



Tarea #1

María Jesús Rodríguez Molina

Universidad CENFOTEC

Proyecto Web 1

Profesor:

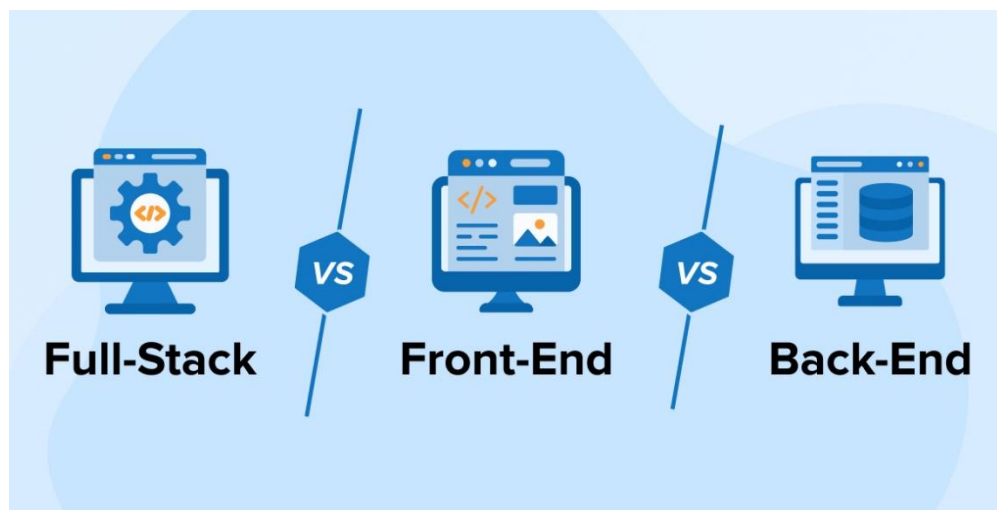
Francisco Jiménez Bonilla

III Cuatrimestre, septiembre 2025

Desarrollo

1-) ¿Cuál es la diferencia entre los términos: Front-End, Back-End, Full-Stack?

- **Front-End:** Corresponde a la parte visual de una aplicación o página web, todo lo que el usuario ve y con lo que interactúa (logotipos, fuentes, botones, colores, etc.). El trabajo de un desarrollador Front-End se enfoca en el diseño web, navegación, estructura, estandarización visual y experiencia de usuario (UX/UI), esto empleando las tecnologías de HTML, CSS y JavaScript. Se resume en lo que el usuario puede ver.
- **Back-End:** Corresponde al apartado de la lógica interna de los sitios web, donde se incluyen los servidores, bases de datos, APIs, etc. Dentro del rol de un desarrollador Back-End, la persona se responsabiliza por trabajar la arquitectura del servidor, lógica de negocio, seguridad, procesamiento de datos y comunicación entre los diferentes componentes de la página, esto usando tecnologías como PHP, Python y SQL. Se resume en lo que el usuario no puede ver.
- **Full-Stack:** Corresponde al desarrollo completo de páginas/aplicaciones web. Una persona capacitada para trabajar como Full-Stack maneja tanto la parte visual (Front-End) como la lógica interna (Back-End). Por ende, dentro del rol de un desarrollador Full-Stack se encarga de crear un producto web completo.

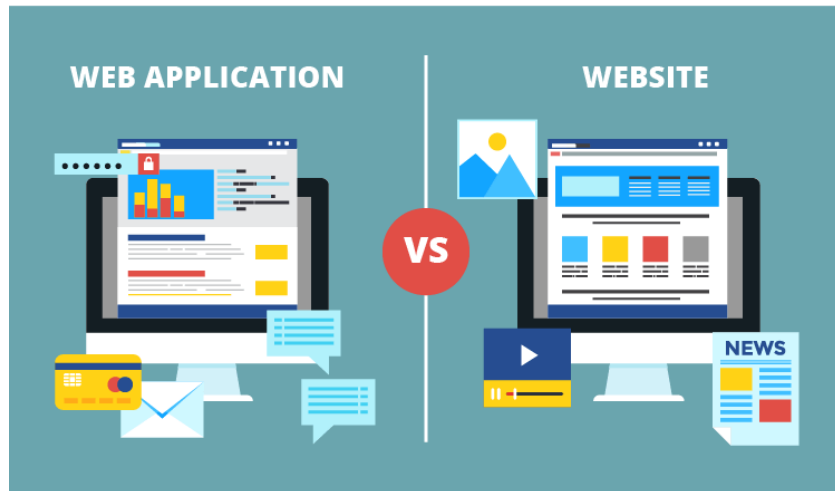


2-) ¿Cuál es la funcionalidad de los siguientes términos: (sistema-sitio web-aplicaciones-redes sociales)? Piense en el objetivo de cada uno.

