

Aspectos principales del desarrollo en web

M: Fundamentos de desarrollo Front-End

| **AE1:** Reconocer los principales aspectos relacionados con el desarrollo web distinguiendo el rol y elementos fundamentales de Front-End

Introducción

El **desarrollo web** es un campo en constante evolución que se enfoca en la creación de sitios web y aplicaciones web. **Implica el diseño, la implementación y el mantenimiento de sitios web** utilizando tecnologías y lenguajes de programación específicos.

Cuando se diseña una página web **se planifica pensando en cumplir un objetivo para dar a conocer un producto y/o servicio**, puede ser para vender o solo informar, pero siempre es tomando datos y haciendo llegar información a un usuario final de una forma interactiva, amena y fácil para cualquiera que la consulte.

En el **desarrollo web**, es esencial comprender los tres componentes principales: **el cliente, el servidor y la base de datos**. El cliente es la interfaz con la que los usuarios interactúan, generalmente a través de un navegador web. El servidor es responsable de procesar las solicitudes del cliente y enviar las respuestas adecuadas. La base de datos es donde se almacena y gestiona la información del sitio web.

Aprendizaje esperado

Cuando finalices esta lección podrás:

- Comprender los fundamentos del desarrollo web.
- Diferenciar entre Front-End, Back-End y Fullstack.
- Conocer los componentes que componen a una web.
- Diferencia entre Web estática y dinámica.
- Familiarizarse con las herramientas y tecnologías que se usan en el desarrollo web.
- Conocer qué es HTML, CSS y Javascript para crear páginas web.

El desarrollo Web

Para el diseño y desarrollo web existen diferentes lenguajes que nos permiten llevar el **diseño en papel (Sketch)** a una estructura que pueda interpretar un computador.

El desarrollo web es un **proceso de crear, construir y mantener sitios web y aplicaciones web**. Para poder llevar a cabo este proceso se utilizan tecnologías, lenguajes de programación y herramientas para diseñar, desarrollar y desplegar sitios web funcionales y atractivos.

El desarrollo web **implica una amplia gama de tecnologías, lenguajes y conceptos**. Para esto, el trabajo se divide en roles específicos permitiendo a los profesionales especializarse en áreas particulares y desarrollar experiencia en ese campo. Los desarrolladores Front-End se enfocan en la interfaz de usuario y la experiencia del usuario, mientras que los desarrolladores Back-End se especializan en la lógica del servidor y la gestión de datos. Esta especialización les permite perfeccionar sus habilidades y conocimientos en áreas específicas. Al tener roles especializados, cada aspecto del desarrollo puede recibir atención y cuidado adecuados. Esto conduce a un producto final de mejor calidad y facilita el mantenimiento a largo plazo de la aplicación web o el sitio web.

Qué se entiende por desarrollo web

El desarrollo de web abarca varias áreas:

- **Diseño web:** es toda la parte de creación de la estructura, el diseño visual y la experiencia del usuario.
- **Programación web:** involucra la escritura de código para construir la página/sitio web. Para la estructura y la estética, se utilizan lenguajes de marcado como HTML y CSS, y para programar la interactividad, la gestión de datos y operaciones del lado del servidor se utilizan lenguajes de programación como Javascript, PHP, Python, entre otros.
- **Desarrollo front-end:** se enfoca en la parte del desarrollo web que tiene que ver con la implementación de la interfaz de usuario y la interacción de los usuarios con un sitio web o una aplicación web. Los desarrolladores front-end trabajan con tecnologías como HTML,

CSS y JavaScript para crear la parte visual y funcional de un sitio web.

- **Desarrollo back-end:** se ocupa de la lógica y el funcionamiento detrás de escena de un sitio web o una aplicación web. Los desarrolladores back-end trabajan en la parte del servidor y se centran en la gestión de datos, el procesamiento de solicitudes y la implementación de la lógica de negocio.
- **Bases de datos:** las aplicaciones web suelen requerir el almacenamiento y recuperación de datos. Los desarrolladores web utilizan sistemas de gestión de bases de datos (SGBD) para diseñar y administrar la estructura de datos, realizar consultas y garantizar la integridad de la información.
- **Despliegue y mantenimiento:** una vez que el sitio web o aplicación web está desarrollado, debe ser implementado en un servidor para que esté accesible en Internet. Los desarrolladores web se encargan de la configuración del servidor, el despliegue de los archivos y la gestión del mantenimiento y las actualizaciones del sitio web.

