****

Universidad Nacional de San Juan

Facultad de Ciencias Exactas, Físicas y Naturales

Departamento de Informática

Licenciatura en Ciencias de la Computación

Licenciatura en Sistemas de Información

**Práctico N° 1: Fundamentos de la Web y la World Wide Web**

Autor: Facundo Saavedra

2024

**Ejercicios**

1-Responde las siguientes preguntas:

a) ¿Qué es el Protocolo HTTP? Es el protocolo de comunicación utilizado para transferir datos en la web. Define cómo se envían y reciben solicitudes y respuestas entre los clientes y los servidores web

b) ¿Cuál es la diferencia entre URI y URL? Una URI, es una cadena de caracteres que identifica de manera única un recurso en la web. Las URL son un tipo de URI que especifica la ubicación de un recurso en Internet.

c) ¿Cuál es el propósito del lenguaje HTML? HTML es el lenguaje de marcado utilizado para crear páginas web. Con el cual se define la estructura y el contenido de una página mediante etiquetas y recursos que describen elementos presentes en la página.

d) ¿Qué función cumple el CSS en una página web? El lenguaje CSS permite dar estilo y diseño, como color, tamaño, tipografía y la disposición en la página a los elementos en páginas web creadas con HTML.

e) ¿Cuál es el rol del JavaScript en el desarrollo web?El propósito de javascript en el desarrollo web es el de agregar interactividad a las páginas web. Permite manipular el contenido de la página en tiempo real, responder a eventos del usuario y comunicarse con servidores web para cargar datos adicionales sin recargar la página completa.

f) ¿Qué es el D.O.M.? ¿Y para qué se utiliza? El D.O.M. es una representación en memoria de la estructura de una página web diseñada con HTML. Permite acceder y manipular los elementos de la página mediante programación.

2- Investiga y completa los siguientes eventos históricos de la Web:

1991: Berners-Lee lanzó el primer sitio web y el primer navegador web, con el concepto de World Wide Web (www) que era una forma de compartir y acceder a información en línea mediante hipervínculos y navegadores web, dando inicio a la era de la web moderna.

1993: Se lanzó el navegador Mosaic, uno de los primeros navegadores gráficos que permitió a los usuarios poder ver imágenes y texto en una misma interfaz

1995: Lanzamiento de Internet Explorer 1.0 de Microsoft, que incluyó en Windows este navegador, y comenzó a ganar terreno en el mercado a costa de Netscape navigator. Daba comienzo la llamada guerra de los navegadores.

1996: Lanzamiento de la primera versión de Google y el nacimiento del primer webmail de la mano de HoTMaiL. También se lanzó el primer motor de búsqueda importante, AltaVista.

2004:La creación de Facebook (thefacebook.com). Con el comienzo de la Web 2.0 este tipo de webs ya no eran estáticas, ni de sólo lectura y permitían, compartir e interactuarcon la información de una manera sencilla. Se crean y extienden sistemas como los blogs, chats, wikis o foros

2011: Ya en el marco de la web 3.0 donde las páginas web ya no sólo eran una forma de interactuar y compartir la información de manera sencilla, ahora, su objetivo era darle significado y enriquecer la experiencia del usuario, se lanzaron instagram y google+ que finalmente no lograria el éxito que si tuvieron otras redes sociales. Microsoft empieza a soportar HTML5 en su navegador Internet explorer

2012: Lanzamiento de Google Drive, servicio de almacenamiento en la nube, que ofreció integración con otras herramientas de Google.

2014: Aumento en el uso de HTTP/2 el trabajo en este protocolo avanzó significativamente. Prometió mejoras en la velocidad y eficiencia de la transferencia de datos en la web. Además se descubrió la vulnerabilidad conocida como Heartbleed en la biblioteca de criptografía OpenSSL, que afectó a una gran cantidad de sitios web y servicios en línea.

2019: Lanzamiento de Google Stadia, un servicio de streaming de videojuegos que permite a los usuarios jugar en la nube sin necesidad de hardware dedicado, marcando un avance importante en el juego en la nube.

Implementación del protocolo HTTP/3. Comienzo del despliegue de la tecnología móvil 5G

2022: Estando en la era de la web 5.0, este año se lanzó ChatGPT-3.5, versión mejorada de la anterior GPT-3. Avances en la Web3 (que incluye tecnologías como blockchain y criptomonedas) y el metaverso.

Ley de Servicios Digitales (DSA) y Ley de Mercados Digitales (DMA) en la Unión Europea.

3- Ingresa a la web <https://http.app/> y analiza la url <https://www.unsj.edu.ar/>

¿Qué datos obtenemos en?

* Server (¿Qué significa?) Muestra en qué servidor web se está manejando la solicitud, en el caso de la página de la UNSJ, **Server: Apache/2.4.61 (Debian)**. El servidor es Apache HTTP Server, en la versión 2.4.61 y que se está ejecutando en un sistema operativo Debian.
* Protocol (¿Qué significa?) **Protocol** [**HTTP 1.1**](https://http.dev/1.1?ref=http.app)**,** indica la versión del protocolo HTTP que están usando el cliente y el servidor para el intercambio de datos
* Request method (¿Porqué GET?) Se utiliza el método GET ya que solicita la recuperación de un recurso específico del servidor sin modificar el estado del recurso en el mismo.
* ¿Es http 1.0 o versión 2.0?. ¿Cuál es la diferencia? En este caso es http 1.1, la principal diferencia con la versión 2.0 es: La eficiencia de conexiones, la versión 2.0 permite utilizar una única conexión TCP para enviar varios flujos de datos a la vez permitiendo también el manejo de múltiples solicitudes/respuestas en una sola conexión, la versión 1.1 carga los recursos uno después de otro con conexiones persistentes.
* Puedes dar un ejemplo de una URL que trabaje con protocolo http 2.0.

