

«Talento Tech»

Desarrollo Web 2

Clase 01





Clase N° 1 | Conceptos básicos

TEMARIO:

- Repaso Git/Github
- Introducción a la programación
- Funcionamiento web
- ¿Qué es JavaScript?



Repaso de Git/Github

A medida que desarrollamos nuestro proyecto, nos suelen pasar que guardamos diferentes versiones del mismo. Esto podría ser un problema si vamos cambiando el nombre del proyecto continuamente para reconocer cuál fue la versión final.

Para esto vamos a conocer una herramienta que nos va a permitir guardar todos los cambios y tener un historial de versiones.

¿Qué es Git?

Git es un sistema de control de versiones, que nos ayuda a guardar diferentes versiones de nuestro proyecto. Si tenemos algún error podemos acceder a una de las versiones anteriores para volver a empezar, esto lo hacemos a través del historial de versiones que se va generando cada vez que se guarda una nueva versión.

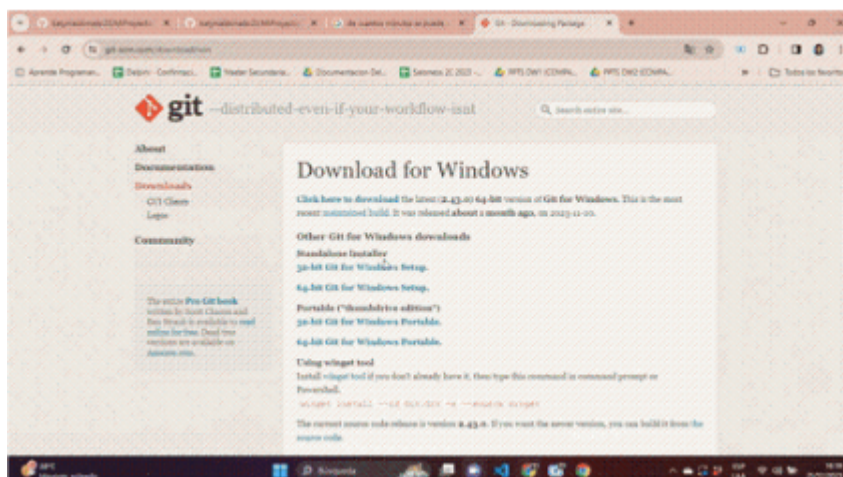
¿Qué es GitHub?

GitHub es una plataforma que te permite **guardar y gestionar proyectos**, es donde vamos a guardar nuestro código y **lleva un registro de los cambios que hacemos y trabajar con otros en el mismo proyecto sin problemas**.

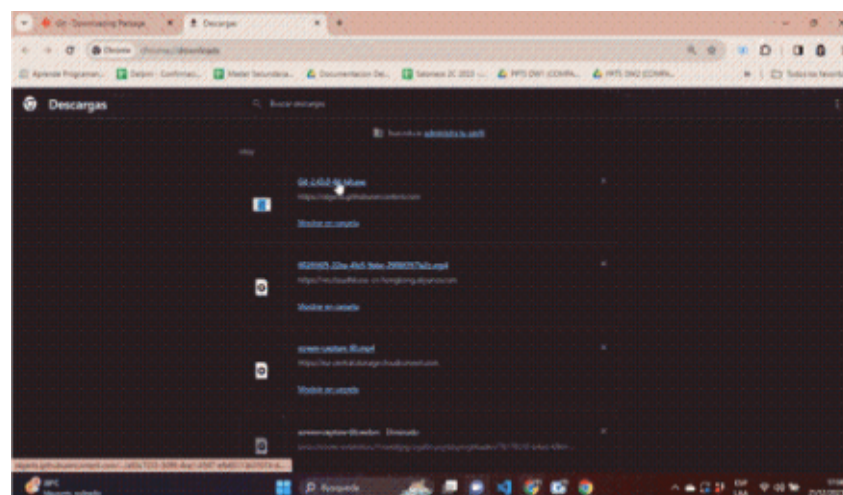
Es como una **red social para desarrolladores** donde se comparten ideas, se colabora en código y se lleva un control de versiones de los archivos.

! Importante: Git guarda cambios en archivos para que podamos seguir su evolución mientras que GitHub es un lugar en internet donde guardamos esos cambios y nos permite trabajar en equipo con otras personas.

