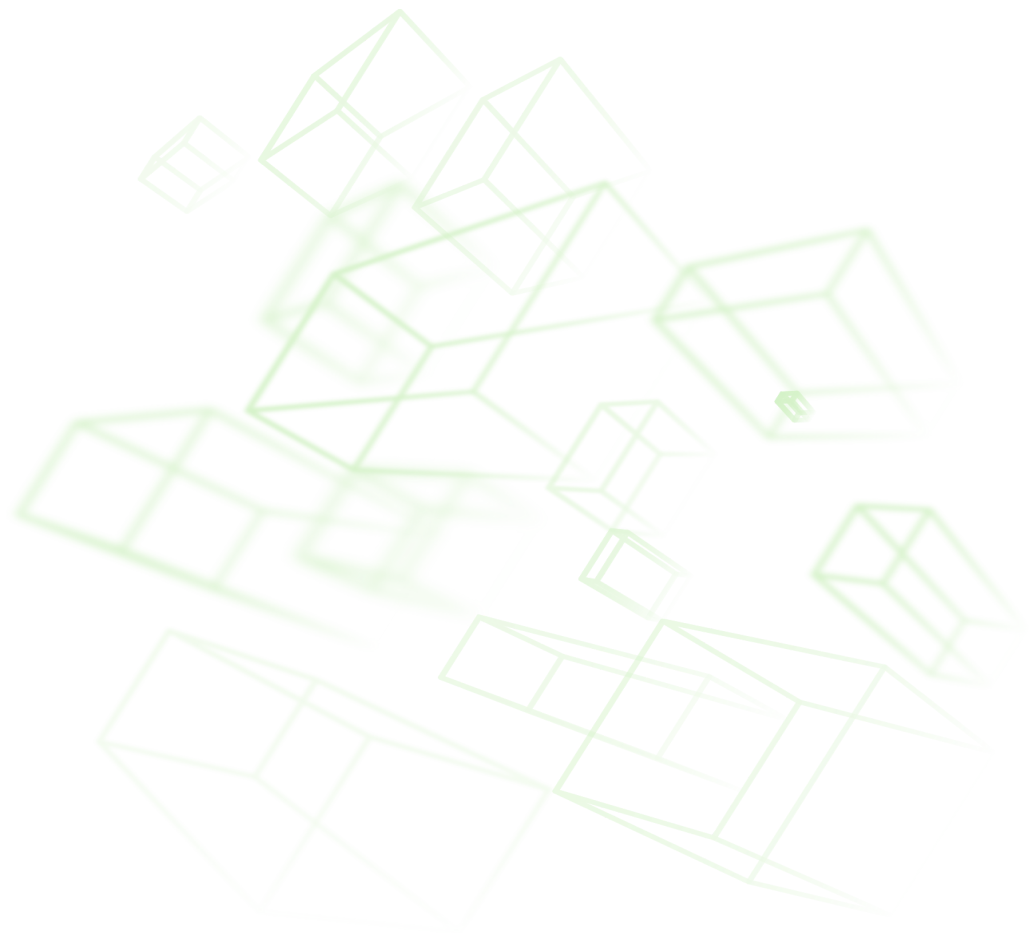


Tabla de contenido

Campus	2
Curso Desarrollador web	2
Introducción	4
¿Qué es una página web?	4
¿Quiénes se encargan del diseño web?	5
Desarrollo Web	6
HTML5 (Contenido y estructura)	7
CSS3 (estilo)	8
Javascript para el diseño web	9
Estructura básica de un proyecto WEB	9
Preparando el entorno: Herramientas necesarias	10
Editor de código	10
Un poco de historia (contenido extra)	11
¿Qué es internet?	11
La evolución de los lenguajes web	12
Backend vs Frontend	13
Navegadores Web y dominios (extra)	14
Descarga del repositorio "hola_mundo_web"	15
¡Hasta la próxima!	15
Links de interés	15

Campus

Este y otros apuntes más lo podrán encontrar en el campus de Inove, ingresando al siguiente link:

<https://inove.com.ar/campus>

Curso Desarrollador web

El curso de Desarrollador Web está compuesto por 24 horas de cursada (2 meses) más el mínimo tiempo recomendado por fuera para profundizar en la materia (24 horas de ejercitación). La cursada a su vez se divide en horas enfocadas a contenidos teóricos y de ejercitación. El objetivo de cada clase es que se lleven un concepto nuevo que puedan aplicarlo y despejar sus dudas en el mismo día, para que luego puedan continuar ampliando y fijando la idea en casa con ejercitación. Se espera que dediquen un rato de cada semana a realizar ejercitación que los ayudará a comprender mejor la teoría y el día de mañana serán valiosas horas de experiencia que los ayudaran en su vida personal.