Por ejemplo el caso de <https://www.netflix.com> usa protocolo http 2.0

4- Completa la siguiente tabla con las ideas principales.

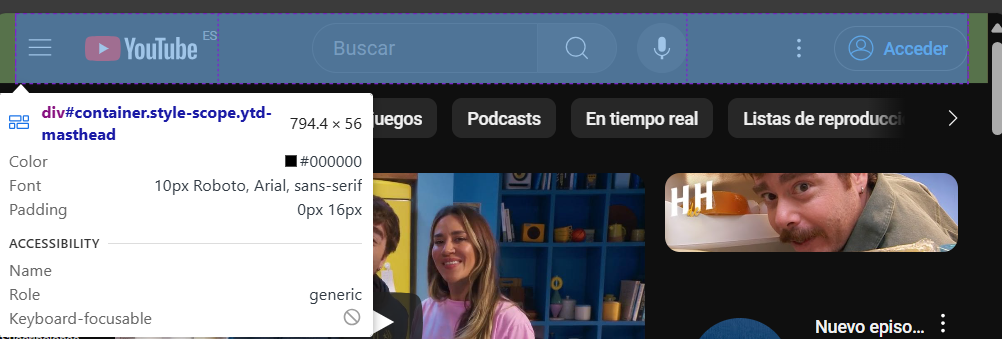
| CARACTERÍSTICA | WEB1.0 | WEB2.0 | WEB3.0 | WEB4.0 | WEB5.0 |
| --- | --- | --- | --- | --- | --- |
| Año de Inicio | 1990s | 2000 | 2010 | 2016 | 2022 |
| Principales Tecnologías | HTML estático, CGI, HTTP/1.0 | JavaScript, HTML5, CSS3, AJAX | Inteligencia artificial. Semantica web | IA avanzada, Machine learning. Deep learning | IoT, AI |
| Interactividad | estática | baja | alta | Muy Alta | Muy Alta |
| Contenido | estático | dinámico | Semántico | personalizado y relevante según el contexto | datos en tiempo real, Interfaces Neurales |
| Ejemplos de Sitios | yahoo, geocities | facebook, youtube, wikipedia | google, siri | chatGPT, google assistant | interfaces neuronales y metaverso |
| Enfoque Principal | Publicación de contenido | Colaboración e interacción del usuario | Datos Interconectados e Inteligencia | Experiencias Avanzadas e Inteligencia Artificial. | Experiencias Hiperpersonalizadas e Inmersivas. |

5- Analiza la siguiente URL, desglosar sus partes:

https://www.google.com.ar/search?q=url+con+%26+y+%23&sclient=gws-wiz#ip=1

* Protocolo:https
* Dominio: www.google.com.ar
* Ruta: /search
* Parámetros: “q=url+con+%26+y+%23”, “sclient=gws-wiz”
* Fragmento: #ip=1

6- Análisis de recursos web: Elige un sitio web popular y analiza cómo se utilizan los elementos fundamentales de la web, como HTML, CSS y JavaScript. Describe brevemente cómo se implementan estos elementos en el sitio y qué contribución hacen al diseño y funcionalidad de la página.

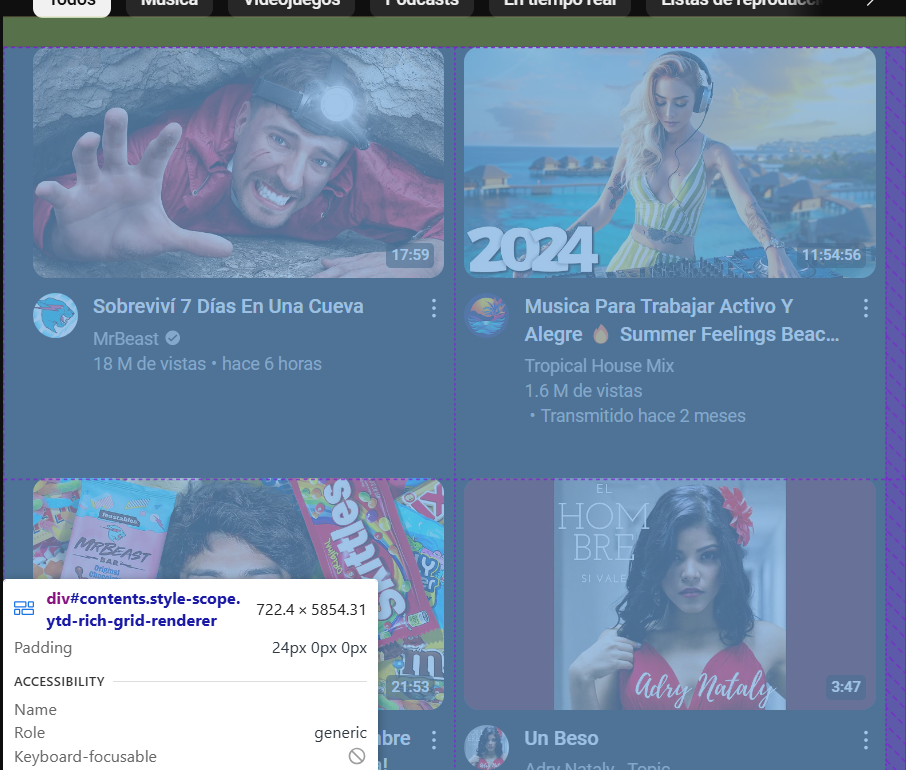
Analizando el caso de [YouTube](https://www.youtube.com/) podemos ver por ejemplo que: En la página principal se utiliza una etiqueta <div> para contener el encabezado principal que se muestra en toda la página y contiene el logo, la barra de búsqueda (para la cual usa una etiqueta <form> y un <input> para pedir la información de la búsqueda), varios botones más como el de notificaciones, abrir el panel lateral o acceder al perfil.