Diferencias entre Front-End, Back-End y Fullstack

Front-End, Back-End y Fullstack son términos que se utilizan para describir diferentes roles y enfoques dentro del desarrollo web. **Cada uno de estos roles se enfoca en aspectos específicos del proceso de desarrollo y requiere habilidades y conocimientos especializados.** Algunos desarrolladores pueden especializarse en un área específica, mientras que otros pueden tener habilidades en ambos y desempeñar roles Fullstack.

La separación entre el desarrollo Front-End y Back-End existe debido a la naturaleza y las necesidades del desarrollo web, esto simplifica a la hora de crear y desarrollar un sitio o aplicación web.

Funcionalidades

Front-End

El Front-End se ocupa de todo lo que los usuarios ven y con lo que interactúan en un sitio web o aplicación. Esto incluye la estructura y el diseño de la página, la

presentación visual utilizando HTML, CSS y JavaScript, la interacción con los elementos de la interfaz, y la optimización de la experiencia del usuario.

Back-end

El Back-End se encarga de las funcionalidades y procesos detrás de la escena que permiten que un sitio web o aplicación funcione correctamente. Esto incluye la gestión de la base de datos, la comunicación con otros sistemas o servicios, el procesamiento de datos, la seguridad, la autenticación y la lógica del servidor.

Roles

Front-end

Los **desarrolladores Front-End** se ocupan de la implementación del diseño, la estructura y la experiencia del usuario en el navegador. Utilizan tecnologías como HTML, CSS y JavaScript para crear y dar estilo a la interfaz de usuario, gestionar la interactividad y mejorar la usabilidad. Su objetivo principal es lograr una interfaz atractiva y fácil de usar para los usuarios finales.

Además de implementar la UI, el Front-End es responsable de la **experiencia de usuario (UX)** en el navegador, de **coordinar con Back-End la integración de APIs/servicios** (definiendo contratos y manejo de errores/estados) y de **garantizar accesibilidad, compatibilidad y rendimiento** (p. ej., criterios WCAG, soporte multi-dispositivo y optimización de carga). En el **proceso de desarrollo**, participa desde diseño (componentes y criterios de accesibilidad) hasta pruebas y despliegue (optimización y monitoreo del cliente).

Back-end

Los **desarrolladores Back-End** se encargan de construir y mantener la lógica del servidor, la base de datos y otros componentes que permiten el funcionamiento de una aplicación web. Utilizan lenguajes de programación como PHP, Python, Ruby, Java, entre otros, y se preocupan por la seguridad, el rendimiento, la escalabilidad y la integridad de los datos.

Su enfoque principal es la implementación de la funcionalidad y la manipulación de los datos en el servidor.

Además de la lógica de negocio y datos, el Back-End es responsable de **diseñar y exponer APIs** consumidas por Front-End y terceros, **asegurar integridad, seguridad y rendimiento** (autenticación/autorización, protección OWASP, caching, escalabilidad) y **coordinar la integración** con Front-End y otros servicios.

En el **proceso de desarrollo**, define el modelo de dominio y contratos, implementa y prueba servicios, y monitorea en operación (logs, métricas, incidentes).

Fullstack

Un **desarrollador Fullstack** tiene habilidades tanto en el Front-End como en el Back-End, lo que les permite trabajar en todas las capas de una aplicación web. Son capaces de diseñar, desarrollar e implementar tanto la interfaz de usuario como la lógica del servidor. Los desarrolladores Fullstack pueden encargarse de todas las etapas del ciclo de desarrollo, desde la planificación y el diseño hasta la implementación y el despliegue. Tienen un conocimiento más amplio y una comprensión integral de todo el proceso de desarrollo web.