Desarrollador Web

24 horas cursada
(2 meses)

15 horas teoría
9 horas práctica

+

24 horas ejercitación

**48 horas
experiencia**

Certificado de Asistencia

Completar al menos el 75% de los cuestionarios para validar presentismo (hay un cuestionario por clase)



Cuestionario para validar presentismo

0 de 0 Intentados

Certificado de Conocimientos alcanzados

- Completar un proyecto terminada la cursada.
- El proyecto es individual y personalizado.
- El alumno puede proponer un proyecto o elegir uno de la lista recomendada.
- Concluido este curso, tienen 2 meses para realizarlo.

Cada cursada finaliza con un trabajo práctico integrador que deben presentar para poder obtener la certificación del curso, caso contrario recibirán un certificado de asistencia si han asistido completado al menos al 75% de los cuestionarios del campus.

Cuanto mayor tiempo le dediquen a la ejercitación planteada para hacer durante la semana, más fácil será para ustedes el desarrollo del proyecto final.

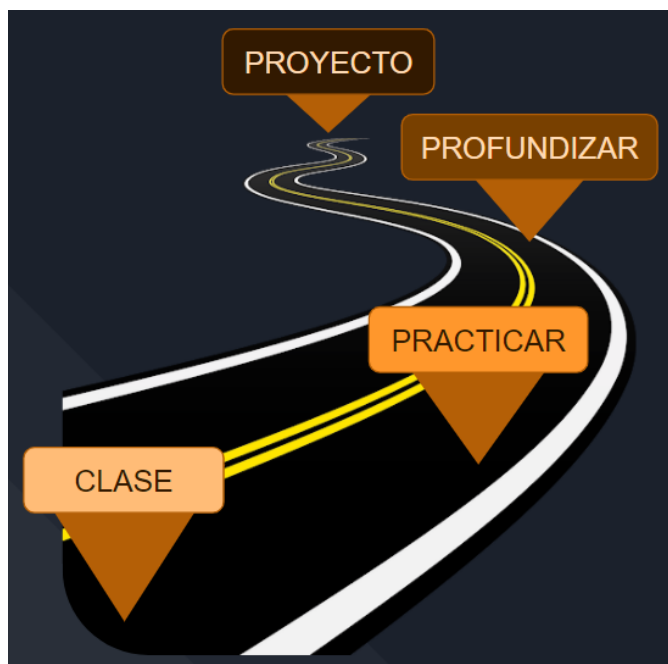
La ejercitación se divide en:

- **Ejemplos de clase:** Estos son los ejercicios y código que el docente mostrará y explicará durante la clase para poner a prueba los conceptos teóricos vistos.
- **Ejercicios de práctica:** Estos son los ejercicios que podrá realizar el alumno para poner a prueba sus conocimientos de lo visto en clase.
- **Ejercicios de profundización:** Para aquellos que posean el tiempo y el deseo de profundizar en los temas vistos en clase, recomendamos realizar los ejercicios de profundización.
- **Proyecto:** El proyecto es un ejercicio final que se realiza al final de la cursada.

La realización de los ejercicios es un camino que transita el alumno, la prioridad es que pueda realizar los ejercicios de práctica ya que son aquellos que garantizan haber entendido el concepto de lo visto en clase.

Los ejercicios de profundización se deben realizar cuando el alumno no tiene ninguna duda sobre lo realizado en los ejercicios de práctica, ya que son ejercicios de mayor dificultad. Recomendamos siempre priorizar la realización de los ejercicios de clase y haber comprendido la unidad antes de continuar con los ejercicios de profundización.

La realización del proyecto es la recta final del camino transitado por el alumno, recomendamos haber realizado y comprendido a la perfección los ejercicios de práctica **y los ejercicios de profundización** antes de comenzar a desarrollar el proyecto, ya que integra todo lo desarrollado durante la cursada.