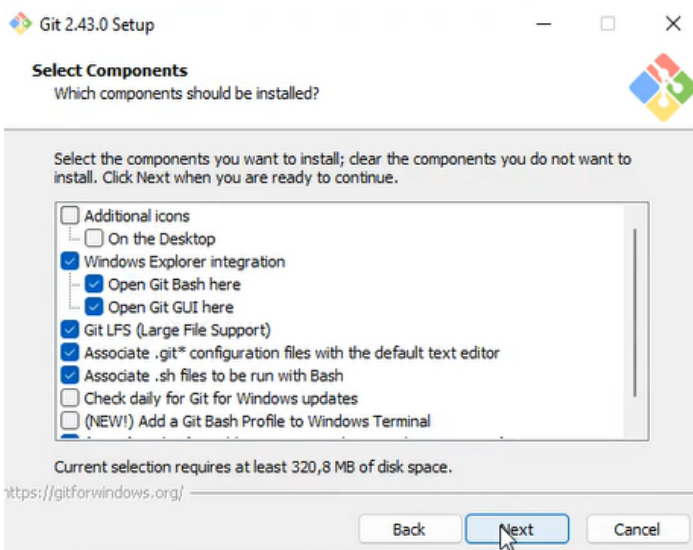
Instalar Git en Windows:



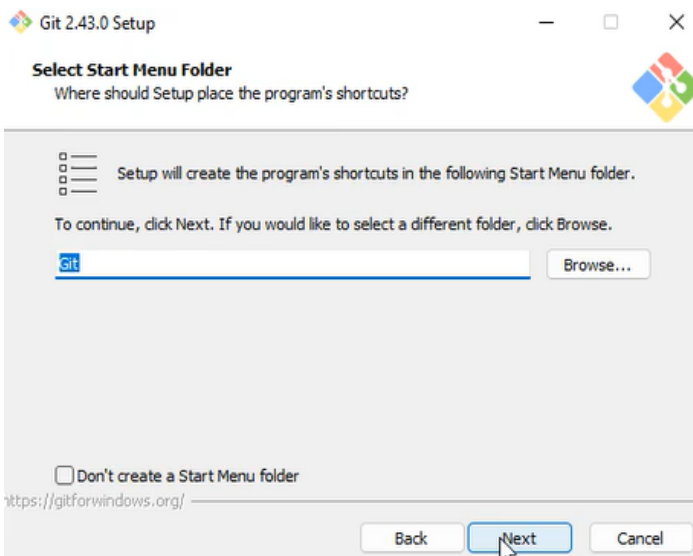
Descarga Git: Ve al sitio web oficial de Git (<https://git-scm.com/>) y haz clic en "**Descargar for Windows**". Luego descarga el instalador según los bits de tu Computadora:



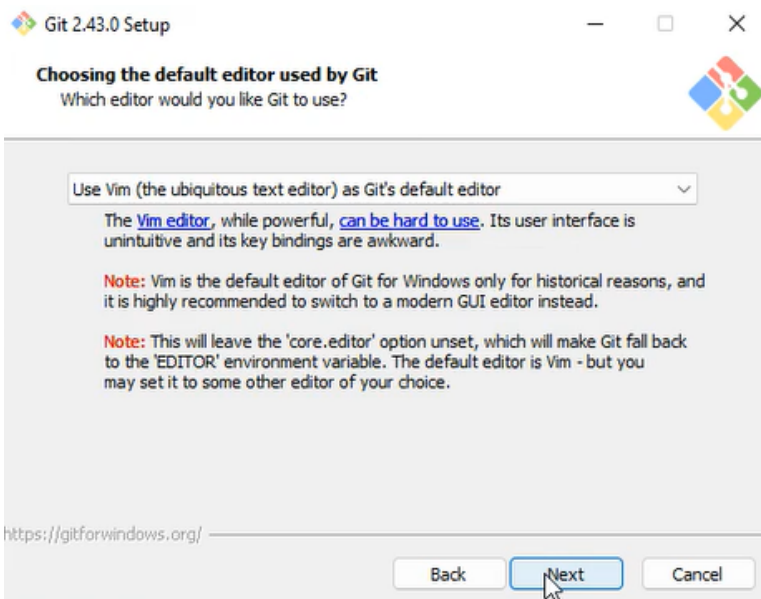
Ejecutar el instalador: Una vez descargado, haz doble clic en el archivo descargado para iniciar el instalador.