El Fullstack asume una **visión end-to-end: conecta Front-End y Back-End**, vela por la **coherencia de los contratos de API** y el cumplimiento de **requisitos no funcionales** (accesibilidad, compatibilidad, rendimiento y seguridad), y **coordina** entre diseño, desarrollo y operación para reducir fricciones. En el **proceso de desarrollo**, interviene en todas las fases, desde el descubrimiento y diseño técnico hasta pruebas integrales y despliegue.

Qué es el lenguaje de marcación de hipertexto (HTML)

HTML (HyperText Markup Language) es un **lenguaje de marcado utilizado para estructurar y presentar el contenido de las páginas web**. Permite definir la estructura y el significado semántico de los elementos en una página web utilizando etiquetas.

HTML es el **lenguaje fundamental para la creación de páginas web** y permite estructurar el contenido de manera semántica. Con el uso adecuado de etiquetas y elementos HTML, se pueden crear sitios web bien organizados, accesibles y compatibles con múltiples dispositivos y navegadores.

Si imaginamos que nuestra página es una persona, HTML sería el esqueleto, y cada etiqueta, un hueso. HTML utiliza las etiquetas para crear elementos, como párrafos, títulos, etc. El conjunto de elementos HTML guardados en un mismo documento con extensión .html es una página. Y un conjunto de páginas relacionadas forman un sitio web.

HTML se combina con CSS (Cascading Style Sheets) para definir el aspecto y la presentación visual de una página web, y con JavaScript para agregar interactividad y funcionalidad dinámica.

Como la mayoría de los lenguajes, HTML ha ido evolucionando desde su creación en 1991, y existen diferentes versiones. La última versión (y la que usaremos nosotros) es HTML5. Es una versión muy eficiente y compatible con distintos navegadores y plataformas (PC de escritorio, portátiles, teléfonos inteligentes, tabletas, etc.).

Características y puntos importantes sobre HTML:

- HTML es un lenguaje de marcado, lo que significa que utiliza etiquetas para definir la estructura y el significado de los elementos en una página web. Cada etiqueta tiene un propósito específico y se utiliza para envolver el contenido y darle formato.
- HTML se utiliza para definir la estructura básica de una página web, como encabezados, párrafos, listas, tablas, formularios, imágenes y enlaces. Cada elemento se representa mediante una etiqueta HTML correspondiente.

El rol del Navegador

Internet

Internet es una red de computadoras que se encuentran interconectadas a nivel mundial para compartir información. Se trata de una red de equipos de cálculo que se relacionan entre sí a través de la utilización de un lenguaje universal.

El concepto Internet tiene sus raíces en el idioma inglés y se encuentra conformado por el vocablo inter (que significa “entre”) y net (proveniente de network que quiere decir “red electrónica”). Es un término que siempre debe ser escrito en mayúscula ya que hace referencia a “La Red” (que conecta a las computadoras mundialmente mediante el protocolo TCP/IP) y sin un artículo que lo acompañe (el/la) para hacerle referencia.

Navegador Web

El navegador web o navegador de internet es el instrumento que permite a los usuarios de internet navegar o surfear entre las distintas páginas de sus sitios

webs preferidos. Se trata de un software que posee una interfaz gráfica compuesta básicamente de: botones de navegación, una barra de dirección, una barra de estado (generalmente, en la parte inferior de la ventana) y la mayor parte, en el centro, que sirve para mostrar las páginas web a las que se accede.

Los principales navegadores web del mercado son:

- Microsoft Internet Explorer (actualmente Microsoft Edge)
- Firefox
- Google Chrome
- Opera
- Safari.

Buscadores Web

Son **los programas dentro de un sitio o página web**, los cuales al ingresar palabras clave, operan dentro de la base de datos del mismo buscador y recopilan todas las páginas posibles, que contengan información relacionada con lo que se esté buscando.

