**Programacion Web**

**Profesor: Rodrigo Fidel Gaxiola**

|  |  |
| --- | --- |
|  | **Nombre del Participante** |
|  |
| Iván Eduardo Vivas Chan |
|
|  |  |
|  | **# Actividad** |
|  | Investigación 3er Parcial |
|  |  |
|  | **Fecha de entrega** |
|  | 26-11-2024 |

**.**

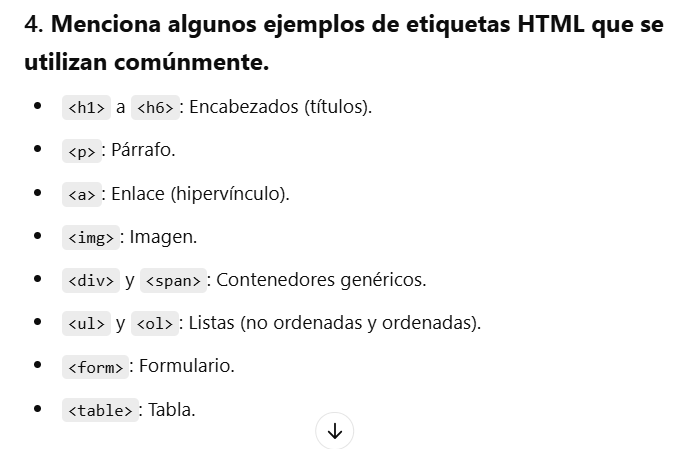
1. **La programación web se refiere al desarrollo de sitios web o aplicaciones web. Implica escribir el código que define el diseño, la funcionalidad y las interacciones de una página o sistema en línea. Combina tecnologías del lado del cliente (frontend) y del lado del servidor (backend) para ofrecer experiencias en la web.**
2. **¿Cuál es la diferencia entre el lado del cliente (frontend) y el lado del servidor (backend) en el desarrollo web?**

**Frontend: Es la parte visual e interactiva que los usuarios ven y con la que interactúan. Se desarrolla con tecnologías como HTML, CSS y JavaScript.**

**Backend: Es la lógica detrás del funcionamiento de un sitio web, incluyendo bases de datos, servidores y APIs. Se desarrolla con lenguajes como Python, PHP, Node.js, entre otros. El backend procesa datos y envía resultados al frontend.**

1. **¿Qué es HTML y para qué se utiliza en la programación web?**

**HTML (HyperText Markup Language) es el lenguaje estándar para estructurar el contenido de una página web. Define elementos como títulos, párrafos, imágenes, enlaces, tablas y más. Es esencial para crear la base de cualquier sitio web.**

****

### **5 - ¿Qué es CSS y cómo se relaciona con HTML?**

CSS (Cascading Style Sheets) es el lenguaje utilizado para dar estilo a los documentos HTML. Permite definir colores, fuentes, márgenes, diseño de página y más. CSS separa la presentación visual del contenido estructurado por HTML.

### **6- ¿Cuál es el propósito de JavaScript en el desarrollo web?**

JavaScript es un lenguaje de programación que permite agregar interactividad y dinamismo a los sitios web. Por ejemplo, validación de formularios, animaciones, actualización de contenido sin recargar la página (AJAX), entre otros.

### **7- ¿Cuál es la diferencia entre un sitio web estático y uno dinámico?**

* **Estático**: El contenido no cambia a menos que se modifique manualmente el código. Generalmente utiliza solo HTML y CSS.
* **Dinámico**: El contenido cambia según las interacciones del usuario o los datos recibidos del servidor. Requiere backend y lenguajes como PHP o JavaScript (Node.js).