- **Sistema:** Corresponde a un conjunto de programas y/o componentes interrelacionados, encargados de interactuar entre sí para cumplir con una función específica. Su función depende del tipo de sistema. El ejemplo más claro es un sistema bancario donde se permite manejar cuentas, transacciones, resolver necesidades complejas...
- **Sitio web:** Corresponde a un conjunto de páginas accesibles desde un navegador en las cuales se presenta información bajo un nombre de dominio. Su objetivo es informar, entretener, proveer servicios o plataformas mediante información ya sea estática o dinámica y se construyen a base de tecnologías como HTML, CSS y JavaScript.
- **Aplicaciones:** Programas de software específicos para dispositivos finales como computadoras y dispositivos móviles, diseñadas y creadas para ejecutar tareas o servir de algún modo específico. Las aplicaciones se pueden encontrar en aplicaciones web o ser instalados directamente en el equipo. Su objetivo es facilitar una funcionalidad al usuario de manera específica, como transacciones bancarias, entretenimiento, comunicación y funcionalidades avanzadas...
- **Redes sociales:** Plataformas web y/o aplicaciones especializadas en crear comunidades e interacciones entre personas de manera conectada. Su objetivo es mantener y promover el contacto social.

Diferencias:

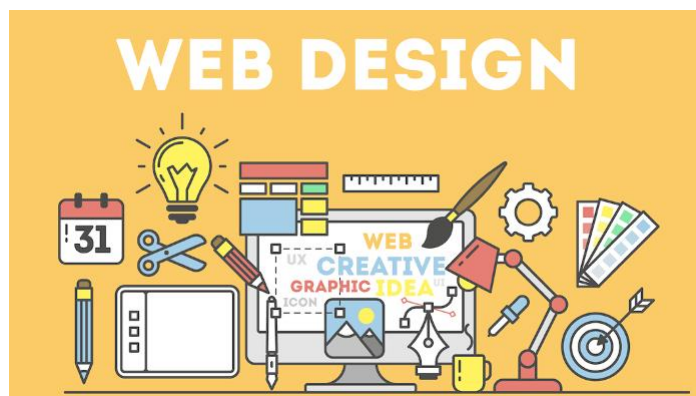
- Un sitio web informa.
- Una aplicación permite realizar alguna acción.
- Una red social es una aplicación específica para temas sociales.
- Un sistema es un término más amplio donde se engloba a los elementos anteriores para un propósito específico.



3-) ¿Qué es la programación al lado cliente? Cite 3 ejemplos.

También llamada Front-End, corresponde al código ejecutado directamente en el navegador del usuario con el propósito de mejorar la experiencia del usuario final y reflejar la experiencia visual e interacción del cliente mediante efectos visuales, validación, manejo de eventos y de errores, entre otros. Ejemplos:

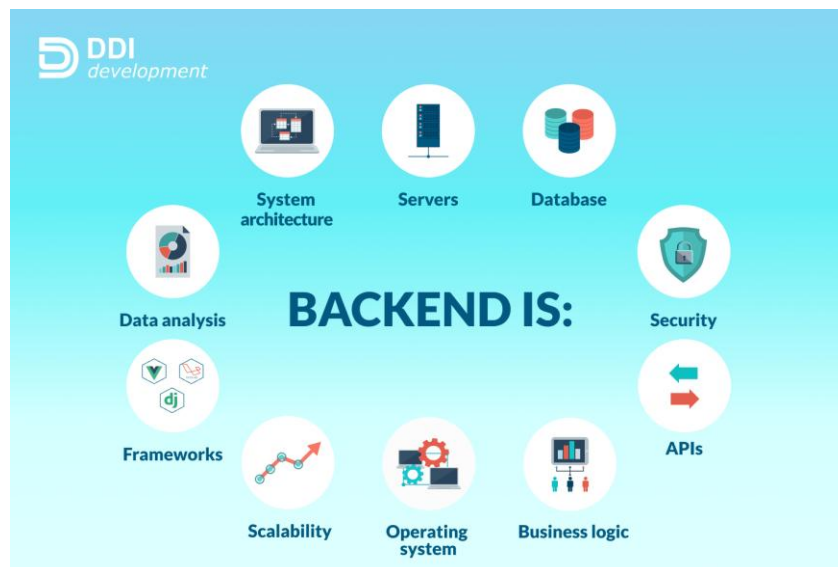
- Validar que información de un formulario sea correcta mediante el uso de JavaScript, por ejemplo, a la hora de validar si un correo es correcto o no.
- Crear un menú desplegable y funcional dentro de la página web mediante el uso de CSS y HTML, el cual posee una animación específica a la hora de abrir el mismo.
- Visualizar elementos como botones y que los mismos reaccionen con una animación o cambien de color cuando se interactúa de alguna forma con ellos (CSS y JavaScript).