Estos motores de búsqueda funcionan mediante el envío de “arañas”, las cuales son pequeños robots que se dedican a rastrear todos los sitios web a lo largo y ancho de Internet. Otro programa, llamado un indexador, a continuación, lee estos documentos y crea un índice basado en las palabras contenidas en cada documento. **Cada motor de búsqueda utiliza un algoritmo propietario para crear sus índices** tales que, idealmente, sólo los resultados significativos se devuelven para cada consulta.

Qué es la W3C

Tim Berners-Lee, es el creador de la **World Wide Web (www)**, estableció el **Consortium World Wide Web (W3C)** en 1994 con el objetivo de asegurar el crecimiento continuo y a largo plazo de la Web.

Desde sus inicios, el **W3C** ha sido una comunidad internacional que incluye a diversas partes interesadas, como organizaciones miembros, personal dedicado a tiempo completo y el público en general. Todos ellos trabajan juntos para desarrollar **estándares web abiertos**.

El W3C es liderado por un director ejecutivo interino llamado Ralph Swick y cuenta con una junta directiva. La misión principal del Consorcio es llevar la web a su máximo potencial.

El **World Wide Web Consortium (W3C)** desarrolla estándares y pautas para ayudar a todos a construir una web basada en los **principios de accesibilidad , internacionalización , privacidad y seguridad**.

El **W3C** también se preocupa por **promover** la adopción de estos **estándares a nivel global**, trabajando con gobiernos, organizaciones y la comunidad en general para fomentar su implementación. Además, el Consorcio se dedica a la investigación y la educación, brindando recursos y documentación para ayudar a las personas a comprender y utilizar eficazmente los estándares web.

Algunos de los estándares más conocidos y ampliamente utilizados incluyen:

- **HTML (HyperText Markup Language)**: Es el lenguaje de marcado estándar utilizado para crear páginas web. Define la estructura y el contenido de una página web.
- **CSS (Cascading Style Sheets)**: Es un lenguaje utilizado para describir la presentación visual de una página web, como el diseño, los colores, las fuentes y otros aspectos de estilo.
- **XML (eXtensible Markup Language)**: Es un lenguaje de marcado utilizado para almacenar y transportar datos estructurados. Se utiliza ampliamente en aplicaciones web y en la interoperabilidad de sistemas.
- **HTTP (Hypertext Transfer Protocol)**: Es el protocolo utilizado para la comunicación entre clientes y servidores en la Web. Define cómo se solicitan y se entregan los recursos web.
- **SVG (Scalable Vector Graphics)**: Es un formato de archivo basado en XML utilizado para describir gráficos vectoriales. Permite la creación de gráficos escalables y de alta calidad en la Web.
- **WCAG (Web Content Accessibility Guidelines)**: Son pautas desarrolladas por el W3C para hacer que el contenido web sea accesible para personas con discapacidades. Proporciona directrices para crear sitios web que sean comprensibles y utilizables por todos.

Evolución del html hacia el html5

- HTML 1 y 2 (1991): Elementos de texto e imágenes.
- HTML 3 (1995): Basado en Tablas. No había un estándar: Cada browser interpretaba lo que quería.
- HTML 4 (1998): Aparece CSS, primer estándar oficial.
- XHTML 1.0 (2000): Es de la W3C, basado en XML.
- TRANSITIONAL: Punto medio entre HTML4 y XHTML
- STRICT: Más restrictivo (por ende más complejo).
- HTML5 (2004): Primer borrador creado por la WHATWG.
- (2008) Pasa a la W3C. Aún lo consideran experimental

HTML5 es la última versión de HTML y ha introducido nuevas características y mejoras. Proporciona un conjunto más amplio de etiquetas y elementos para trabajar, como el uso de elementos semánticos (como `<header>`, `<nav>`, `<section>`, `<article>`, etc.) que brindan un significado más claro y ayudan en la accesibilidad y optimización para motores de búsqueda.

HTML5 es ampliamente compatible con los navegadores modernos y se ha convertido en el estándar de facto para el desarrollo de sitios web. Sin embargo, es importante tener en cuenta las diferencias en la implementación y el soporte de características específicas entre los diferentes navegadores.

