

Fundamentos del Desarrollo Web

I. Pilares de las Tecnologías Web

El ecosistema web se sustenta en diversas tecnologías que interactúan para construir y presentar el contenido que vemos en línea.

A. Tecnologías Esenciales: La Base de la Web

- **HTML (HyperText Markup Language):** El lenguaje fundamental para estructurar el contenido de las páginas web. Define los elementos (párrafos, encabezados, imágenes, enlaces) y su jerarquía, dando forma al esqueleto de cualquier sitio.
- **CSS (Cascading Style Sheets):** Responsable de la apariencia visual de la web. Permite controlar el diseño, los colores, las fuentes, la disposición de los elementos HTML y la adaptación a diferentes dispositivos, haciendo que las páginas sean estéticamente atractivas.
- **HTTP (Hypertext Transfer Protocol):** El protocolo de comunicación esencial para la World Wide Web. Facilita la transmisión de documentos hipertexto (como páginas HTML, imágenes y videos) entre los navegadores web (clientes) y los servidores.

B. Lenguajes de Interacción: Dinamismo en el Navegador

- **JavaScript:** Un lenguaje de programación versátil que se ejecuta directamente en el navegador del usuario. Es crucial para crear sitios web interactivos, animaciones, manipular el contenido de forma dinámica y desarrollar aplicaciones web complejas.
- **Web APIs (Application Programming Interfaces):** Conjuntos de interfaces que permiten a los desarrolladores web acceder a funcionalidades avanzadas del navegador y del dispositivo. Esto incluye desde la manipulación del Modelo de Objeto del Documento (DOM) hasta la reproducción multimedia y la creación de gráficos 3D.

C. Gráficos y Visualización: Más Allá del Texto

- **SVG (Scalable Vector Graphics):** Un formato de imagen basado en vectores que describe gráficos como conjuntos de formas geométricas. Su ventaja clave es la escalabilidad sin pérdida de calidad, lo que los hace ideales para logotipos e ilustraciones responsivas.
- **WebGL:** Una API de JavaScript que permite renderizar gráficos 2D y 3D de alto rendimiento directamente en el navegador, aprovechando la capacidad de la tarjeta gráfica del dispositivo.

II. ¿Cómo Funciona la Web?

Comprender el funcionamiento interno de la web implica conocer cómo los diferentes componentes se interconectan para entregar contenido.

A. Elementos Clave en la Comunicación Web

- **DNS (Domain Name System):** Actúa como la "agenda telefónica" de Internet, traduciendo los nombres de dominio legibles por humanos (ej., google.com) a direcciones IP numéricas (ej., 172.217.160.142), que son las que realmente utilizan los dispositivos para comunicarse.
- **TCP/IP (Transmission Control Protocol/Internet Protocol):** El conjunto fundamental de protocolos que rige la comunicación en Internet. Permite que las computadoras intercambien datos de manera confiable, dividiéndose en capas con funciones específicas:
 - **Capa de Enlace:** Gestiona el acceso a los medios físicos de la red.
 - **Capa de Internet:** Se encarga del enrutamiento de paquetes de datos (IP).
 - **Capa de Transporte:** Asegura la entrega de datos de extremo a extremo (TCP para conexiones confiables, UDP para transmisiones rápidas).
 - **Capa de Aplicación:** Define el significado y formato de los mensajes intercambiados por las aplicaciones (donde opera HTTP).

B. Componentes Físicos y Lógicos

El funcionamiento de la web involucra la interacción de:

- **Cliente:** El dispositivo del usuario (navegador web).
- **Servidor:** La computadora que almacena y entrega los archivos web.
- **Conexión a Internet:** La infraestructura de red que permite la comunicación.
- **DNS:** Para la resolución de nombres de dominio.
- **Protocolo TCP/IP:** Para la comunicación subyacente.
- **HTTP:** Para la transferencia de contenido web.
- **Archivos Web:** El contenido real (HTML, CSS, JavaScript, imágenes, etc.).