4-) ¿Qué es la programación al lado servidor? Cite 3 ejemplos.

También llamado Back-end, corresponde a las acciones o ejecuciones realizadas dentro de un servidor, donde se procesa información, se gestiona la seguridad y se conecta la web con la base de datos. Ejemplos:

- Recibir información de un formulario (nombre, correo y demás datos) un cliente mediante PHP y almacenar dicha información en una base de datos MySQL.
- Revisar que el inicio de sesión de un cliente sea correcto buscando la información registrada en la base de datos MySQL.
- Procesar una compra en línea, donde el servidor valida el pago utilizando Node.js, y luego actualizando el estado de la compra a “pagado”.

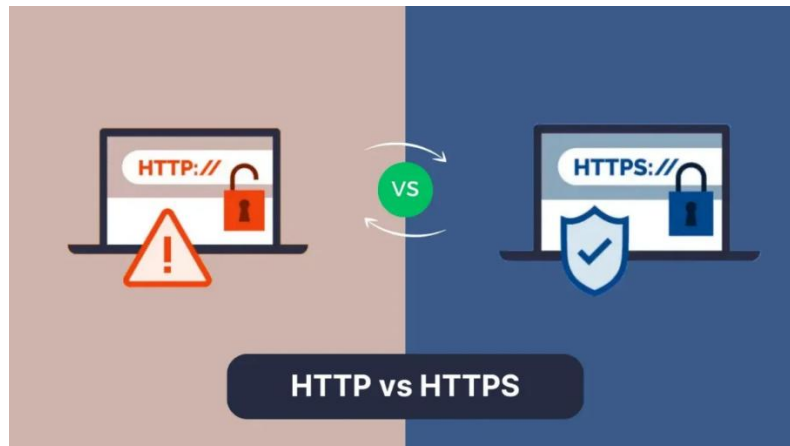


5-) ¿Que es un protocolo HTTP-HTTPS y que tipos existen?

HTTP (Hypertext Transfer Protocol) corresponde a un protocolo de la capa de aplicación, es decir, reglas de comunicación enfocadas a definir el cómo se comunican los clientes con un servidor a la hora de acceder a páginas web. En este escenario el navegador envía una solicitud HTTP al servidor, la cual sirve el contenido deseado.

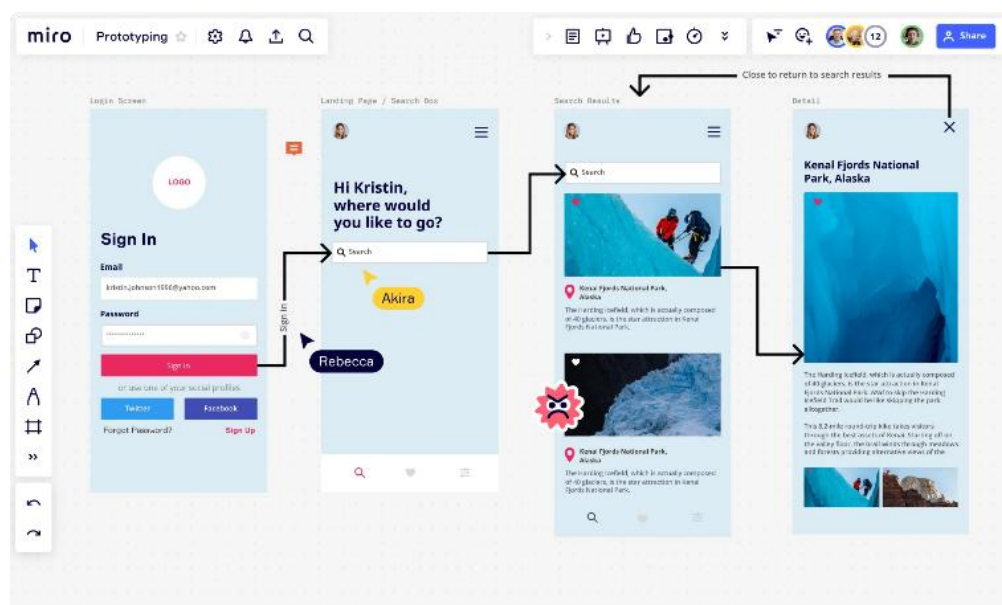
Con respecto a las diferencias de HTTP y HTTPS, en el caso de HTTP el contenido transferido no es cifrado, por lo que la información es susceptible a ser interceptada y leída por terceros. Es por eso por lo que existe HTTPS (Hypertext Transfer Protocol