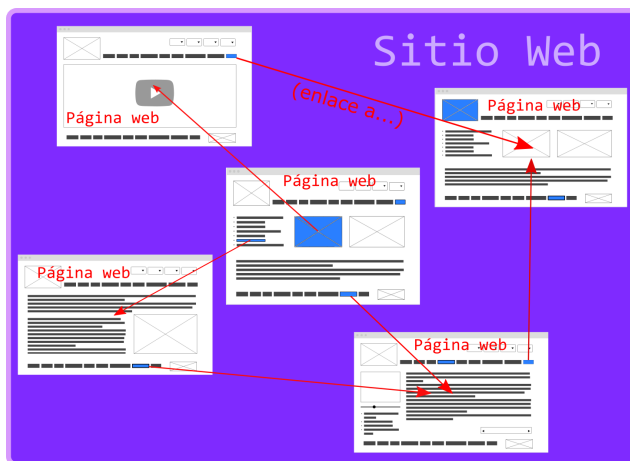


Introducción

¿Qué es una página web?

Una página web, es un documento electrónico (o varios) que están diseñados para ser interpretados por los navegadores para presentar distintos tipos de información (fotos, texto, videos, sonido, etc;), además tienen la capacidad de presentar vínculos con otras páginas web o direcciones mediante URLs (localizador de recurso uniforme) y de procesar información para ser presentada al usuario.

Las páginas web suelen tener el formato de un hipertexto, y un sitio web es una colección de páginas web, que están bajo el mismo dominio, así pues, se suele encontrar el dominio, por ejemplo inove.com.ar siendo la página principal del sitio, con enlaces a las otras:



A las páginas secundarias se les accede desde el link a su dirección que suele ser www.eldominio.com/el_nombre_de_la_pagina_secundaria.html (La barra indica que la página es secundaria) pero también podemos escribir su dirección en el buscador.

¿Quiénes se encargan del diseño web?

En las empresas dedicadas a la producción y mantenimiento de páginas web, se suele contar con un equipo de personas para este fin. Cada integrante se especializa en una tarea puntual, que involucra una serie de conocimientos:

- **Diseñadores UX** [User Experience] generan la estructura de la página web y del sitio en una serie de "wireframes" que hacen de modelo del sitio:



[Wireframe]

- **Diseñadores gráficos:** Toman estos Wireframes y generan el contenido gráfico, definen el estilo de la página.
- **Frontend developer:** Maquetador de HTML5/CSS3, son quienes toman el trabajo del diseñador gráfico, y lo llevan a un documento electrónico de hipertexto, con estilo, pero sin funcionalidades.
- **Backend developer:** Son los que les dan funcionalidad a la página por medio de distintos lenguajes, como javascript, php, manejan la relación con las bases de datos, para poder hacer que la maqueta realice tareas.
- **Desarrollador SEO** (Search Engine Optimization): se encarga de optimizar el código escrito para que los buscadores (como google) encuentren la página web y la posicionen por encima del resto de las demás.
- **Project Manager:** es el encargado de gestionar todo el proyecto y articular entre las distintas partes para cumplir con un cronograma definido, además, debe interactuar entre su equipo y el cliente para llevar el producto a lo que el cliente busca, dentro del marco del contrato.

El desarrollador web abarca los conocimientos relacionados con el frontend y el backend, pudiendo iniciar el proyecto desde cero, hasta tener una página web funcional, suficiente para realizar pequeños proyectos, debido a su baja complejidad. Para proyectos de gran escala, es necesario de la participación del resto de los actores para obtener un resultado de alta calidad en un tiempo adecuado.

Desarrollo Web

Dijimos que las páginas web, son documentos electrónicos que interpretan los navegadores para poder presentar la información en pantalla, estos documentos (de ahora en adelante "archivos"), contienen la información bajo distintas codificaciones, es decir, en distintos lenguajes.

Así como un libro puede estar escrito en castellano o en inglés, para que un determinado grupo de personas pueda entenderlo, los archivos electrónicos también pueden estar escritos en distintos lenguajes, y cada uno de ellos con propósitos distintos.

La extensión de un archivo es la que nos indica a qué tipo de codificación corresponde:



Extensión

lista de compras.pdf

Nombre de archivo

Algunas extensiones conocidas son:

- mp3 (archivo de sonido comprimido)
- jpeg (archivo de imagen)
- txt (archivo de texto)

Los navegadores web tienen la capacidad de abrir distintos tipos de archivos, entre ellos los anteriores descriptos, pero están optimizados el tipo de archivos de extensión HTML entre otros.

Los archivos html han ido evolucionando, y hoy por hoy contamos se encuentra en su versión 5, siendo **HTML5**.