Luego podemos ver que todo el área principal donde se ubican los videos es otra etiqueta <div> ubicada con recursos de css como:

*padding-top: 24px;*

*display:flex;*

*flex-wrap:wrap;*

**

Donde cada video se renderiza al cargarse la página como otra etiqueta <div> ubicada con la herramienta grid de css para que todos los elementos estén bien ubicados y además el contenido se adapte a diferentes tamaños de pantallas

7- Análisis de seguridad web: Investiga y enumera las principales preocupaciones de seguridad en la web actual junto con una breve descripción de las mismas.

Principales preocupaciones de seguridad en la web:

1. **Inyección de SQL:** Un atacante inserta código SQL malicioso en una consulta para manipular la base de datos subyacente y permitir el acceso, modificación o eliminación no autorizada de datos.
2. **Cross-Site Request Forgery (CSRF):** El atacante engaña a un usuario autenticado para que realice una acción no deseada en una página web.
3. **Fugas de datos (Data Breaches):** Exposición no autorizada de datos sensibles debido a ataques, vulnerabilidades o malas prácticas de seguridad.
4. **Phishing:** Técnicas utilizadas para obtener información sensibles de los usuarios como datos financieros o credenciales de acceso, por medio de mensajes fraudulentos
5. **Ataques DoS y DDoS:** Ataques que buscan interrumpir o detener el servicio de un sitio web o aplicación al abrumar al servidor con una cantidad excesiva de tráfico de solicitudes al servidor.
6. **Malware y Ransomware:** Software malicioso diseñado para dañar sistemas, robar información o exigir un rescate recuperar el acceso a los datos bloqueados
7. **Vulnerabilidades en bibliotecas y software de terceros:** Vulnerabilidades presentes en software o bibliotecas de terceros utilizadas en el desarrollo de una página web

8- Comparativa de navegadores web: Elige dos navegadores web populares y realiza una comparativa de sus características y funcionalidades. Describe las ventajas y desventajas de cada navegador y menciona cualquier característica que los distinga.

|  | Microsoft Edge | Opera GX |
| --- | --- | --- |
| Ventajas | -Integración con windows -Rendimiento y velocidad -Herramientas avanzadas de seguridad y productividad con gran cantidad de opciones. -Actualizaciones constantes | -Diseño y personalización  -Bloqueador de anuncios integrado  -Funcionalidades integradas |
| Desventajas | -Personalización limitada  -Imposibilidad de poder desinstalarse fácilmente | -Compatibilidad con extensiones  -Desempeño en ciertos escenarios |
| Características especiales | -Asistente de IA incorporado e integrado con el sistema operativo windows (Copilot) | -Espacio para Gamers  -Modo juego  -Control de recursos (CPU, RAM y ancho de banda) para evitar que interfiera en el uso de otras aplicaciones |

9- Exploración de tecnologías emergentes: Investiga sobre las tecnologías emergentes en el campo de la web, como WebVR, WebRTC o Progressive Web Apps (PWA). Elige una de estas tecnologías y describe en qué consiste, qué beneficios ofrece y cómo podría impactar en el futuro de la web.