Secure), que incorpora una capa extra de seguridad por medio de certificados SSL para lograr cifrar la información enviada y recibida entre el navegador y el servidor web, logrando que los datos permanezcan privados y protegidos de personas que deseen interceptarlos.



6-) ¿A qué se le llama un prototipo de una aplicación web?

Un prototipo de una aplicación web corresponde a una representación visual tipo maqueta en la cual se planea cómo se verá y funcionará una página web antes de realizar la programación. Se pueden realizar prototipos en herramientas como Figma, y su propósito es mostrar cómo se verían los diferentes botones, campos de texto, colores de la interfaz, etc.



7-) Describir la historia del lenguaje HTML hasta llegar al HTML5

- **1991 – Nacimiento de HTML**

- HTML fue creado por Tim Berners-Lee, sin embargo, consistía en una primera versión muy simple, basada en un conjunto básico de etiquetas donde solo permitía enlazar documentos con hipervínculos y su objetivo era meramente compartir y divulgar información de índole científica a modo de textos simples.

- **1995 – HTML 2.0**

- Corresponde a la primera versión estandarizada. Si bien no era un HTML muy novedoso, sino más un estándar definido, permitió mayor interacción en comparación a su versión anterior.

- **1997 – HTML 3.2**

- Añadió soporte para scripts (JavaScript) y hojas de estilo (CSS) como las conocemos hoy en día. Así mismo, permitió insertar imágenes más flexibles, tablas avanzadas y applets de Java y fue clave para las páginas web dinámicas.