HTML5 (Contenido y estructura)

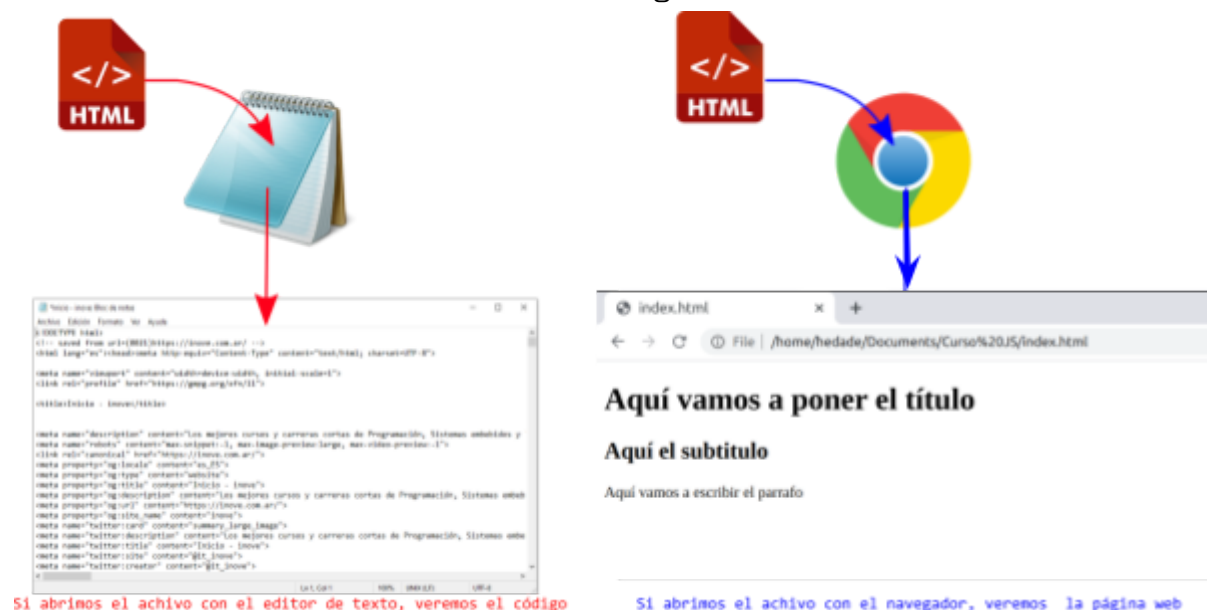
HTML (HyperText Markup Language) es un tipo de lenguaje de etiquetas (tags), que interpretan los navegadores para poder estructurar la información en la pantalla. Dicho lenguaje es un estándar utilizado mundialmente.

Pero, ¿qué significa que sea un lenguaje de etiquetas?

Supongamos que se trata de un libro, en donde tenemos por un lado el índice, por otro lado los distintos capítulos, el prefacio, etc; cada uno de estos nombres corresponde a una "etiqueta", que le indica al lector, que sector del libro estamos viendo...

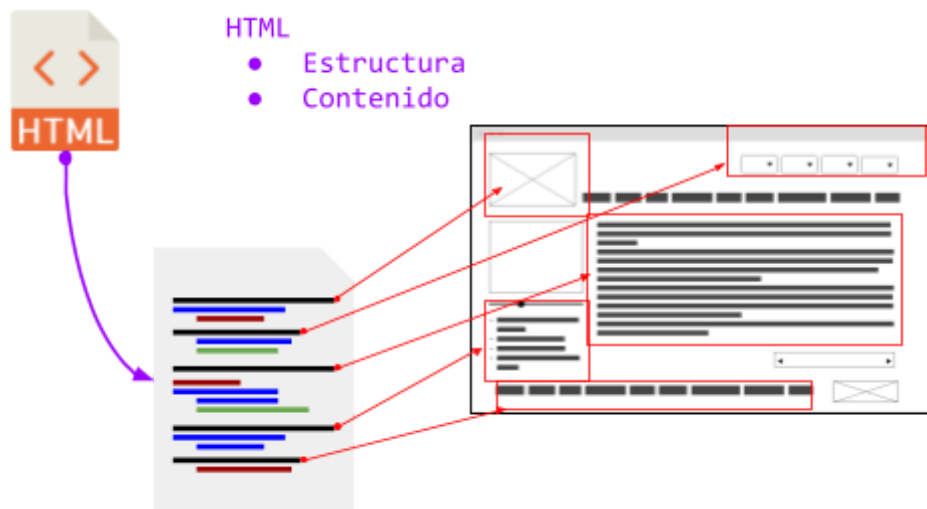
Así funciona el lenguaje HTML en principio, se trata de una serie de indicadores (llamados etiquetas) que proporcionan información para que interpreten los navegadores y puedan traducirla y representarla en pantalla.