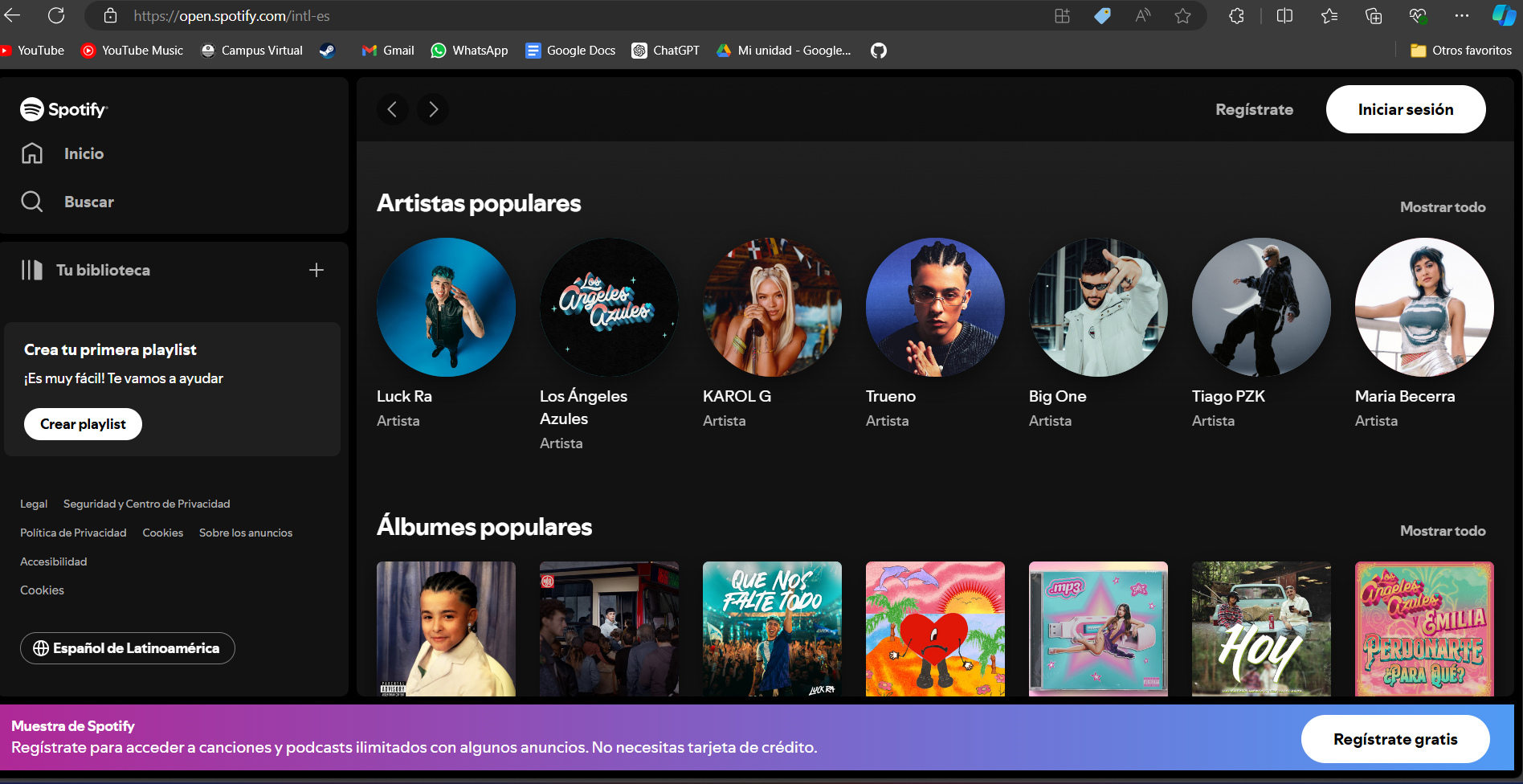
Actualmente la tecnología WebVR ha sido reemplazada por [WebXR Device API,](https://www.w3.org/TR/webxr/) que tiene un soporte más amplio, más funciones, mejor rendimiento y admite tanto VR como AR. WebVR es una especificación abierta que permite experimentar la realidad virtual en el navegador. Su propósito es hacer que la realidad virtual sea accesible a través de la web, sin necesidad de aplicaciones o software adicionales. Se usa en una página para mostrar contenido 3D (como vídeo de 360 grados, o un modelo 3D o un juego 3D) a través de lentes de realidad virtual o realidad aumentada. Por ejemplo tours virtuales (museos o propiedades inmobiliarias) o juegos VR.

Los beneficios que aporta tienen que ver com el acceso universal de este tipo de contenidos, sin necesidad de plugins y el desarrollo unificado ya que proporciona una API unificada para crear experiencias en VR.