Características y mejoras que HTML5 introdujo:

- ❖ **Nuevas etiquetas semánticas:** HTML5 introduce una serie de etiquetas semánticas que permiten describir con mayor precisión el contenido de una página web. Algunas de estas etiquetas incluyen `<header>`, `<nav>`, `<section>`, `<article>`, `<footer>`, entre otras. Esto facilita la comprensión del contenido y mejora el SEO.
- ❖ **Soporte multimedia:** HTML5 ofrece una integración nativa de audio y video, eliminando la necesidad de plugins como Flash. Ahora es posible incrustar elementos de audio y video utilizando las etiquetas `<audio>` y `<video>`, lo que proporciona una reproducción más fluida y una mejor compatibilidad en diferentes dispositivos.

- ❖ **Gráficos y animaciones:** HTML5 incluye el elemento **<canvas>**, que permite crear gráficos dinámicos y animaciones utilizando JavaScript. También se introdujo la **API SVG** (Scalable Vector Graphics) para la creación de gráficos vectoriales escalables en la web.
- ❖ **Formularios mejorados:** HTML5 introduce nuevos tipos de entrada de formulario, como **email, tel, date, number**, entre otros. Además, ofrece validación de formularios en el lado del cliente sin necesidad de JavaScript adicional.
- ❖ **Almacenamiento local:** HTML5 proporciona **dos nuevas API** para almacenar datos en el navegador del usuario: **LocalStorage** y **SessionStorage**. Estas API permiten a las aplicaciones web almacenar datos de forma persistente o temporal en el navegador, lo que brinda una experiencia más rápida y sin conexión.
- ❖ **Geolocalización:** HTML5 incluye una **API de geolocalización** que permite a las aplicaciones web acceder a la ubicación geográfica del usuario, siempre que se le haya dado permiso.

La triada html, css y javascript

¿Qué es el HTML?

Es un "**lenguaje**" de **marcado** (de etiquetas) para crear documentos para web. Permite indicar dónde queremos cada elemento (párrafos, negritas, itálicas, imágenes, etc.). Sólo se encarga de lo estructural, no del diseño (colores y tamaños son responsabilidad de CSS). Ha sido estandarizado por la W3C (www.w3.org). Existen varias "versiones" (html4, xhtml, html5).

HTML se centra en el contenido y proporciona la estructura básica de una página web.

- **Etiqueta:** Como bien su palabra lo indica, es una palabra que hace referencia o "etiqueta" algo, ese algo es una oración, bloque de texto, imagen o lo que se desee mostrar.
- **Atributo:** Es una característica especial que se le dará a una etiqueta.
- **Estructura:** Es la organización que tendrá todo el lenguaje para llevar un orden y un paso a paso de lo que debe realizar.

¿Qué es el CSS?

El CSS, en español «Hojas de estilo en cascada», está definido como un **lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado**. Con CSS, se pueden definir reglas y estilos que afectan la apariencia de los elementos HTML. Esto incluye propiedades como colores, fuentes, tamaños, márgenes, posicionamiento y efectos visuales. CSS separa el diseño y la presentación del contenido, lo que permite crear diseños atractivos y consistentes en todo el sitio web.

- **Estilo:** Atributos que se le asignan al HTML para darle un estilo particular.
- **Reglas:** Características que deben cumplir las sentencias a la hora de crear la hoja de estilos.
- **Medidas:** Valores que se le asignan a cada atributo para que tomen un tamaño.
- **Fuente:** Tipos de letras.

¿Qué es el JavaScript?

JavaScript es un **lenguaje de programación de alto nivel, interpretado y orientado a objetos, que se utiliza principalmente en el desarrollo web**. Fue creado para agregar interactividad y comportamiento dinámico a las páginas web. Permite realizar acciones y responder a eventos, como hacer cambios dinámicos en el contenido, validar formularios, crear animaciones, manipular elementos HTML y establecer comunicación con servidores. JavaScript complementa HTML y CSS al permitir la creación de experiencias interactivas y funcionales en el navegador del usuario.