A la información contenida en el archivo la llamamos código y puede ser visto y modificado con un visor de texto simple, así pues si a un archivo de extensión html lo abrimos con un navegador, veremos su página web, mientras que si al mismo archivo lo abrimos con un editor de texto veremos su código.



La tendencia actual es que los archivos HTML se utilicen exclusivamente para dotar a la página web de contenido y de estructura, nada más, el resto (estilo y funcionalidades) queda a cargo de otros lenguajes, aunque HTML soporta el manejo de varias funcionalidades más.

Entonces y en resumen:

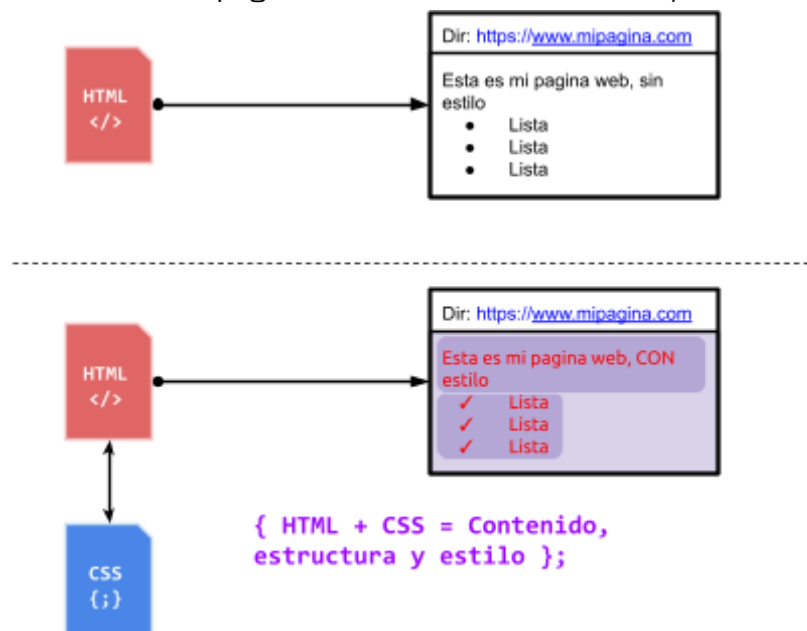


CSS3 (estilo)

CSS corresponde a las siglas "Cascading Style Sheets" o en español, "Hojas de estilo en cascada". Es un lenguaje que sirve para dotar a nuestra página de **estilo**.

El estilo en una página web tiene que ver con: colores, tipografía, forma de representación de los bloques HTML, comportamiento de los mismos, etc; todo lo relativo a la parte gráfica del sitio.

Los archivos .css se vinculan a los .html, es decir, estos archivos "se relacionan entre sí" para dar como resultado una página con estructura, contenido y estilo.



Javascript para el diseño web

A grandes rasgos, podemos decir que Javascript es un lenguaje de programación que se utiliza para proporcionar funcionalidades a las páginas web.

Dentro de estas funcionalidades se encuentran, por ejemplo:

- Chats
- Reproductores multimedia
- Juegos
- Interacción, etc;

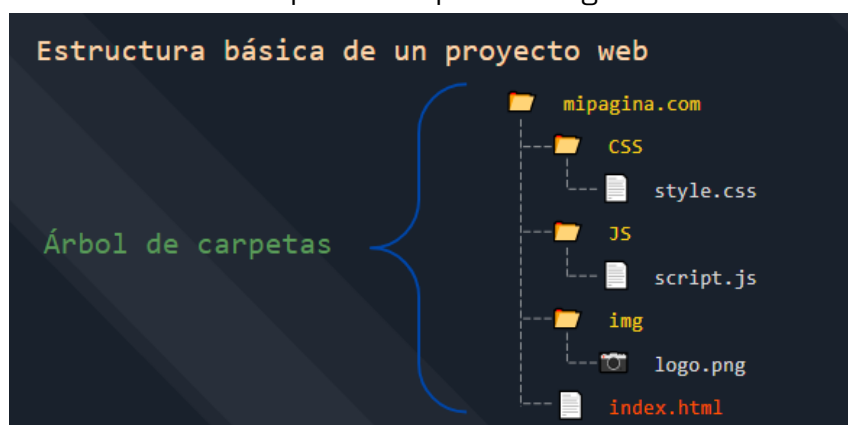


Aunque se puede editar una página web con un editor de texto básico, existen herramientas concebidas específicamente para este fin, y están pensadas para brindarle mayor practicidad a la hora de realizar el trabajo

Estructura básica de un proyecto WEB

Un proyecto web, una página o sitio, involucra tantos archivos y recursos como sean necesarios.