### **8- ¿Qué es un navegador web y cómo interpreta HTML, CSS y JavaScript?**

Un navegador web es un software (como Chrome, Firefox o Edge) que permite a los usuarios acceder y visualizar sitios web. Los navegadores:

1. Interpretan el código HTML para estructurar el contenido.
2. Aplican estilos definidos en CSS.
3. Ejecutan JavaScript para agregar interactividad y dinamismo

### **9- ¿Qué es el diseño responsive en el contexto de la programación web?**

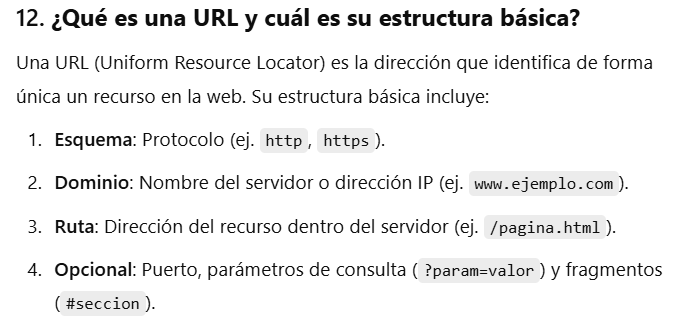
El diseño responsive (adaptativo) asegura que los sitios web se vean y funcionen correctamente en dispositivos con diferentes tamaños de pantalla, como computadoras, tablets y teléfonos. Se logra con CSS mediante técnicas como media queries, diseño flexible y uso de unidades relativas.

### **10- ¿Cuál es el objetivo de la optimización para motores de búsqueda (SEO) en un sitio web?**

El SEO (Search Engine Optimization) busca mejorar la visibilidad de un sitio web en los resultados de búsqueda. Esto incluye optimizar el contenido, las palabras clave, la velocidad de carga y la estructura del sitio para atraer más visitantes y mejorar el ranking en motores como Google.

### **11- Describe brevemente el proceso de solicitud y respuesta en el modelo cliente-servidor.**

El cliente (por ejemplo, un navegador web) envía una solicitud al servidor a través de la red, generalmente usando el protocolo HTTP. El servidor procesa la solicitud (como entregar una página web o realizar operaciones en una base de datos) y envía una respuesta al cliente, que contiene los datos solicitados, como un archivo HTML, JSON o imágenes.



### **13- Menciona algunos de los lenguajes de programación utilizados en el lado del servidor.**

* **PHP**
* **Python** (usando frameworks como Django o Flask)
* **JavaScript** (con Node.js)
* **Ruby** (con Ruby on Rails)
* **Java** (con Spring o JSP)
* **C#** (usando ASP.NET).

### **14- ¿Qué es un framework en el desarrollo web y cómo puede beneficiar a los desarrolladores?**

Un framework es un conjunto de herramientas, bibliotecas y convenciones predefinidas que facilitan el desarrollo web al proporcionar soluciones listas para tareas comunes (como manejo de rutas, autenticación o plantillas). Beneficios:

* Acelera el desarrollo.
* Promueve buenas prácticas.
* Simplifica tareas repetitivas.
* Proporciona mayor seguridad y soporte comunitario.

### **15 - Explica el concepto de "modelo de caja" (box model) en CSS.**

El box model describe cómo se estructuran los elementos HTML. Cada elemento se compone de:

1. **Contenido**: Texto, imágenes o elementos dentro.
2. **Relleno (padding)**: Espacio entre el contenido y el borde.
3. **Borde (border)**: Línea que rodea el contenido y el relleno.
4. **Margen (margin)**: Espacio fuera del borde, separando el elemento de otros.

### **16- ¿Qué son las cookies y para qué se utilizan en la programación web?**

Las cookies son pequeños archivos de texto almacenados en el navegador del usuario. Se utilizan para:

* Almacenar datos de sesión (ej. autenticación).
* Recordar preferencias del usuario (idioma, tema).
* Rastrear actividades para análisis o personalización.
* Implementar carritos de compras en e-commerce.

### **17- ¿Cuál es la importancia de la seguridad en el desarrollo web?**

La seguridad protege a los usuarios y sistemas de amenazas como:

* Robo de datos personales o financieros.
* Ataques comunes (inyección SQL, cross-site scripting - XSS).
* Acceso no autorizado.  
  Implementar buenas prácticas, como cifrado, validación de entradas y protección contra vulnerabilidades, es esencial para evitar daños a los usuarios y a la reputación de un sitio.