III. Arquitectura de Aplicaciones para Internet

Las aplicaciones web modernas se dividen conceptualmente en dos grandes áreas.

- **Backend (Servidores):** La parte de la aplicación que reside en el servidor. Se encarga de la lógica de negocio, el procesamiento de datos, la interacción con bases de datos, la autenticación de usuarios y otras funcionalidades que no son directamente visibles para el usuario.
- **Frontend (Clientes):** La parte de la aplicación que interactúa directamente con el usuario en su navegador. Incluye la interfaz de

usuario (UI), el diseño visual y toda la lógica que se ejecuta en el lado del cliente para proporcionar una experiencia interactiva.

IV. Roles en el Desarrollo de Aplicaciones Web

El desarrollo de una aplicación web es un esfuerzo colaborativo que involucra diversas especialidades.

- **Desarrollador Frontend:** Se enfoca en la interfaz de usuario y la experiencia del cliente, utilizando HTML, CSS, JavaScript y frameworks asociados.
- **Desarrollador Backend:** Se especializa en la lógica del servidor, las bases de datos y la infraestructura subyacente.
- **Otros Roles Cruciales:**
 - **Gestor de Proyecto:** Supervisa y coordina todo el proceso de desarrollo.
 - **Arquitecto de Información:** Diseña la estructura y organización del contenido.
 - **Diseñador Gráfico:** Crea los elementos visuales y la estética de la interfaz.
 - **Especialista en Usabilidad y Experiencia de Usuario (UX/UI):** Se asegura de que la aplicación sea intuitiva y agradable de usar.
 - **Administrador de Bases de Datos:** Gestiona el almacenamiento y recuperación de la información.
 - **Redactor y Editor de Contenidos:** Crea y optimiza el texto de la aplicación.
 - **Especialista en Seguridad de la Información:** Protege la aplicación contra vulnerabilidades.

V. Desarrollo Frontend y Backend en Detalle

A. Lado del cliente - Frontend

El desarrollo del lado del cliente abarca diferentes tipos de aplicaciones.

- **Aplicaciones Web:** Basadas en navegadores, utilizan las tecnologías web estándar.
- **Aplicaciones de Escritorio:** Desarrolladas para sistemas operativos específicos:
 - **Ventanas:** C#,VB.Net,J# con Visual Studio .Neto.
 - **Linux:**DO, C++ con Atom + compilador.
 - **Multiplataforma (Java):**Java con Netbeans.

- **Androide:**Java,C++ con Android Studio.
- **iOS:**Objective-C con X-code.
- **Herramientas y Tecnologías Comunes:**
 - **VCS (Sistema de control de versiones):** Sistemas como Git para gestionar cambios en el código.
 - **Gestores de paquetes:** Herramientas para gestionar librerías y dependencias.
 - **Herramientas de construcción:** Automatizan tareas de desarrollo (compilación, optimización).
 - **Especificaciones del idioma:** Estándares que definen cómo funcionan los lenguajes.
 - **Bibliotecas/Marcos de interfaz de usuario:** Componentes y estructuras predefinidas para construir interfaces de usuario (ej.,Reaccionar,Angular,Vista).
 - **Marcos CSS:** Facilitan el diseño responsivo (ej.,Oreja,Viento de cola CSS).
 - **Preprocesadores CSS:** Extienden las capacidades de CSS (ej.,HABLAR CON DESCARO A,MENOS).
 - **CSS3:** Última especificación de CSS con nuevas propiedades, elementos y soporte multimedia.
 - **Lienzo:** Elemento HTML5 para gráficos dinámicos y animaciones.
 - **Formularios Web:** Elementos interactivos para la entrada de datos por parte del usuario.
 - **Geolocalización:** Acceso a la ubicación del usuario.
 - **Aplicaciones Web Offline:** Permiten la funcionalidad sin conexión a internet.
 - **Herramientas y Frameworks JS:** Ecosistema rico para el desarrollo de JavaScript.

B. Lado del servidor - Backend

El backend es el motor detrás de la aplicación.