En una primera instancia, una página web solo necesita de un archivo html, que contenga toda la información, pero para nuestro estudio en esta etapa, comenzaremos por definir una estructura básica que se compone del siguiente árbol de carpetas:



La misma se compone de una carpeta raíz, con el nombre del sitio, que contiene un archivo al que llamaremos "index.html", 3 carpetas con recursos: CSS, JS, img. Cada una de ellas con los archivos de código e imágenes correspondientes.

Esta estructura será la que vamos a mantener en el transcurso de esta etapa, lo que haremos será editar estos archivos y agregar nuevos.

Preparando el entorno: Herramientas necesarias

Para comenzar, debemos contar con una de las herramientas fundamentales para todo desarrollador, un editor de código.

Editor de código

Para nuestro trabajo utilizaremos el software Visual Studio Code (o simplemente VS Code) es un **"editor de código optimizado"** según sus creadores, desarrollado por la empresa Microsoft, y que se ha posicionado como una de las plataformas de trabajo para el desarrollo de software más utilizadas, debido a que se trata de un software libre, gratuito y con una gran comunidad, lo que lleva tranquilidad a sus usuarios debido a la constante actualización para la corrección de errores en el programa, y desarrollo constante de extensiones para expandir sus capacidades.

Descarga e instalación:

Para la descarga e instalación, por favor referirse al documento "Instructivo de instalación: Visual Studio Code" disponible entre los archivos del campus. De no encontrarse allí, por favor, tenga a bien comunicarse con alumnos@inove.com.ar para su solicitud.

Por el momento basta decir que este software va a ser nuestro aliado para poder editar el código, en las sucesivas unidades vamos a conocerlo en profundidad para poder sacarle el mayor provecho.

Un poco de historia (contenido extra)

¿Qué es internet?

Internet viene de "red internacional" surge en el año 1979 como idea para cubrir la necesidad de conectar dispositivos dispersos geográficamente "más allá de las fronteras de un país" para que puedan compartir información, o tareas .

Sin embargo, la idea precursora de lo que hoy conocemos como una "página web" fue concebida años atrás, entre 1945 y 1965, por el científico Vannevar Bush y posteriormente por Theodore Holm Nelson, éste último es quién acuñó el término **hipertexto**: un documento que puede contener imágenes, texto, etc; ordenado de manera no secuencial, y que podamos acceder por medio de conexiones (links), que nos lleven a otros documentos a su vez, con las mismas capacidades.



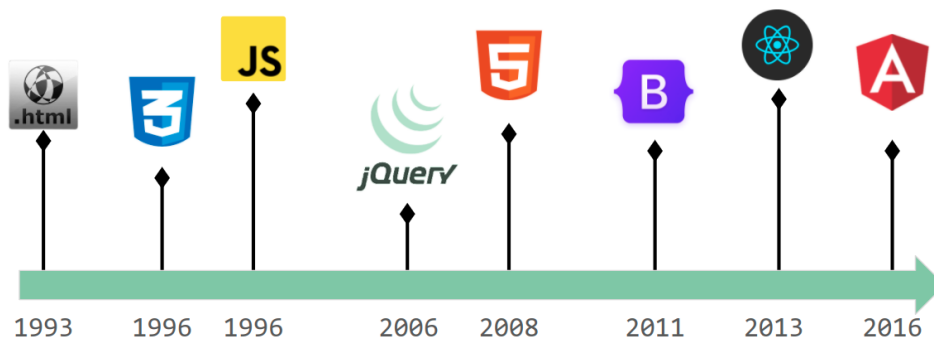
Sin embargo, en esa época, era imposible de realizar técnicamente.

En 1989, Tim Berners-Lee, planteó la idea de cómo hacer que las conexiones sean sencillas y homogéneas: la World Wide Web (abreviada como WWW), que significa "red(o telaraña) global" a partir de allí, la web fue evolucionando, y pasó por:

- **WEB 1.0:** era un lugar virtual donde se depositaba información contenida en códigos difícilmente manipulables y actualizable, en la cual no existía la interacción ya que era unidireccional
- **WEB 2.0:** se facilitan las interacciones y se estandarizan los lenguajes para una mejor reutilización del código
- **WEB 3.0:** evoluciona hacia una web semántica, en donde la inteligencia humana y la de las máquinas se comunican, creando un lenguaje más natural propiciado por algoritmos genéticos.
- **WEB 4.0:** suma la cualidad de ubicuidad, red móvil, usabilidad, accesibilidad, una web indexable y actualizable, una web cerebral.