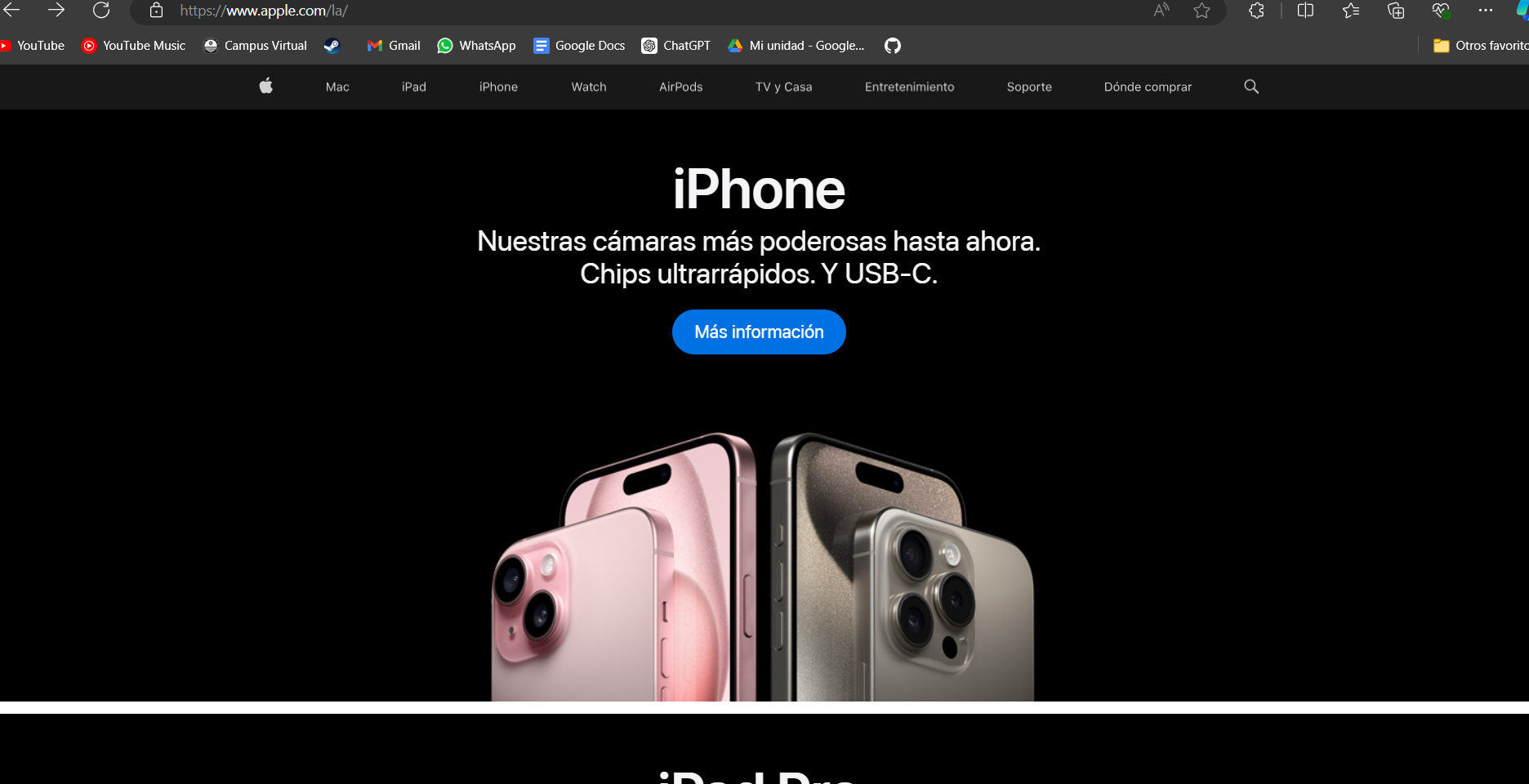
Esta tecnología podría tener un gran impacto en el ámbito de la educación, comercio interactivo, así como también el desarrollo de juegos y contenido multimedia, con la capacidad de crear experiencias más inmersivas.

10- Análisis de casos de éxito en la web: Investiga sobre proyectos web exitosos y analiza qué elementos contribuyeron a su éxito. Elige al menos dos casos y describe cómo se utilizaron los fundamentos de la web para lograr resultados destacados.

Primer caso, Spotify: Su éxito se debe principalmente a su enfoque en la experiencia de usuario. El sitio web está centrado en mostrar de manera clara y llamativa la música y la exploración de nuevas canciones, artistas y listas de reproducción.



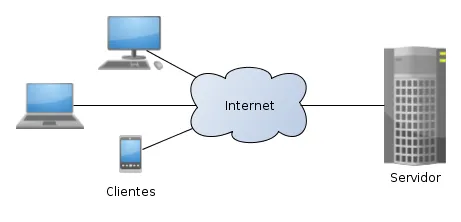
La página de inicio del sitio web está diseñada para ser visualmente agradable y fácil de navegar colores neutros y animaciones atractivas. También cuenta con secciones personalizadas para que los usuarios descubran nueva música basada en sus gustos y preferencias. El diseño del sitio es constante y similar en todas las plataformas, lo que permite a los usuarios disfrutar de la misma experiencia sin importar el dispositivo que utilicen. Del lado del backend, spotify utiliza los estándares más altos de seguridad informática y recursos de desarrollo web que permiten que la experiencia de usuario sea óptima y segura lo que incentiva y transmite tranquilidad a los usuarios de usar su servicio.

Segundo caso Apple: El gran éxito de la marca también se ve influenciado por el éxito de su página web, al igual que en sus productos, su sitio web se caracteriza por un minimalismo y simplicidad en la navegación excepcional. La información está organizada de manera intuitiva lo que facilita la búsqueda de los usuarios. La web cuenta con un sin fin de animaciones y transiciones utilizando CSS y JavaScript que hacen que la experiencia visual sea muy agradable al usuario.

Además cuenta con la capacidad de comprar los dispositivos y accesorios de la marca con absoluta seguridad de datos personales.

11-Investigación sobre la arquitectura cliente-servidor: Investiga y describe los conceptos fundamentales de la arquitectura cliente-servidor. Explica cómo funciona esta arquitectura en el contexto de la web y cuál es el papel del cliente y el servidor en la comunicación.

La arquitectura cliente-servidor es un modelo de diseño de software. Donde, las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes.



El cliente es la parte de las aplicaciones que realizan las peticiones de servicios hacia el servidor usando la red como mecanismo de comunicación. Del lado del cliente encontramos por ejemplo los navegadores web.