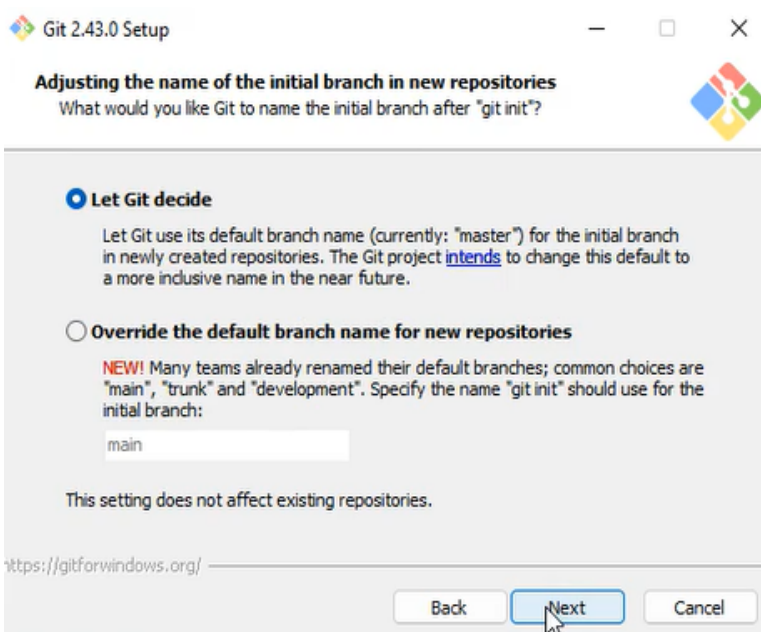
Configurar opciones: Cuando es momento de seleccionar componentes a instalar, es muy importante que hagamos click en estos que se ven en la imagen.



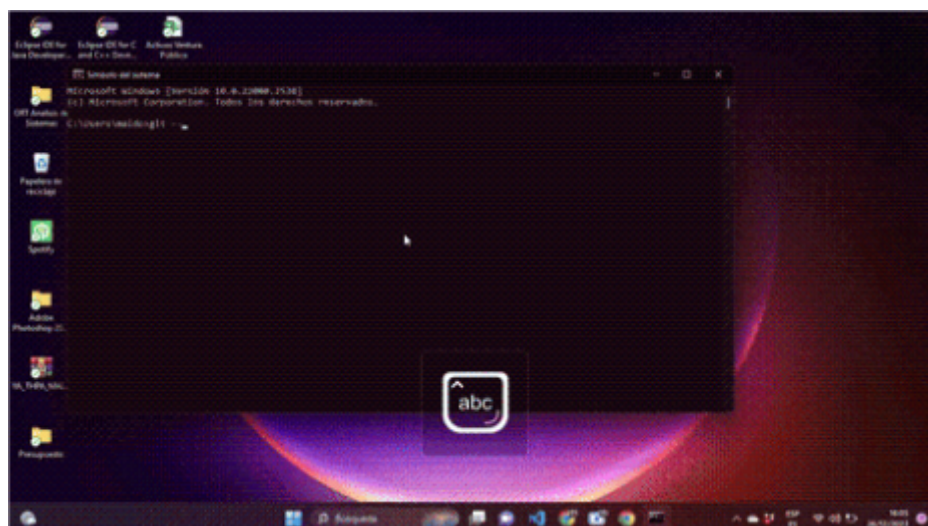
Se puede seleccionar la carpeta donde guardamos nuestro progreso. Podemos clickear en next, y se va a almacenar todo en una carpeta llamada **"Git"**. Allí tendremos todo nuestro trabajo.



Vim es el editor predeterminado que se usa en GIT para Windows. Cuando nos dé la opción de seleccionar, lo elegimos y hacemos click en next.



Finalizar la instalación:
Después de completar la instalación, verificar si se instaló correctamente abriendo la línea de comandos. Para eso, puedes buscar "cmd" en el menú de inicio y escribir git --version. Si se instaló correctamente, mostrará la versión de Git que acabas de instalar.



Crear tu perfil de GitHub

- **Ingresa a GitHub:** Abre tu navegador y ve a <https://github.com/>
- **Registrarse:** Busca y haz clic en el botón "**Sign up**" o "**Registrarse**". Está en la esquina superior derecha de la página.
- **Completa la información:** Ingresa tu nombre de usuario, una dirección de correo electrónico válida y una contraseña segura. Luego, haz clic en "**Create an account**" o "**Crear cuenta**".
- **Verificación de correo:** GitHub enviará un correo electrónico de verificación a la dirección que proporcionaste. Revisa tu bandeja de entrada y sigue las instrucciones para verificar tu cuenta.
- **Configuración de perfil:** Una vez que hayas verificado tu cuenta, puedes personalizar tu perfil agregando una imagen de perfil y completando la información de tu perfil si lo deseas. Haz clic en tu avatar en la esquina superior derecha y selecciona "**Your profile**" o "**Tu perfil**".