La evolución de los lenguajes web

A la vez que la web iba evolucionando también lo hicieron los lenguajes que se utilizan para crearla y sus herramientas:



A comienzos de los 90' nace HTML y unos años más tarde CSS y Javascript. Este trío inseparable son los responsables de todas las webs que conocemos desde finales de los 90' hasta el día de hoy.

En un comienzo había muchas dificultades para desarrollar páginas que se vieran bien en cualquier explorador web, ya que no estaba estandarizado del todo cómo se interpretaban diferentes funcionalidades de estos lenguajes.

En 2006 y 2008 nacen dos grandes, JQuery y HTML5. HTML5 es el estándar de HTML que todas las páginas de hoy usan, y JQuery es una de las librerías de Javascript más populares cuyo misión fue estandarizar código, hacerlo más simple y limpio y facilitar a los desarrolladores programar en Javascript eventos y acciones responsivas.

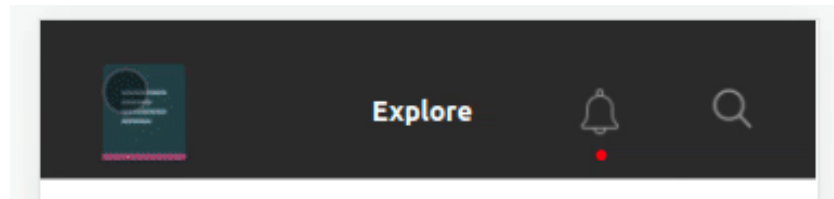
JQuery ayudó mucho a los desarrolladores Web, pero aún faltaba una librería o framework que ayudaría con el manejo de estilos (CSS). El manejo de estilos llevaba bastante tiempo y dedicación por lo que era difícil poder desarrollar una página rápida a modo de concepto. Por este motivo nacen frameworks de librerías de estilos como Bootstrap a comienzos de la década del 2010, simplificando el estilado de las páginas que elementos o componentes ya armados para usarse.

Llegando al año 2015 nacen los frameworks de Javascript que vienen a darle una evolución a lo que hizo JQuery por el lenguaje, estandarizar Javascript y darle un enfoque moderno, simple y rápido que permitiera tener páginas optimizadas para correr en dispositivos móviles y sean tan rápidas y responsivas como se las puede ver en un equipo de escritorio. Ayudaron a organizar el código y permitir que el manejo de estilos se organice por componentes y pueda reutilizarse el código con mayor facilidad.

Dicho lo anterior, en la web van a encontrar ejemplos según la versión de javascript o según el set de herramientas que haya utilizado el desarrollador. En este curso nos vamos a enfocar en una opción de uso de los últimos frameworks de Javascript (React, Angular, Vue, etc).