- **Interactividad:** JavaScript permite interactuar con los elementos de una página web y responder a las acciones del usuario, como hacer clic en un botón, mover el cursor o enviar un formulario.
- **Manipulación del DOM:** JavaScript permite acceder y manipular el Document Object Model (DOM) para representar la estructura de una página web como un árbol de objetos. para modificar dinámicamente el contenido, los estilos y la estructura de una página.
- **Programación basada en eventos:** JavaScript se basa en el modelo de programación basado en eventos, donde se definen funciones que se ejecutan en respuesta a eventos específicos.

- **Funciones y variables:** JavaScript permite definir funciones y variables para organizar y reutilizar el código.
- **Comunicación con el servidor:** JavaScript facilita la comunicación con un servidor web para enviar y recibir datos en segundo plano. Esto se puede lograr mediante la tecnología AJAX (Asynchronous JavaScript and XML) o utilizando las API de Fetch y XMLHttpRequest para realizar solicitudes HTTP asíncronas.

El Entorno de Desarrollo

Editor de Texto

Son programas que **nos permite realizar o escribir código fuente de nuestros proyectos**, al ser dinámicos son idóneos para cuando se desarrolla un proyecto con varios lenguajes de programación, permitiendo que se codifican todos en un mismo lado, como por ejemplo: en el caso de diseño web se puede usar, HTML, CSS, JavaScript o PHP, con el editor de texto se puede manejar cualquier lenguaje y no causar ningún conflicto entre archivos.

El código en sí no es más que texto, que será interpretado como código cuando se ejecute en el contexto adecuado. La diferencia es que los editores de texto nos ayudan con la tarea de verificar dicho código, autocompletando funciones o palabras claves propias del lenguaje de programación.

Es de suma importancia aclarar que **Microsoft Word u OpenOffice, no son editores de texto** por más que manejen textos planos.

Entre los editores más conocidos se tiene:

- Sublime Text.
- Atom.
- Brackets.
- Visual Studio Code.
- PHPStorm.

¿Qué es Visual Studio Code?

En 2015, Microsoft lanzó una nueva herramienta llamada **Visual Studio Code**, abreviado como **VS Code**. Es un **editor de código fuente** moderno que ofrece numerosas características útiles para trabajar con código.

Visual Studio Code (VS Code) no se limita a los lenguajes de programación propios de Microsoft, como C# y VB. En cambio, **admite** una amplia gama de **lenguajes de programación**, incluyendo Java, Go, C, C++, Ruby, Python, PHP, Perl, JavaScript, Groovy, Swift, PowerShell, Rust, DockerFile, CSS, HTML, XML, JSON, Lua, F#, Batch, SQL, Objective-C, y muchos más.

Una de las ventajas de VS Code es su **compatibilidad con los tres sistemas operativos más utilizados: Windows, Linux y macOS**. Simplemente tienes que visitar el sitio web oficial y descargar los binarios correspondientes para tu sistema operativo preferido. Además, VS Code se actualiza regularmente, lo que te permite disfrutar de nuevas funciones y mejoras.

Es importante destacar que VS Code no debe confundirse con Visual Studio. Visual Studio es un entorno de desarrollo integrado (IDE) más completo y potente, mientras que VS Code es un editor de código fuente. Sin embargo, ambos ofrecen una gran productividad y se pueden complementar en diferentes escenarios de desarrollo.

Además, **VS Code proporciona integración con Microsoft Azure**, lo que te permite trabajar con servicios en la nube y desplegar proyectos directamente desde el editor.

Características

Intellisense: Visual Studio Code ofrece una función llamada Intellisense, que es capaz de predecir y autocompletar instrucciones mientras escribes código. Esto aumenta la productividad al evitar tener que escribir todo el código manualmente y reduce la posibilidad de cometer errores de sintaxis.

Open Source: Visual Studio Code es un software de código abierto y está disponible en la plataforma de desarrollo colaborativo GitHub. Esto significa que se puede descargar, examinar su código fuente, realizar modificaciones y enviar contribuciones mediante el uso de Git. Esta naturaleza de código abierto permite a la comunidad de desarrolladores personalizar y mejorar el editor según sus necesidades.