### **18- ¿Qué es control de versiones y cómo puede ayudar en el desarrollo web colaborativo?**

El control de versiones (ej. Git) permite gestionar y rastrear cambios en el código. Beneficios:

* Registro histórico de cambios.
* Colaboración simultánea entre desarrolladores.
* Facilita revertir errores.
* Permite trabajar en múltiples versiones o ramas del proyecto.

### **19- ¿Qué es un dominio y cómo se relaciona con una dirección IP?**

Un dominio es un nombre legible para los humanos que identifica un recurso en Internet (ej. google.com). Se relaciona con una dirección IP porque los servidores DNS traducen el dominio en una dirección IP, que es el identificador único del servidor donde se aloja el sitio.

### **20- Describe brevemente cómo funciona el protocolo HTTP en la navegación web.**

HTTP (HyperText Transfer Protocol) es un protocolo que define cómo se comunican los navegadores y los servidores.

1. El cliente envía una **solicitud HTTP** (GET, POST, etc.) al servidor.
2. El servidor procesa la solicitud y devuelve una **respuesta HTTP**, que incluye un código de estado (ej. 200 OK, 404 Not Found) y el contenido solicitado.
3. El navegador interpreta la respuesta y muestra el contenido al usuario.

## 21- Evolución histórica de las aplicaciones web modernas

Las aplicaciones web han recorrido un largo camino desde sus inicios. Aquí te presento un resumen de los hitos más importantes:

* **Los inicios: HTML estático:** Las primeras páginas web eran documentos HTML estáticos, con poca interactividad. La información se presentaba de forma lineal y no existía la posibilidad de realizar acciones dinámicas.
* **La llegada de los lenguajes del lado del servidor:** Con la aparición de lenguajes como PHP, ASP y Perl, las páginas web se volvieron más dinámicas. Se podían generar contenidos personalizados en función de las acciones del usuario, como por ejemplo mostrar diferentes productos según las preferencias del visitante.
* **El auge de JavaScript:** JavaScript revolucionó la experiencia del usuario al permitir crear interfaces interactivas directamente en el navegador. Esto dio lugar a aplicaciones web más ricas y complejas.
* **El surgimiento de las aplicaciones web de una sola página (SPA):** Las SPA cargan todo el contenido de una aplicación web en una sola página, lo que proporciona una experiencia más similar a las aplicaciones de escritorio.
* **La adopción de frameworks y librerías:** Frameworks como Angular, React y Vue.js simplificaron el desarrollo de aplicaciones web al proporcionar componentes preconstruidos y herramientas para gestionar la complejidad.
* **El cloud computing y los microservicios:** La nube permitió escalar las aplicaciones web de forma más fácil y económica, mientras que los microservicios permitieron descomponer las aplicaciones en servicios más pequeños y manejables.



## 23- Tecnologías y servicios que impulsaron el avance de las aplicaciones web

* **Lenguajes de programación:** JavaScript, PHP, Python, Ruby, etc.
* **Frameworks y librerías:** Angular, React, Vue.js, jQuery, Bootstrap, etc.
* **Bases de datos:** MySQL, PostgreSQL, MongoDB, etc.
* **Servidores web:** Apache, Nginx, etc.
* **Cloud computing:** Amazon Web Services (AWS), Google Cloud Platform (GCP), Microsoft Azure, etc.
* **Contenedores:** Docker
* **Microservicios:** Arquitectura basada en microservicios
* **APIs:** Interfaces de programación de aplicaciones
* **Protocolos de comunicación:** HTTP, WebSocket, etc.

**En resumen,** las aplicaciones web han evolucionado de simples páginas estáticas a aplicaciones complejas y dinámicas gracias a avances en lenguajes de programación, frameworks, infraestructura en la nube y otros factores. Su accesibilidad y facilidad de distribución las han convertido en la forma más popular de software en la actualidad.

