

1. Desarrollo web:

Proceso de creación, mantención o gestión de servicios relacionados al internet. Creación de páginas web, sistemas de web, aplicaciones, etc. Para ello es necesario programar, diseñar y análisis de interacciones en web para una mejor experiencia para los clientes y/o usuarios.

2. Diferencias entre Front-End, Back-End y Fullstack

El front se dedica a trabajar la parte visual de la interfaz. Ver que la información esté organizada en la forma que se prestableció y que la interacción responda de la forma esperada. El backend es el que se encarga de hacer las conexiones detrás de lo visual. Que al enviar un formulario, este esté conectado con una base de datos, que exista una lógica detrás de los procesos, y que las variables o constantes que se definan dentro del entorno de la página web o app o desarrollo responda de la manera adecuada. El Full Stack es el que es capaz de hacer estas dos áreas sin problema.

3. **Lenguaje de marcación de hipertexto (HTML):**

- Documento HTML básico:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document title</title>

  <link rel="stylesheet" href="ubicación/carpeta/archivo.css">
</head>
<body>
<script href="ubicación/carpeta/archivo.js"></script>
  <h1>Hola Archivo de Doc</h1>
</body>
</html>
```

4. El rol del Navegador:

Interpreta y muestra de la manera esperada los documentos HTML.

5. W3C (World Wide Web Consortium):

Permite construir una compatibilidad y estandarización para el uso de archivos en HTML. De esta forma, hay una coherencia en lo preestablecido y trabajado en html. También crea pautas para buenas prácticas en cuanto al uso de los elementos en HTML.

6. Evolución del HTML hacia el HTML5:

En html5, lo principal fue la incorporación de una semántica mejorada en cuanto a nuevas etiquetas como el header, footer y nav, entregando mejor estructura de los archivos. También incorporó la multimedia, con soporte nativo para audio y video en etiquetas <audio> y <video>. Otro elemento importante fue la incorporación y adaptación de apis

como la geolocalización, almacenamiento, canvas y así varias, permitiendo una mejor interacción de los usuarios.

7. La triada HTML, CSS y JavaScript:

html es la información de la página, los elementos, el orden, el párrafo y el espacio. El CSS es el orden, la coherencia, le entrega una estructura espacial a los elementos, junto a un estilo y una gráfica. El Javascript le permite interactuar y generar una dinámica más realista al usuario. Un buen ejemplo similar a esto, sería la estructura y proceso de una casa, en donde HTML son los cimientos, las murallas, las divisiones, la materia gruesa. El CSS son los guardapolvos, la cerámica, el piso flotante, las puertas, las ventanas, las luces, y esas cosas más visuales. Y el Javascript sería las conexiones eléctricas, el gas, las tuberías de agua, y esa clase de interacciones que le permiten dar vida a una casa.

8. Utilizar el potencial de un editor de texto para el desarrollo:

1. Editar archivos de manera rápida y concisa, con una gran cantidad de adaptaciones de formatos.
2. Compatibilidad con Github.
3. Muchas extensiones que permiten personalizar el espacio y forma de trabajo.

9. Conociendo el inspector de elementos en un navegador:

Para abrir el inspector de elementos, doy click derecho en cualquier parte de una página, y le doy a inspeccionar. Según el tipo de página y lo que quiero ver, mover la herramienta a abajo o lo dejo al costado. Por ejemplo, si quiero ver responsividad, es mejor moverlo a la parte inferior de la página.

El inspector tiene varias pestañas. Las más relevantes son la de Elementos, que nos mostraran el html y el css respectivo de la página, permitiendo mostrar a qué elemento visual renderizado se refiere cada sección. También está la pestaña de Consola, en donde nos mostrará si hay algún tipo de error en la página, en cuanto al área de back o programación.