Con el perfil en GitHub creado, podemos explorar repositorios, colaborar en proyectos o incluso crear tus propios proyectos y subirlos a tu perfil.

Insertar video de instalación Github

 [Como subir un Proyecto a GitHub desde Visual Studio \(Video\)](#)

Comandos Básicos de Git

git init	Inicializa un nuevo repositorio Git en el directorio actual.
git config	Establece el nombre de usuario y dirección de correo electrónico. Esto es importante porque los "commits" de Git usan esta información. Por ej: git config --global user.name "USER" git config --global user.mail MAIL
git clone URL	Clona un repositorio Git existente en tu máquina local desde la URL proporcionada.
git status	Muestra el estado actual de los archivos en tu repositorio
git add *	Agrega todos los cambios realizados para ser incluidos en el próximo commit.
git commit -m "mensaje"	Confirma los cambios que has agregado con un mensaje descriptivo de los cambios realizados .
git push	Sube tus commits locales a un repositorio remoto y a una rama específica . Por ej: git push origin main .
git pull	Trae los cambios del repositorio remoto a tu repositorio local. Por ej: git pull origin main .
git branch	Muestra una lista de ramas en tu repositorio. Agregar -a mostrará todas las ramas, incluyendo las remotas.
git checkout [nombre-rama]	Cambia a una rama específica Por ej: git checkout develop cambiará a la rama llamada "develop".
git merge [nombre-rama]	Fusiona una rama específica en la rama actual. Por ej: Si estás en la rama "main" y quieres fusionar los cambios de "develop", usa git merge

Introducción a la programación

En el curso de **Desarrollo Web 1** investigamos todo acerca de **HTML y CSS**, los cuales nos permitieron poder crear nuestro sitio web. Ahora, ambos lenguajes tienen una característica en particular: ninguno de los dos se considera un lenguaje de programación.

🤔 ¿Por qué no son considerados como lenguajes de programación?

Para ello es muy importante tener el concepto de que es la programación en claro.

La programación es el proceso de crear secuencias de instrucciones para que una computadora realice una tarea específica. Es un conjunto de reglas y técnicas utilizadas para diseñar y escribir código en lenguajes de programación, que luego se traducen en instrucciones comprensibles por la máquina.



La programación implica el análisis del problema, la planificación de la solución, la escritura de código, la depuración y la optimización del programa resultante. Permite a los desarrolladores crear software, aplicaciones y sistemas que automatizan tareas, resuelven problemas o mejoran la eficiencia en diversas áreas de la vida y los negocios.

Los **lenguajes de programación** nos permiten usar variables, funciones, ciclos y condicionales, con el objetivo de desarrollar un **Algoritmo**.

Nos referimos al concepto de algoritmo cuando hablamos de una secuencia de pasos o instrucciones que buscan resolver un problema específico.

Este problema a resolver puede ser tan cotidiano como, por ejemplo, lavarnos los dientes. Si estamos en casa y deseamos lavarnos los dientes, debemos seguir la siguiente serie de pasos (algoritmo).

Algoritmo para lavarnos los dientes

- 1- Ir hasta el baño
- 2- Tomar el cepillo de dientes
- 3- Tomar la pasta de dientes
- 4- Colocar la pasta en el cepillo
- 5- Cepillarse los dientes
- 6- Abrir la canilla
- 7- Enjuagarse la boca
- 8- Guardar el cepillo en su lugar

Los algoritmos se pueden expresar con diagramas de flujo o trabajando con lenguajes de programación. Se utilizan en muchas disciplinas, como la ingeniería o la matemática, pero son **trascendentales** en la programación, ya que mediante el uso de algoritmos se logra desarrollar software.

Funcionamiento de la Web

El funcionamiento de la web se divide en dos partes principales para entender mejor cómo funciona:

FrontEnd: es la parte visible y con la que interactúas cuando navegas por internet. Esto sirve para crear una experiencia de usuario atractiva y funcional. HTML proporciona la estructura, CSS ofrece el estilo y diseño, y JavaScript añade la interactividad, permitiendo que las páginas web sean más que solo información estática.