- **1999 – HTML 4.0 y 4.01**

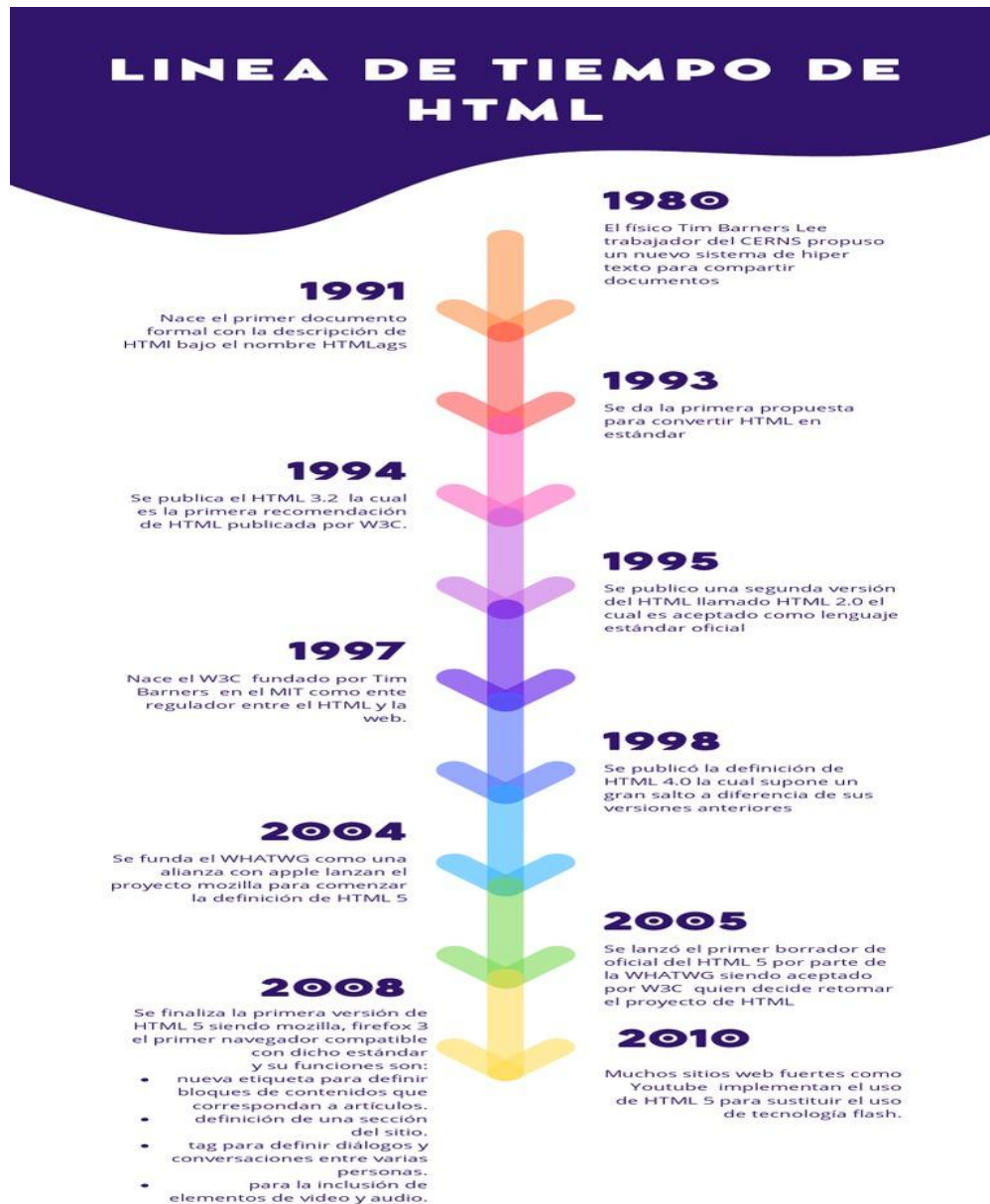
- Fue muy usado durante más de 10 años, incluso en combinación con Flash para multimedia. Esta versión lograba obtener soporte para estilos, capas y marcos de líneas. Servía a su vez como parche y corrección de errores de versiones anteriores.

- **2000 – XHTML**

- Variante de HTML basada en XML. Más estricta en la sintaxis, basada en XML, pero no reemplazó por completo a HTML 4.01. Su objetivo fue proporcionar mayor estructura al lenguaje.

- **2014 – HTML5 (versión oficial)**

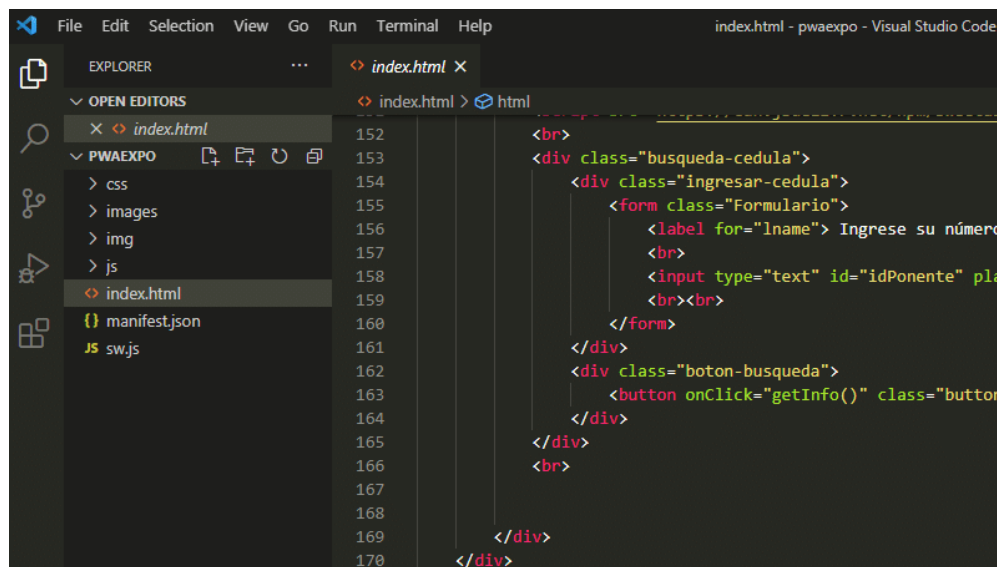
- Evolución moderna de HTML donde se introducen importantes nuevas características para audio, video, formularios e interacción en general sin necesidad de tecnología externa como Flash. Introduce etiquetas semánticas nuevas: <header>, <footer>, <article>, <section> y se convierte en la base de la web moderna junto con CSS3 y JavaScript.