Depuración: Si bien Visual Studio Code es un editor de código poderoso, tiene ciertas limitaciones en comparación con un entorno de desarrollo integrado (IDE) completo como Visual Studio. Una de las limitaciones es la capacidad de depuración. En VS Code, se puede ver y modificar el código de un proyecto, pero no se puede ejecutar para ver las ventanas o interfaces en tiempo de ejecución ni inspeccionar los valores de los objetos mientras se ejecuta el programa.

Compilación: Visual Studio Code está diseñado principalmente como un editor de código y, por lo tanto, no incluye un compilador integrado. Esto significa que se puede editar y crear código en VS Code, pero necesitarás utilizar herramientas externas o integrar el editor con un compilador específico para compilar y ejecutar tu código.

Instalar Visual Studio Code

1. Abrir el navegador web e ingresa a la página oficial de Visual Studio Code: <https://code.visualstudio.com/>
2. Visual Studio Code es compatible con los sistemas operativos Windows, macOS y Linux. En la página principal, encontrarás información sobre sus características, extensiones y documentación.
3. En la página principal, se verá un botón de **descarga** grande y destacado. Haz clic en él para comenzar la descarga. Asegúrate de seleccionar la versión adecuada para tu sistema operativo. Por ejemplo, si usas Windows, deberías descargar el instalador para Windows.
4. Una vez completada la descarga, localiza el archivo de instalación en tu computadora. En la mayoría de los casos, se encontrará en la carpeta "Descargas". Haz doble clic en el archivo para ejecutarlo y comenzar la instalación.
5. Acepta los términos de licencia y elige las configuraciones predeterminadas.
6. Durante la instalación, se te pedirá que aceptes los términos de licencia. Lee los términos y, si estás de acuerdo, marca la casilla correspondiente. Luego, elige las configuraciones predeterminadas o personaliza las opciones según tus preferencias. Por lo general, las opciones predeterminadas son adecuadas para la mayoría de los usuarios.

Una vez que hayas configurado las opciones de instalación, haz clic en el botón "Instalar" o "Siguiente" para comenzar el proceso de instalación. El instalador copiará los archivos necesarios y configurarlo a Visual Studio Code en tu sistema. Una vez que la instalación esté completa, encontrarás el icono de Visual Studio Code en tu escritorio o en el menú de inicio (en Windows). Haz clic en el icono para iniciar el editor.

Conociendo el inspector de elementos en un navegador

Es una **herramienta** que se encuentra en la mayoría de los navegadores web modernos y **permite a los desarrolladores inspeccionar y analizar el código HTML, CSS y JavaScript de una página web en tiempo real**. Proporciona una forma interactiva de examinar la estructura y los estilos de los elementos de una página, así como de depurar problemas y realizar cambios en el código.

Características clave del inspector de elementos:

- **Inspección de elementos:** Se puede seleccionar y examinar cualquier elemento de una página web, ya sea un párrafo, una imagen, un botón u otro elemento HTML. Esto te permite ver su estructura HTML, sus atributos, estilos aplicados y otros detalles relevantes.
- **Edición en vivo:** Se puede realizar cambios en el código HTML y CSS de una página directamente en el inspector de elementos y ver los resultados en tiempo real. Esto es útil para experimentar con diferentes estilos o realizar ajustes rápidos sin tener que modificar los archivos originales.
- **Depuración de JavaScript:** El inspector de elementos también ofrece herramientas de depuración para el código JavaScript. Se pueden establecer puntos de interrupción, examinar el estado de las variables, seguir la ejecución del código paso a paso y realizar un seguimiento de los errores y excepciones.
- **Estilos y diseño:** Se puede ver y modificar los estilos CSS aplicados a los elementos de una página, lo que te permite experimentar con diferentes

estilos, ajustar márgenes y dimensiones, cambiar colores y fuentes, entre otros.