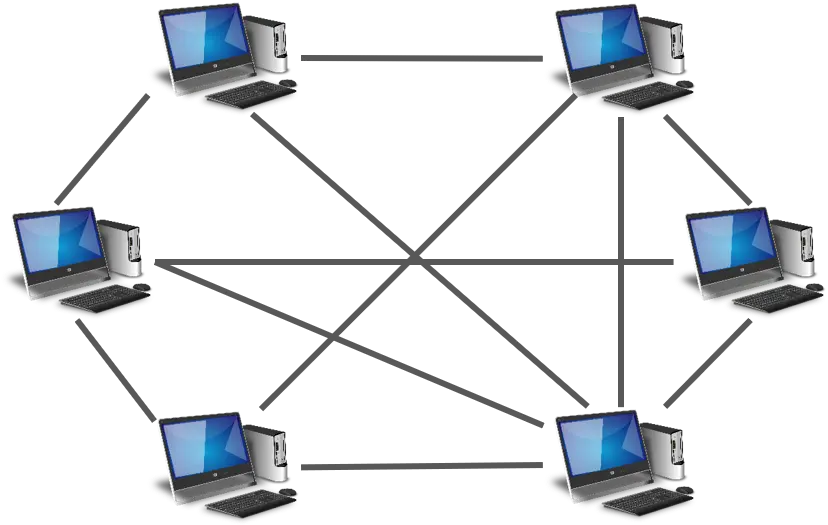
El servidor, es el componente que recibe, procesa las solicitudes de los clientes y proporciona los servicios o recursos. Puede consistir en diversos tipos de componentes de software instalados sobre un hardware. Entre los roles de los [servidores](https://www.arsys.es/servidores) se encuentran tareas de almacenamiento de información, procesamiento de datos.

El funcionamiento de esta arquitectura se basa en la comunicación de ambas mediante distintos protocolos como pueden ser HTTP, FTP, SMTP, entre otros. Y sigue los siguientes pasos:

1. El cliente realiza la solicitud (request) de un recurso o un servicio al servidor, generalmente por HTTP o HTTPS.
2. El servidor recibe la solicitud y procesa la entrada de datos enviada por el cliente, componiendo una respuesta.
3. Esa respuesta (response) se envía al cliente, esta respuesta suele incluir el código de estado HTTP (como 200 OK, 404 Not Found),

12-Comparativa de arquitecturas: Compara la arquitectura cliente-servidor con otras arquitecturas de comunicación, como la arquitectura de punto a punto (peer-to-peer). Describe las diferencias clave entre estas arquitecturas y menciona los casos de uso más apropiados para cada una.

La arquitectura Peer-to-peer (P2P) es una red de computadoras donde todos los dispositivos conectados a la red actúan como cliente y servidor al mismo tiempo. En esta arquitectura no es necesario un servidor central que administre la red.



Estas son algunas características de ambas arquitecturas a modo de comparativa:

| Cliente-Servidor | Peer-to-Peer |
| --- | --- |
| -Administración y actualizaciones centralizadas.  -Mayor control sobre la calidad y seguridad de los servicios ofrecidos  -Mejor rendimiento para aplicaciones que requieren procesamiento intensivo  -Dependencia de un servidor central, lo que lo hace vulnerable a fallos.  -Menos eficiente en términos de [escalabilidad](https://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/220). | -Menor dependencia de un servidor central, lo que reduce la vulnerabilidad a fallas únicas.  -Escalabilidad sin comprometer el rendimiento general.  -Compartir recursos y archivos de manera eficiente.  -Menor control sobre la calidad y legalidad de los contenidos compartidos.  -Puede ser menos eficiente para tareas que requieren alta velocidad y capacidad de procesamiento.  -Posible exposición a riesgos de seguridad y privacidad. |

13-Análisis del Protocolo HTTP: Describe en detalle el Protocolo HTTP (Hypertext Transfer Protocol). Explica cómo se establece una conexión HTTP, cómo se envían las solicitudes y respuestas, y qué tipo de información se puede transmitir a través de este protocolo.

HTTP (Hypertext Transfer Protocol) es un protocolo de comunicación entre aplicaciones basado en el intercambio de texto. Los navegadores web utilizan este protocolo para solicitar páginas web, imágenes, videos, entre otros a los servidores. Es un protocolo basado en el principio de cliente-servidor y opera sobre el protocolo TCP/IP.

El protocolo HTTP define una serie de códigos de estado**,**  cabeceras HTTP y métodos que especifican la acción que el cliente desea realizar sobre un recurso determinado en el servidor (GET, POST, DELETE, etc).

Para establecer la conexión, primero se realiza la Resolución de DNS, el cliente necesita conocer la dirección IP del servidor al que quiere conectarse. Una vez obtenida la IP, el cliente inicia una conexión TCP/IP con el servidor. Esto implica un proceso de tres pasos conocido como three-way handshake.

Una vez establecida la conexión, el cliente envía una solicitud HTTP al servidor. Esta contiene una línea de solicitud, encabezados opcionales, y en algunos casos, un cuerpo de mensaje.

Una solicitud típica de HTTP contiene:

1. Línea de solicitud: Método, URI y versión de HTTP utilizado
2. Encabezados de la solicitud
3. Cuerpo del mensaje, en caso de enviar datos al servidor

14-Identificación de métodos HTTP: Enumera y describe los principales métodos HTTP, como GET, POST, PUT y DELETE. Explica en qué situaciones se utiliza cada método y qué tipo de operaciones realizan en el servidor.