**24- La arquitectura de tres capas** divide una aplicación en tres niveles lógicos:

1. **Presentación:** La interfaz de usuario, que interactúa directamente con el usuario.
2. **Lógica de negocio:** La parte que contiene las reglas de negocio y las operaciones de la aplicación.
3. **Datos:** Donde se almacenan los datos de la aplicación.

**¿Cómo se comunican estos niveles?**

La comunicación principal se establece entre el cliente (navegador web) y el servidor (donde se ejecuta la lógica de negocio y se almacenan los datos). Esta comunicación se realiza principalmente a través de **solicitudes HTTP** y **respuestas HTTP**.

**Proceso típico de una solicitud:**

1. **El usuario interactúa con la interfaz:** El usuario hace clic en un enlace, llena un formulario o realiza cualquier otra acción en la interfaz.
2. **El navegador envía una solicitud HTTP:** El navegador envía una solicitud HTTP al servidor. Esta solicitud contiene información sobre la acción que el usuario ha realizado (por ejemplo, la URL solicitada, los datos de un formulario, etc.).
3. **El servidor procesa la solicitud:** El servidor recibe la solicitud, la analiza y ejecuta la lógica de negocio correspondiente. Esto puede implicar consultar una base de datos, realizar cálculos, o llamar a otros servicios.
4. **El servidor envía una respuesta HTTP:** El servidor envía una respuesta HTTP al navegador. Esta respuesta contiene el contenido que debe ser mostrado en el navegador (por ejemplo, una página HTML, un archivo JSON, etc.).
5. **El navegador renderiza la respuesta:** El navegador recibe la respuesta y la renderiza en la pantalla del usuario, actualizando la interfaz de usuario según sea necesario.

**Protocolos y tecnologías involucradas:**

* **HTTP (Hypertext Transfer Protocol):** Es el protocolo fundamental para la comunicación entre el cliente y el servidor. Define cómo se envían y reciben las solicitudes y respuestas.
* **HTTPS (Hypertext Transfer Protocol Secure):** Es una versión segura de HTTP que utiliza el cifrado SSL/TLS para proteger la comunicación entre el cliente y el servidor.
* **HTML 1 (HyperText Markup Language):** Es el lenguaje utilizado para estructurar el contenido de las páginas web.

[1. loscreativos.co](https://loscreativos.co/marketing-digital/que-es-http-definicion-caracteristicas/" \t "_blank)

[loscreativos.co](https://loscreativos.co/marketing-digital/que-es-http-definicion-caracteristicas/" \t "_blank)

* **CSS (Cascading Style Sheets):** Se utiliza para dar estilo y diseño a las páginas web.
* **JavaScript:** Permite crear interfaces de usuario interactivas y manipular el contenido de las páginas web.
* **JSON (JavaScript Object Notation):** Es un formato de intercambio de datos ligero y de fácil lectura que se utiliza comúnmente para transmitir datos entre el servidor y el cliente.

**Ejemplo:**

Cuando un usuario ingresa una dirección web en la barra de direcciones de su navegador, el navegador envía una solicitud HTTP al servidor web. El servidor procesa la solicitud, consulta la base de datos para obtener la página solicitada y envía una respuesta HTTP al navegador, que incluye el código HTML de la página. El navegador interpreta el código HTML y muestra la página en la pantalla.

**En resumen:**

La comunicación entre el cliente y el servidor en una arquitectura web de tres capas se basa en el envío de solicitudes HTTP y la recepción de respuestas HTTP. Este proceso permite a los usuarios interactuar con las aplicaciones web de forma dinámica y obtener la información

## 28. HTML5 y sus mejoras

**HTML5** es la última versión del lenguaje de marcado de hipertexto, el estándar para estructurar el contenido de las páginas web. Introdujo una serie de mejoras significativas con respecto a versiones anteriores, como:

* **Nuevas etiquetas semánticas:** HTML5 incorpora etiquetas más semánticas, lo que facilita la comprensión del contenido por parte de los motores de búsqueda y de los usuarios con discapacidad. Ejemplos de estas etiquetas son <header>, <nav>, <section>, <article>, <footer>, etc.
* **Multimedia integrado:** Permite la inclusión directa de elementos multimedia como audio y video sin necesidad de plugins externos, mejorando la experiencia del usuario.
* **Formularios mejorados:** Ofrece nuevos tipos de entrada para formularios, como <input type="email"> o <input type="number">, lo que facilita la validación de datos y la creación de formularios más intuitivos.
* **API de almacenamiento local:** Permite a las aplicaciones web almacenar datos localmente en el navegador del usuario, mejorando la experiencia del usuario al permitir la creación de aplicaciones más ricas y offline.
* **Geolocalización:** Permite a las aplicaciones web determinar la ubicación geográfica del usuario, lo que abre la puerta a nuevas aplicaciones y servicios.
* **Canvas:** Proporciona un lienzo 2D para dibujar gráficos y animaciones directamente en el navegador.
* **WebSockets:** Permite una comunicación bidireccional en tiempo real entre el cliente y el servidor, lo que es fundamental para aplicaciones como chats y juegos en línea.

## 29. Importancia de CSS

**CSS (Cascading Style Sheets)** es un lenguaje de estilo utilizado para describir la presentación de un documento HTML. Su principal función es separar la estructura del contenido (HTML) de su presentación visual. Esto ofrece múltiples ventajas:

* **Mayor mantenimiento:** Al separar el contenido del diseño, es más fácil realizar cambios en la apariencia de una página web sin afectar su estructura.
* **Reutilización de estilos:** Los estilos CSS pueden ser reutilizados en múltiples páginas, lo que ahorra tiempo y esfuerzo.
* **Mejor accesibilidad:** CSS permite personalizar la apariencia de una página web para diferentes dispositivos y usuarios, mejorando la accesibilidad.
* **Diseño responsivo:** CSS3, la última versión de CSS, incluye características como media queries que permiten crear diseños que se adaptan automáticamente a diferentes tamaños de pantalla.

## 30. Lenguajes de programación del lado del servidor

Los lenguajes de programación del lado del servidor se utilizan para procesar las solicitudes del cliente, acceder a bases de datos y generar el contenido dinámico de una página web. Algunos de los más utilizados son:

* **PHP:** Es uno de los lenguajes más populares para el desarrollo web, especialmente para crear sitios web dinámicos y aplicaciones web. Es fácil de aprender y tiene una gran comunidad de desarrolladores.
* **Python:** Es un lenguaje versátil y de alto nivel que se utiliza para una amplia variedad de tareas, incluyendo el desarrollo web. Frameworks como Django y Flask facilitan la creación de aplicaciones web robustas y escalables.
* **Node.js:** Permite ejecutar JavaScript fuera del navegador, lo que lo hace ideal para crear aplicaciones web en tiempo real y servidores de aplicaciones.
* **Java:** Es un lenguaje de programación orientado a objetos muy utilizado para desarrollar aplicaciones empresariales a gran escala. Frameworks como Spring facilitan el desarrollo de aplicaciones web en Java.

**Contextos en los que son preferibles:**

* **PHP:** Ideal para proyectos de tamaño mediano a grande, CMS (sistemas de gestión de contenido) como WordPress, y aplicaciones web dinámicas.
* **Python:** Excelente para proyectos que requieren un alto nivel de abstracción, ciencia de datos, aprendizaje automático y aplicaciones web escalables.
* **Node.js:** Perfecto para aplicaciones en tiempo real, APIs RESTful, aplicaciones de una sola página (SPA) y microservicios.
* **Java:** Apto para grandes empresas, aplicaciones empresariales, sistemas distribuidos y aplicaciones que requieren un alto rendimiento y seguridad.

## 31. Youtubers/Tiktokers para aprender desarrollo web

**En español:**

* **Soy Dalto:** Uno de los más populares, con una gran variedad de tutoriales desde lo básico hasta temas más avanzados.
* **FreeCodeCamp en español:** Ofrece cursos completos y gratuitos sobre desarrollo web.
* **Traversy Media:** Aunque es en inglés, tiene una gran cantidad de subtítulos en español y cubre una amplia gama de temas.