BackEnd: es la parte "oculta" o detrás de escena de una página web. Aunque no la vemos directamente, desempeña un papel importante en el funcionamiento de todo el sistema web. Se encarga de procesar datos, gestionar usuarios, realizar cálculos y acceder y manipular la base de datos.

Trabaja en conjunto con el frontend para proporcionar una experiencia completa y funcional al usuario en la web. Sin el backend, los sitios web no podrían almacenar información, manejar solicitudes o realizar acciones complejas.

Ejemplo del funcionamiento de un Restaurante con FrontEnd y BackEnd:



Para un completo funcionamiento de la web se necesita de otros factores que también son igual de importantes, ya que son los que se encargan de mostrarnos las páginas en nuestras computadoras.

Navegadores

También conocido como navegador web o browser, tienen la capacidad de leer sólo tres lenguajes: HTML, CSS y JavaScript. Permite a los usuarios acceder y visualizar páginas web en Internet. Los navegadores son como intermediarios entre los usuarios y los servidores web.

Servidores

Un servidor es una computadora o un sistema que almacena y proporciona acceso a los archivos y recursos necesarios para que los sitios web funcionen correctamente.



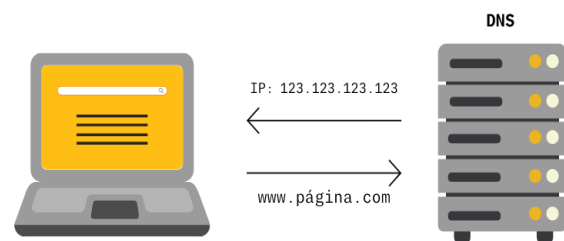
Proceso del Funcionamiento de la Web

Cuando se ingresa una dirección web en la barra de direcciones del navegador, realizamos una solicitud al servidor correspondiente, para obtener los archivos que componen tal sitio web, como **HTML, CSS, JavaScript, imágenes y otros recursos**. Luego, el servidor procesa nuestra solicitud y nos devuelve los archivos necesarios. Por último, el navegador recibe los archivos necesarios para que podamos ver e interactuar con el sitio web. Cuando accedemos a un sitio web, que luego se muestran en el navegador.

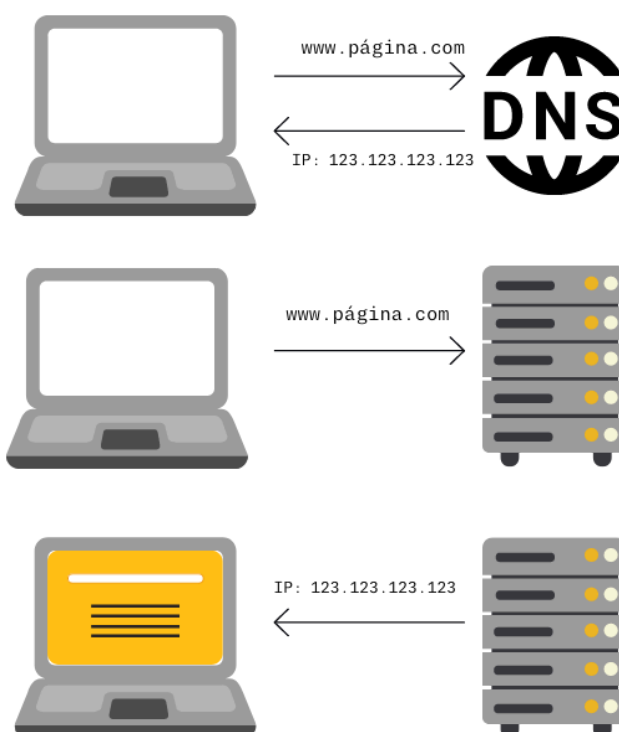
DNS

El **DNS (Domain Name System)** es un sistema que se utiliza en Internet para traducir los nombres de dominio de las páginas web en direcciones **IP** numéricas. Cada dispositivo conectado a Internet, como un servidor web o un cliente (como nuestra computadora o teléfono), tiene una dirección **IP** única que se utiliza para identificarlo en la red.

Cuando escribimos una dirección web en el navegador (como `www.ejemplo.com`), el **DNS** traduce ese nombre de dominio en la dirección IP numérica del servidor donde se encuentra en esa página web.