* **GET**: Solicita la recuperación de un recurso específico del servidor, como páginas web, imágenes o archivos.
* **POST**: Envía datos al servidor para que sean procesados. Es utilizado comúnmente en formularios web para enviar datos al servidor.
* **PUT**: Envía datos al servidor para ser almacenados en un recurso específico. Es utilizado para actualizar o crear recursos en el servidor.
* **DELETE**: Solicita la eliminación de un recurso específico almacenado en el servidor.

15-Códigos de estado HTTP: Investiga y enumera los códigos de estado más comunes en las respuestas HTTP, como 200 OK, 404 Not Found y 500 Internal Server Error. Explica el significado de cada código y cómo se utilizan para indicar el estado de una solicitud.

* **200 OK:** Indica que la solicitud se ha completado con éxito y que el servidor ha devuelto los datos solicitados al cliente.
* **404 Not Found:** Indica que el recurso solicitado no se ha encontrado en el servidor. Es comúnmente utilizado cuando una página web o recurso no está disponible.
* **500 Internal Server Error:** Indica que se ha producido un error interno en el servidor al procesar la solicitud del cliente.
* **403 Forbidden:** muy similar al código de estado 401, se produce cuando se envía una solicitud válida, pero el servidor se niega a aceptarla. Esto sucede si un cliente/usuario requiere el permiso necesario para acceder al recurso.
* **401:** Se produce cuando se requiere autenticación, pero se ha producido un error o no se ha proporcionado.

16-Manipulación de cabeceras HTTP: Describe el propósito de las cabeceras HTTP y cómo se utilizan para transmitir información adicional en las solicitudes y respuestas. Elige tres ejemplos de cabeceras HTTP y explica su función y aplicación.

Las cabeceras HTTP son componentes clave de las solicitudes y respuestas HTTP. Estas proporcionan información adicional sobre la solicitud o la respuesta, lo que permite una comunicación más completa y eficiente entre los componentes de la aplicación web.

Se suelen utilizar en solicitudes para:

* Especificar el tipo de datos que el cliente espera recibir
* Proporcionar credenciales o tokens de autenticación
* Indicar detalles de la conexión

Y en respuestas del servidor para indicar:

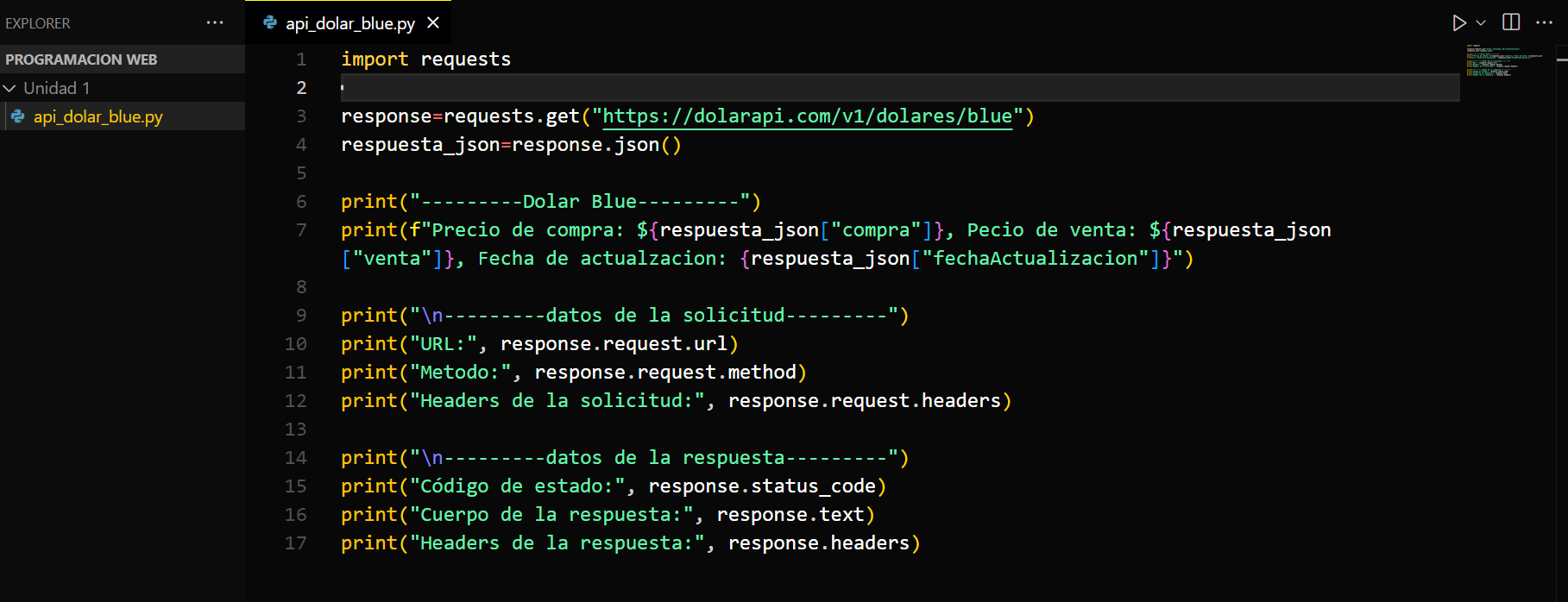
* El tipo de contenido que está enviando
* Instrucciones de caché y de redirección
* Cookies

