



Tarea #1

Proyecto Web 1

Profesor:

Francisco Jose Jimenez Bonilla

Alumno:

Pablo Chavarria Campos

Fecha de entrega:

Semana 3

I Ciclo 2025



Índice

1. ¿Cuál es la diferencia entre los términos: Front-End, Back-End, Full-Stack?.....	3
2. ¿Cuál es la funcionalidad de los siguientes términos?: Sistema, Sitio web, Aplicaciones, Redes sociales. Piense en el objetivo de cada uno.....	4
3. ¿Qué es la programación al lado cliente? Cite 3 ejemplo.....	5
4. ¿Qué es la programación al lado servidor? Cite 3 ejemplos.....	6
5. ¿Qué son los protocolos HTTP y HTTPS, y que tipos existen?.....	7
6. ¿A qué se le llama un prototipo de una aplicación web?.....	8
7. ¿Describir la historia del lenguaje HTML hasta llegar al HTML5?.....	9
8. ¿Qué es el HTML5 y cuál es su función principal?.....	10
9. ¿Qué es el CSS y cuál es su función principal?.....	11
10. ¿Qué es Java Script y cuál es su función principal?.....	12
¿Qué enseñanza le brindó la tarea?.....	13

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

▶▶ **Front-End:** Se refiere a la parte visible y con la que interactúa el usuario en una aplicación o sitio web (ej: diseño, interfaz). Usa tecnologías como HTML, CSS y JavaScript.

▶▶ **Back-End:** Gestiona la lógica, servidores y bases de datos detrás de escena. Trabaja con lenguajes como Python, Java o Node.js y herramientas como bases de datos.

▶▶ **Full-Stack:** Combina ambas áreas, manejando tanto Front-End como Back-End para desarrollar aplicaciones completas.



Full Stack Developer - Dalle AI generated

2. ¿Cuál es la funcionalidad de los siguientes términos?: Sistema, Sitio web, Aplicaciones, Redes sociales. Piense en el objetivo de cada uno.

Sistema:

▶▶ **Funcionalidad:** Organizar y gestionar componentes (hardware, software, datos) para cumplir un propósito específico.

▶▶ **Objetivo:** Automatizar procesos, mejorar la eficiencia o resolver problemas complejos (ej: sistema operativo, sistema de inventario).

Sitio web:

▶▶ **Funcionalidad:** Mostrar contenido o servicios accesibles mediante un navegador.

▶▶ **Objetivo:** Informar, vender, comunicar o interactuar con usuarios (ej: blog, tienda en línea).

Aplicaciones:

▶▶ **Funcionalidad:** Ejecutar tareas específicas en dispositivos (móviles, desktop, web).

▶▶ **Objetivo:** Solucionar necesidades concretas del usuario (ej: app de mensajería, app de gestión financiera).

Redes sociales:

- ▶▶ **Funcionalidad:** Conectar usuarios para compartir contenido, ideas o interactuar.
- ▶▶ **Objetivo:** Fomentar la comunicación, el entretenimiento o la construcción de comunidades (ej: Instagram, LinkedIn).

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

Programación del lado cliente es el código que se ejecuta en el navegador del usuario, permitiendo interactividad y dinamismo sin depender del servidor. Usa tecnologías como HTML, CSS y JavaScript.

Ejemplos:

- ▶▶ Validación de formularios en tiempo real (ej: verificar campos vacíos).
- ▶▶ Animaciones interactivas en una página web (ej: efectos al hacer clic).
- ▶▶ Aplicaciones de una sola página (SPA) que actualizan contenido sin recargar (ej: usando React o Angular).

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

Programación del lado servidor es el código que se ejecuta en el servidor, gestionando lógica, bases de datos y generando contenido dinámico para enviar al cliente. Usa lenguajes como Python, PHP, Java o Node.js.

Ejemplos:

- ▶▶ Autenticación de usuarios (ej: verificar credenciales en una base de datos).
- ▶▶ Procesamiento de pagos en una tienda en línea.
- ▶▶ Generación de contenido personalizado (ej: mostrar productos recomendados).

5. ¿Qué son los protocolos HTTP y HTTPS, y que tipos existen?

HTTP (HyperText Transfer Protocol):

- ▶▶ Es un protocolo de comunicación que permite la transferencia de información en la web.
- ▶▶ Funciona sobre el modelo cliente-servidor: el navegador (cliente) solicita datos y el servidor responde.
- ▶▶ No es seguro. Los datos se transmiten en texto plano, lo que los hace vulnerables a interceptaciones.



HTTPS (HyperText Transfer Protocol Secure):

- » Es la versión segura de HTTP, que utiliza cifrado (SSL/TLS) para proteger la transferencia de datos.
- » Garantiza la autenticidad del servidor y la integridad de la información.
- » Es esencial para proteger información sensible, como contraseñas o datos de tarjetas de crédito.



Tipos de métodos HTTP/HTTPS:

- » **GET:** Solicita datos de un recurso específico (ej: cargar una página web).
- » **POST:** Envía datos al servidor para crear o actualizar recursos (ej: enviar un formulario).
- » **PUT:** Actualiza un recurso existente en el servidor.
- » **DELETE:** Elimina un recurso específico.
- » **PATCH:** Aplica modificaciones parciales a un recurso.
- » **HEAD:** Solicita solo los encabezados de una respuesta, sin el cuerpo.
- » **OPTIONS:** Describe las opciones de comunicación para un recurso.

Estos métodos son la base de las operaciones en APIs y aplicaciones web.