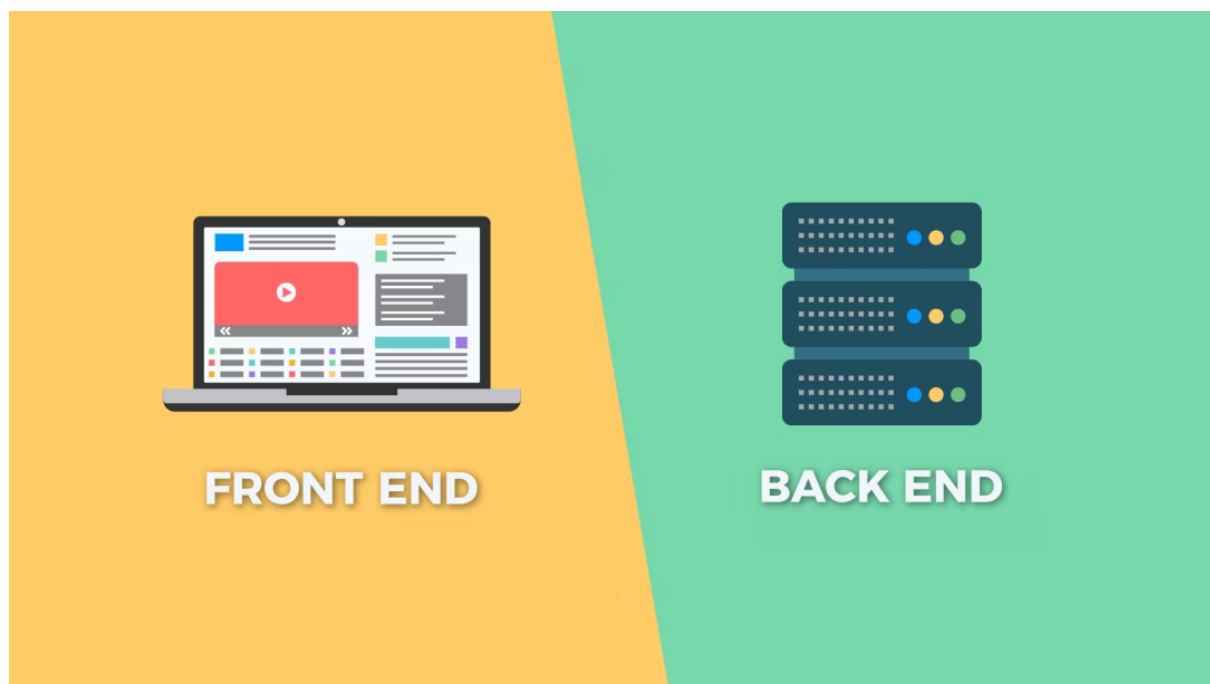
Backend vs Frontend

Se dice que es más fácil ser un desarrollador frontEnd porque los resultados que se obtienen son más tangible, ¡el resultado se puede ver!



También **se dice** que los desarrolladores backend tienen que usar más lógica y crear programas más complejos que una interfaz

¡PERO un desarrollador front requiere conocer más lenguajes! Por lo que puede resultar más sencillo al comienzo pero ¡puede llevar más práctica dominar el frontend y ser un maestro en ello!



Fullstack se lo llama al desarrollador que tiene conocimientos tanto del BackEnd como del FrontEnd. En la jerga se lo conoce como "Fullstack Ninja" o "Django Ninja" que tiene el entendimiento de todas las herramientas y lenguajes que confirman la aplicación.

Navegadores Web y dominios (extra)

Los navegadores son las herramientas que nos ayudan a encontrar en esta inmensa red el sitio (o página) que estamos buscando, cada sitio tiene un código único, una dirección, a la cual visitamos para ver su información.

Un dominio de internet es simplemente un alias que sirve para poder identificar una página específica, única y diferenciarla de las demás, este método trabaja con el "sistema de nombres de dominio" y lo que hace es traducir las direcciones IP de cada dispositivo conectado a la red, recordemos que estas direcciones IP suelen ser dinámicas, por ello, si bien una página se asocia a un dominio, pero no necesariamente debe estar asociada a una dirección IP estática.

Actualmente existen distintos navegadores de uso gratuito, que ofrecen además diversas herramientas, algunas gratuitas, otras no, que podemos utilizar para el desarrollo de nuestra propia página web.

En este curso, utilizaremos principalmente chrome de google, y mozilla firefox, pero también tendremos que comprobar el desempeño de nuestra página web en distintos navegadores y dispositivos.

Descarga del repositorio "hola_mundo_web"

Para poder realizar esta actividad deben haber primero instalado Git en sus computadores y haberse creado una cuenta de GitHub para poder trabajar. Para ello referirse al "Instructivo de instalación: Git & GitHub" disponible entre los archivos del campus. De no encontrarse allí, por favor, tenga a bien comunicarse con alumnos@inove.com.ar para su solicitud.

Completada la configuración del entorno, debemos descargar el repositorio "Hola Mundo!" del repositorio de Inove. El repositorio que descargamos del siguiente:

https://github.com/InoveAlumnos/hola_mundo_web

Recomendamos seguir el siguiente video tutorial para realizar la correcta descarga del material:

[Link al video de youtube](#)

Una vez descargado el repositorio como un archivo "zip" lo deben ubicar en una carpeta donde colocarán de aquí en más los repositorios de este curso y descomprimir los archivos para poder abrirlos en Visual Studio Code.

¡Hasta la próxima!

Con esto finaliza esta clase, a partir de ahora tienen todo preparado para empezar a realizar programas y continuar con la ejercitación de la clase.

Si desean conocer más detalles sobre el contenido pueden iniciar un tema de discusión en el lugar correspondiente, o visitar los "Links de interés" que se encuentra al final de este apunte.

Links de interés

- [Campus de Inove](#)