- **Análisis de rendimiento:** El inspector de elementos proporciona información detallada sobre el rendimiento de la página, como tiempos de carga, uso de recursos y rendimiento de scripts. Esto te permite identificar cuellos de botella y optimizar el rendimiento de la página.

¿Cómo funciona el inspector web?

Para usar el inspector web, sigamos estos pasos:

1. Abre la herramienta del inspector web en tu navegador. Para esto dirígete a los tres puntos en el borde superior derecho, opción más herramientas, herramienta para inspectores o Ctrl + Shift + I
2. Ingresa "google.com" en la barra de búsqueda y presiona Enter.
3. En la sección de "Network" o "Red", verás una lista de elementos clasificados por nombre, estado, tipo, iniciador y tamaño.
4. Haz clic en la primera petición, llamada "google.com", para ver sus detalles.
5. En la sección de "Headers" o "Cabeceras", encontrarás información como el método de petición (GET), la URL solicitada y el código de estado (301).
6. También encontrarás dos tipos de cabeceras: cabeceras de petición (request headers) y cabeceras de respuesta (response headers).
7. Las cabeceras de petición viajan del navegador al servidor, mientras que las cabeceras de respuesta viajan del servidor al navegador.
8. Algunos ejemplos de cabeceras comunes son "User-Agent", que identifica el navegador, y "Content-Length", que indica el tamaño de la respuesta del servidor.
9. En la categoría de "Response" o "Respuesta", puedes ver el estado de la respuesta y su código.
10. La categoría de "Preview" o "Previsualización" muestra cómo el navegador interpreta el código de la respuesta.

11. Podes encontrar diferentes tipos de interpretaciones en la previsualización, como texto, imágenes o archivos SVG.
12. Explora las distintas peticiones y categorías disponibles en el inspector web para obtener más información sobre cómo funciona.

Herramientas que nos permiten inspeccionar

Usar herramientas para inspeccionar es un plus. Esto nos permite hacer de la vida del desarrollado más fácil. Chrome DevTools es un conjunto de herramientas para desarrolladores web integrado en el navegador Google Chrome. Con estas herramientas se puede inspeccionar el HTML renderizado (DOM) y la actividad de red de sus páginas. Se puede utilizar DevTools para solucionar problemas relacionados con el servicio de anuncios.

Link: [DevTools](#)

Cierre

Como pudimos abordar en esta lección, el desarrollo web implica la creación de sitios web y aplicaciones web utilizando tecnologías y lenguajes de programación específicos. Los desarrolladores web se encargan del diseño, implementación y mantenimiento de estos sitios. Los componentes principales son el cliente, el servidor y la base de datos. El cliente es la interfaz con la que los usuarios interactúan, el servidor procesa las solicitudes y la base de datos almacena la información del sitio web. Además, existen diferentes roles en el desarrollo web, como los desarrolladores Front-End, Back-End y Fullstack.

Comprendimos que HTML es fundamental para el desarrollo web, nos brinda estructura y presenta el contenido. CSS define el aspecto visual y JavaScript agrega interactividad. Los navegadores web permiten navegar por las páginas web, como Chrome, Firefox, Safari y Edge. Los motores de búsqueda recopilan información basada en palabras clave.

Por último, aprendimos que el W3C desarrolla estándares para la web, promoviendo accesibilidad, internacionalización, privacidad y seguridad. Estándares conocidos incluyen HTML, CSS y XML.

Referencias

Mozilla. (s.f.). HTML: Lenguaje de marcado de hipertexto. MDN Web Docs.

<https://developer.mozilla.org/es/docs/Glossary/HTML>

Mozilla. (s.f.). CSS: Cascading Style Sheets. MDN Web Docs.

<https://developer.mozilla.org/en-US/docs/Web/CSS>

World Wide Web Consortium (W3C). (s.f.). W3C – World Wide Web Consortium.

<https://www.w3.org/>

Microsoft. (s.f.). Visual Studio Code documentation.

<https://code.visualstudio.com/docs>

¡Muchas gracias!

Nos vemos en la próxima lección