**En inglés:**

* **Traversy Media:** Ya mencionado, es una excelente opción para aquellos que se sienten cómodos con el inglés.
* **The Net Ninja:** Ofrece tutoriales claros y concisos sobre una variedad de tecnologías web.
* **Brad Traversy:** Cubre temas desde lo básico hasta lo avanzado, con un enfoque práctico y directo.

**Importante:** Esta lista es solo una pequeña muestra de los muchos creadores de contenido disponibles en YouTube y TikTok. Te recomiendo explorar y encontrar los canales que mejor se adapten a tu estilo de aprendizaje y a tus intereses.

## 32. Planificación de aplicaciones web: diseño y desarrollo

La planificación de una aplicación web implica definir todos los aspectos del proyecto desde el inicio, desde la concepción de la idea hasta su implementación final. Esto incluye:

* **Diseño:**
  + **Diseño de la interfaz de usuario (UI):** Creación de wireframes, prototipos y maquetas para visualizar la apariencia y la interacción del usuario con la aplicación.
  + **Diseño de la experiencia de usuario (UX):** Definición de los flujos de usuario, la navegación y la interacción general para garantizar una experiencia intuitiva y satisfactoria.
* **Desarrollo:**
  + **Arquitectura de la aplicación:** Definición de la estructura general de la aplicación, incluyendo las tecnologías a utilizar, los componentes y la forma en que se comunicarán entre sí.
  + **Base de datos:** Diseño de la estructura de la base de datos para almacenar la información de la aplicación de manera eficiente.
  + **Desarrollo del backend:** Creación de la lógica del servidor que gestionará las solicitudes de los usuarios y realizará las operaciones necesarias.
  + **Desarrollo del frontend:** Creación de la interfaz de usuario utilizando tecnologías como HTML, CSS y JavaScript.

## 33. Metodologías ágiles

Las metodologías ágiles son enfoques de desarrollo de software que se centran en la flexibilidad, la colaboración y la entrega iterativa de software. Algunas de las más populares son:

* **Scrum:** Divide el proyecto en sprints cortos (iteraciones) y utiliza roles como el Product Owner, el Scrum Master y el equipo de desarrollo.
* **Kanban:** Visualiza el flujo de trabajo utilizando un tablero y tarjetas, permitiendo una gestión flexible y transparente.
* **Extreme Programming (XP):** Se enfoca en la calidad del código, la comunicación y la retroalimentación continua.

## 34. Importancia de considerar el público objetivo

Conocer a tu público objetivo es fundamental para crear una aplicación web exitosa. Esto permite:

* **Personalizar la experiencia del usuario:** Adaptar la interfaz, el contenido y las funcionalidades a las necesidades y preferencias de los usuarios.
* **Definir las características clave:** Priorizar las funcionalidades que más valorarán los usuarios.
* **Crear un mensaje claro y atractivo:** Comunicar los beneficios de la aplicación de manera efectiva

## 35. Pasos esenciales en la planificación de la arquitectura

* **Definición de los requisitos:** Identificar las funcionalidades y características que debe tener la aplicación.
* **Selección de la tecnología:** Elegir las tecnologías adecuadas para el frontend, backend y la base de datos.
* **Diseño de la base de datos:** Crear un modelo de datos que represente la información de la aplicación.
* **Diseño de la arquitectura:** Definir la estructura de la aplicación y cómo se comunicarán los diferentes componentes.
* **Planificación de la seguridad:** Implementar medidas de seguridad para proteger los datos de los usuarios.

## 36. Importancia de una estructura de carpetas y archivos organizada

Una estructura de carpetas y archivos bien organizada es esencial para:

* **Facilitar la colaboración:** Permite a varios desarrolladores trabajar en el proyecto de forma eficiente.
* **Mejorar la mantenibilidad:** Hace que el código sea más fácil de entender y modificar.
* **Acelerar el desarrollo:** Reduce el tiempo de búsqueda de archivos y facilita la localización de errores.
* **Promover buenas prácticas de desarrollo:** Fomenta la creación de un código limpio y bien estructurado.