- **Servidores web:** Programas que atienden las solicitudes de los clientes (ej.,Apache,Nginx).
- **Lenguajes del Lado Backend:** Lenguajes de programación utilizados para la lógica del servidor (ej.,Pitón,Nodo.js,PHP,Rubí,Java).
- **Gestores de Base de Datos:** Sistemas para almacenar, organizar y recuperar datos (ej.,MySQL,PostgreSQL,Base de datos MongoDB).

VI. Diseño UX/UI: Experiencia y Usabilidad

El diseño UX/UI es fundamental para crear productos digitales exitosos.

- **UI (User Interface - Interfaz de Usuario):** Se refiere al diseño visual y la interacción de la aplicación. Su objetivo es crear una interfaz intuitiva, atractiva y fácil de usar. Las herramientas comunes para crear prototipos visuales incluyen Figma, Adobe XD, Gliffy, Balsamiq y Gravedad.
- **UX (User Experience - Experiencia de Usuario):** Aborda la percepción y sentimientos generales del usuario al interactuar con un producto o servicio. Busca asegurar que la interacción sea eficiente, satisfactoria y agradable.
- **Objetivo Conjunto del Diseño UX/UI:** Entregar soluciones digitales que no solo sean visualmente atractivas, sino también intuitivas, eficientes y que brinden una experiencia gratificante al usuario.
- **Figma:** Una popular herramienta de diseño web y de aplicaciones basada en la nube que permite la colaboración en tiempo real, facilitando el trabajo en equipo en el diseño de interfaces.

VII. Glosario de Términos Clave

- **HTML (lenguaje de marcado de hipertexto):** Lenguaje estándar para la creación de páginas web, definiendo su estructura y contenido.
- **CSS (hojas de estilo en cascada):** Lenguaje utilizado para describir la presentación visual de un documento HTML, controlando el estilo y la disposición.
- **HTTP (Protocolo de transferencia de hipertexto):** Protocolo para la transferencia de documentos hipermedia en la World Wide Web.
- **JavaScript:** Lenguaje de programación que permite la interactividad y funcionalidades dinámicas en las páginas web ejecutándose en el navegador.
- **API web (interfaces de programación de aplicaciones web):** Conjuntos de funciones y protocolos que permiten a las aplicaciones web interactuar con diversas funcionalidades del navegador y servicios externos.
- **SVG (Gráficos vectoriales escalables):** Formato de imagen vectorial que mantiene la calidad sin importar el tamaño de escalado.
- **WebGL:** API de JavaScript para renderizar gráficos 2D y 3D en el navegador web.
- **DNS (Sistema de nombres de dominio):** Sistema que convierte nombres de dominio legibles en direcciones IP.

- **TCP/IP (Protocolo de control de transmisión/Protocolo de Internet):** El conjunto de protocolos que forman la base de Internet para la comunicación de datos.
- **Interfaz:** La parte de una aplicación web con la que el usuario interactúa directamente en el navegador.
- **Backend:** La parte de una aplicación web que se ejecuta en el servidor, manejando la lógica de negocio y la base de datos.
- **UX (Experiencia de usuario):** La experiencia general y la percepción de un usuario al interactuar con un producto o servicio.
- **UI (Interfaz de usuario):** El diseño visual y los elementos interactivos con los que el usuario interactúa.

VIII. Procedimientos y Prácticas

- **Creación de Portafolio Electrónico en GitHub:** Los estudiantes deben documentar su proceso de aprendizaje y proyectos en un sitio web alojado en GitHub, incluyendo descripciones de temas, ejercicios de laboratorio, resultados y reflexiones.
- **Desarrollo Web (Proceso):** Implica la combinación de HTML para la estructura, CSS para el estilo, JavaScript para la interactividad, Web APIs para funcionalidades avanzadas, y otras tecnologías web relevantes.
- **Práctica de Laboratorio 01 con Visual Studio Code:** El uso de Emmet para codificación rápida y la creación de fragmentos de código personalizados.