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

Un **prototipo de una aplicación web** es una versión preliminar y simplificada que sirve para visualizar y probar la estructura, diseño y funcionalidades básicas antes de desarrollar el producto final.

Sus características principales son:

- ▶▶ No es funcional al 100%: Simula la interacción y el flujo de la aplicación sin tener toda la lógica programada.
- ▶▶ Enfocado en la experiencia de usuario (UX): Permite validar la usabilidad y el diseño.
- ▶▶ Herramientas comunes: Se crea con herramientas como Figma, Adobe XD, Sketch o incluso con código básico (HTML/CSS/JS).

Objetivos:

- ▶▶ Validar ideas y conceptos con stakeholders o usuarios.
- ▶▶ Identificar problemas de diseño o flujo antes del desarrollo.
- ▶▶ Servir como guía visual para el equipo de desarrollo.

7. ¿Describir la historia del lenguaje HTML hasta llegar al HTML5?

HTML (1991):

- ▶▶ Creado por Tim Berners-Lee para compartir documentos científicos.
- ▶▶ Versión básica con etiquetas simples como <p>, <a>, y <h1>.

HTML 2.0 (1995):

- ▶▶ Primera versión estandarizada por el IETF.
- ▶▶ Introdujo formularios y tablas.

HTML 3.2 (1997):

- ▶▶ Desarrollado por el W3C.
- ▶▶ Agregó soporte para applets de Java y mejoras en tablas y formularios.

HTML 4.01 (1999):

- ▶▶ Introdujo hojas de estilo (CSS) y mejoras en accesibilidad.
- ▶▶ Última versión clásica antes de XHTML.

XHTML (2000):

- ▶▶ Reformulación de HTML basada en XML.
- ▶▶ Más estricto en sintaxis, pero no tuvo gran adopción.

HTML5 (2014):

- ▶▶ Versión moderna y revolucionaria.

- ▶▶ Introdujo nuevas etiquetas semánticas (<header>, <footer>, <article>), soporte multimedia nativo (<audio>, <video>), y APIs para aplicaciones web avanzadas (geolocalización, almacenamiento local).
- ▶▶ Enfoque en compatibilidad y experiencia de usuario en dispositivos móviles.

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



HTML5 es la quinta y última versión del lenguaje de marcado HTML (HyperText Markup Language), utilizado para estructurar y presentar contenido en la web. Fue oficialmente lanzado en 2014 por el W3C (World Wide Web Consortium).

Función principal:

- ▶▶ Crear y organizar el contenido de páginas y aplicaciones web de manera semántica y accesible.
- ▶▶ Permitir la integración de multimedia (audio, video) y gráficos sin necesidad de plugins externos.
- ▶▶ Facilitar el desarrollo de aplicaciones web interactivas y dinámicas mediante nuevas APIs y características.

Características clave:

- ▶▶ Etiquetas semánticas: <header>, <footer>, <article>, <section>, etc., para mejorar la estructura y accesibilidad.
- ▶▶ Multimedia nativo: Soporte integrado para <audio> y <video>.
- ▶▶ Gráficos y animaciones: Uso de <canvas> y SVG para gráficos dinámicos.

- ▶▶ Almacenamiento local: APIs como localStorage y sessionStorage para guardar datos en el navegador.
- ▶▶ Compatibilidad con dispositivos móviles: Diseño responsive y mejoras para experiencias en móviles.

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



CSS (Cascading Style Sheets) es un lenguaje de diseño utilizado para definir la presentación y el estilo de documentos HTML o XML. Fue creado en 1996 por el W3C (World Wide Web Consortium).

Función principal:

- ▶▶ Controlar el aspecto visual de una página web, incluyendo colores, fuentes, diseños, espaciados y animaciones.
- ▶▶ Separar el contenido (HTML) de su presentación (diseño), lo que facilita el mantenimiento y la consistencia en el diseño.

Características clave:

- ▶▶ **Selectores:** Permiten aplicar estilos a elementos específicos del HTML.
- ▶▶ **Propiedades:** Controlan aspectos como color, tamaño, margen, padding, bordes, etc.
- ▶▶ **Responsive Design:** Media queries permiten adaptar el diseño a diferentes dispositivos y tamaños de pantalla.

▶▶ **Animaciones y transiciones:** Crear efectos visuales dinámicos sin necesidad de JavaScript.

▶▶ **Frameworks y preprocesadores:** Herramientas como Bootstrap, Sass o Less amplían las capacidades de CSS.

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



Es un lenguaje de programación de alto nivel, interpretado y orientado a objetos, utilizado principalmente para añadir interactividad y dinamismo a las páginas web.

Función principal:

- ▶▶ Crear contenido dinámico y manipular el DOM.
- ▶▶ Gestionar eventos (clics, teclas, etc.).
- ▶▶ Comunicarse con servidores (AJAX, APIs).

Características:

- ▶▶ Interpretado y de alto nivel.
- ▶▶ Orientado a objetos.
- ▶▶ Multiplataforma (navegadores, servidores con Node.js).
- ▶▶ Compatible con HTML y CSS.
- ▶▶ Soporta programación asíncrona.

¿Qué enseñanza le brindó la tarea?

La tarea me enseñó las bases de la web cómo el Front-End (lo que se ve) y el Back-End (lo que no se ve) trabajan en equipo, usando herramientas como HTML5 para estructurar, CSS para diseñar y JavaScript para dar vida a las páginas. También aprendí la importancia de protocolos como HTTPS para seguridad, cómo crear prototipos antes de programar.