8-) ¿Qué es el HTML5 y cuál es su función principal?

Corresponde a la última versión de HTML (Lenguaje de Marcado de Hipertexto). A diferencia de las versiones anteriores de HTML, esta forma de organizar los elementos de la página web se realizan por medio de semánticas, que no son más que etiquetas las cuales describen el tipo de contenido trabajado, proporcionando una descripción clara de lo que se trabaja, facilitando la programación y comprensión.

Su función principal es estructurar y presentar contenido web organizado y definido, como texto en diferentes niveles, imágenes, hipervínculos... Dentro de la programación web corresponde a algo así como el “esqueleto” de una casa.





```
152 <br>
153 <div class="busqueda-cedula">
154   <div class="ingresar-cedula">
155     <form class="Formulario">
156       <label for="lname"> Ingrese su número
157       <br>
158       <input type="text" id="idPonente" placeholder="Ingrese su número"
159       <br><br>
160     </form>
161   </div>
162   <div class="boton-busqueda">
163     <button onClick="getInfo()" class="button">
164   </div>
165 </div>
166 <br>
167
168
169 </div>
170 </div>
```

9-) ¿Qué es el CSS y cuál es su función principal?

Corresponde a “Hojas de estilo de cascada”, un lenguaje donde se le da estilo y presentación a los documentos de HTML para servir diferentes interacciones con los elementos.

Su función es mejorar y controlar la apariencia de la página web y sus elementos, dando colores, fuentes, diseño, posicionamiento de elementos, animaciones y experiencia de usuario atractiva a nivel visual. Dentro de la programación web corresponde a algo así como la pintura, muebles y adornos de una casa.

```

34  /* A reference to a type */
35  span.ts span.type-ref {
36    color:  rgb(175, 0, 219) !important;
37  }
38
39  /* Signature details */
40  div.signature > table {
41    border-collapse: collapse;
42    border: thin  darkgray solid;
43    width: 60%;
44  }

```

10-) ¿Qué es Java Script y cuál es su función principal?

Corresponde a un lenguaje de programación que provee una capa de interacción y dinamismo a los elementos del sitio web, así como procesar y transformar datos enviados y recibidos. Su función es permitir el uso de animaciones, respuestas a eventos, actualización de contenido y demás dinámicas. Dentro de la programación web corresponde a algo así como las luces funcionales y timbres de una casa.

```

15
16  const LOCALE = globalThis.navigator.language
17
18  const div = document.body.appendChild(document.createElement('div'))
19  const list = div.appendChild(document.createElement('ol'))
20
21  const dayNames = new Map()
22
23  for (let i = 0; i < 7; ++i) {
24    const d = Temporal.PlainDate.from({
25      year: Temporal.Now.plainDateISO().year,
26      month: 1,
27      day: i + 1,
28    })
29
30    dayNames.set(d.dayOfWeek, d.toLocaleString(LOCALE, { weekday: 'long' }))
31  }
32
33  for (const num of [...dayNames.keys()].sort((a, b) => a - b)) {
34    list.appendChild(Object.assign(
35      document.createElement('li'),
36      { textContent: dayNames.get(num) },
37    ))
38  }
39

```

Conclusión

En conclusión, el desarrollo web involucra un amplio abanico de destrezas y tecnologías de las que podemos imaginar como usuarios inexpertos. A lo largo de este documento se logró explorar cada una de las categorías existentes en el desarrollo de software como Front-End, Back-End y Full-Stack, las herramientas básicas como HTML, CSS y JavaScript, así como protocolos de comunicación y seguridad como HTTP y HTTPS y su importancia dentro del desarrollo de software, los cuales son una visión fundamental para considerar a la hora de crear un producto de calidad, seguridad y funcionalidad.