Esquema del Funcionamiento Web



En conclusión, el funcionamiento Web es la comunicación entre el navegador y los servidores que almacenan la información de las páginas. Cuando escribimos una dirección web, el navegador busca la dirección del servidor a través del **DNS**, envía una solicitud a ese servidor pidiendo la información de la página y recibe archivos como HTML, imágenes y estilos.

El navegador ensambla esos archivos para mostrarnos la página web que solicitamos, permitiéndonos interactuar con ella y acceder a su contenido

¿Qué es JavaScript?

JavaScript es un lenguaje de programación utilizado para generar mayor dinámica e interactividad a nuestra página web. Su característica principal es que nos permite interactuar con el contenido de las páginas, veamos algunos ejemplos:

- **Validar formularios:** Gracias a JavaScript, podemos validar los datos que ingresamos en el formulario que creamos con HTML.
- **Contenido dinámico:** JavaScript nos permite implementar carruseles, menús desplegables y más.

Se implementa la siguiente lógica algorítmica:

“Si aprieto este botón (sea del carrusel o del menú desplegable), se generará una modificación estética de la página”. Esto ocurre en carruseles, cuando presionamos el botón para avanzar a la siguiente imagen, y también en los menú desplegables, cuando apretamos en la barra y se despliega un menú de opciones.

Para implementar JavaScript vamos a empezar trabajando con la consola de nuestro navegador web. La consola es una herramienta fundamental para el desarrollo de código de JavaScript, ya que nos permite probarlo y ejecutar código muy fácilmente, y visualizar los errores que pueden aparecer en el proceso.

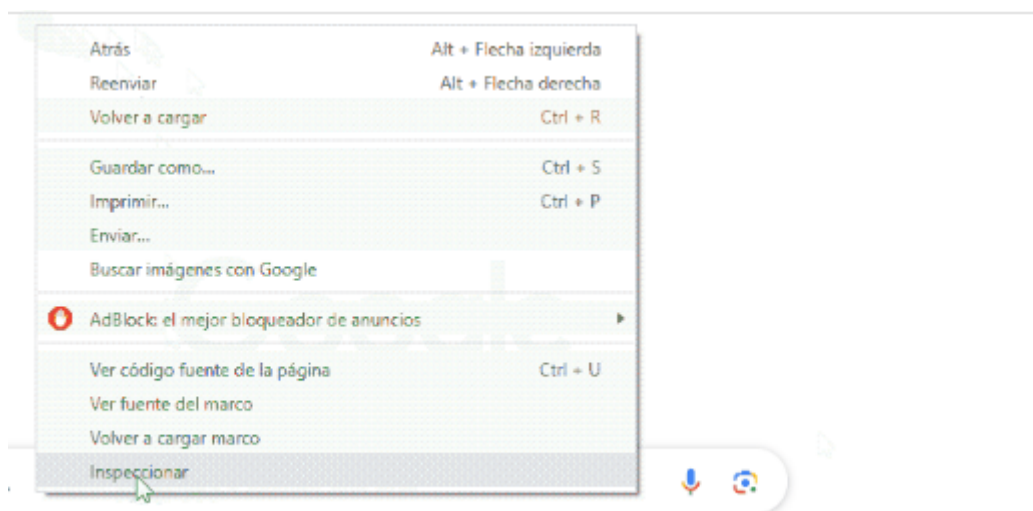
Consola JavaScript en el navegador

La consola en los navegadores web es una herramienta fundamental para los programadores. Ayuda a encontrar errores en el código de **JavaScript**, permite probar nuevas ideas rápidamente y muestra datos útiles para hacer que las aplicaciones web funcionen mejor y más rápido.

Para acceder a la consola:

- Vamos a nuestro navegador web.
- Apretamos click derecho en la página -> **Inspeccionar** -> **Consola**

Y desde allí podremos empezar a desarrollar código:



? Pregunta de reflexión o de llamado de atención

Después de repasar los conceptos de **Git/GitHub**, la instalación de **Git en Windows**, la creación de un perfil en **GitHub**, los comandos básicos de **Git**, y la introducción a la programación y el funcionamiento de la web, así como la importancia de JavaScript en la dinámica de las páginas web:

- ¿Se te ocurre alguna aplicación práctica para estos conocimientos un E-Commerce digital?
- ¿De qué modo influye Javascript en nuestra interacción con las páginas web?



Buenos Aires
aprende
Agencia de Actividades para el Futuro

BA Buenos
Aires
Ciudad