Por ejemplo las cabeceras:

* **Cache-Control:** Es un encabezado que dicta el comportamiento de la [caché](https://www.cloudflare.com/learning/cdn/what-is-caching/) del navegador. Cuando alguien visita un sitio web, su navegador guarda ciertos recursos, en un almacén conocido como caché. Cuando ese usuario vuelve a visitar el mismo sitio web, el control de caché establece las reglas que determinan si esos recursos se cargarán desde su caché local, o si el navegador tendrá que enviar una solicitud al servidor para conseguir recursos nuevos.
* **Content-Security-Policy:** La política de seguridad de contenido (CSP) es una capa adicional de seguridad que ayuda a detectar y mitigar ciertos tipos de ataques, incluyendo Cross-Site Scripting (XSS) y ataques de inyección de datos. La configuración de una política de seguridad de contenido implica agregar el encabezado HTTP Content-Security-Policy a una página web y asignarle valores para controlar qué recursos puede cargar el agente de usuario para esa página.
* **Set-Cookie:** Esta cabecera es enviada por el servidor en respuesta a una petición HTTP, que se utiliza para crear una cookie en el sistema del usuario. El encabezado Cookie es incluido por la aplicación cliente con una petición HTTP enviada a un servidor, si existe una cookie que tenga un dominio y ruta coincidentes.

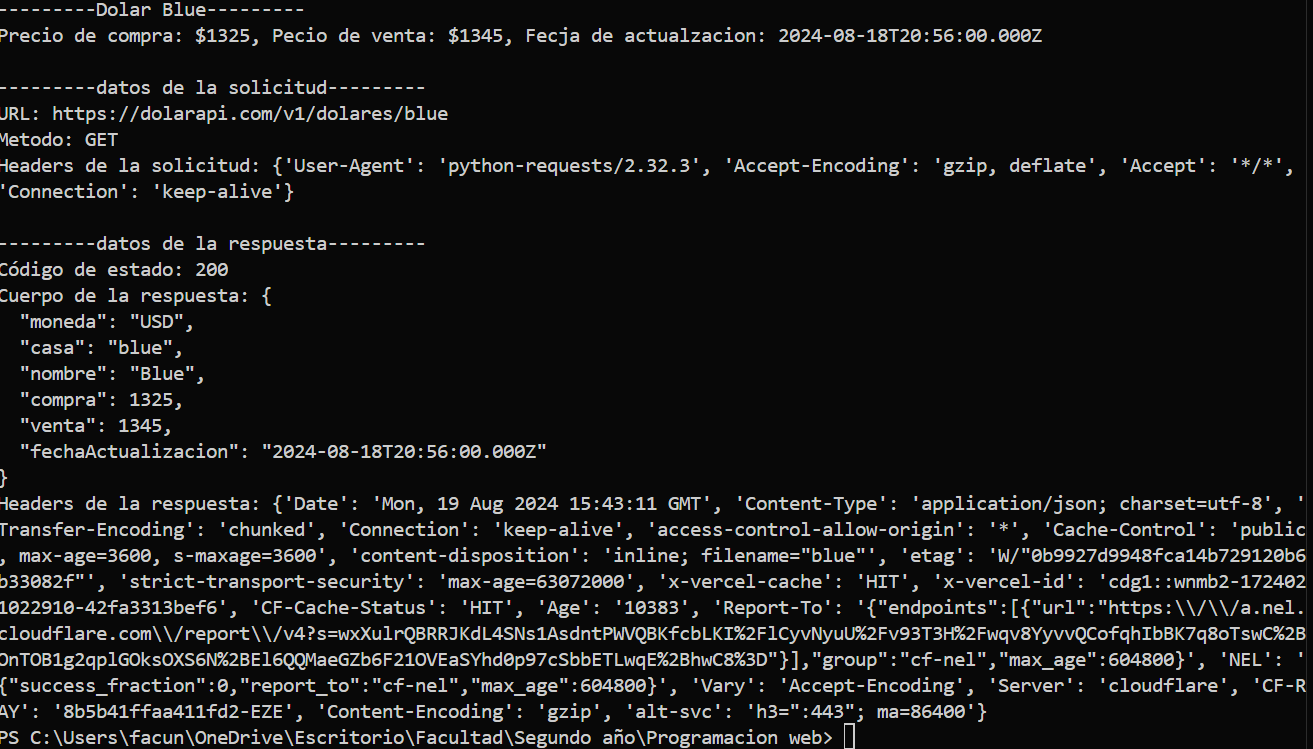
17-Implementación de solicitudes HTTP: Utilizando un lenguaje de programación de tu elección, crea un programa que realice una solicitud HTTP a un servidor de tu elección. Muestra cómo se configuran los encabezados y cómo se manejan las respuestas del servidor.

Se usará una API para obtener el valor del dólar blue en argentina usando python:



En este caso no se configuran encabezados de autenticación ya que la API no requiere API Key.

Se imprime la información de la solicitud y la respuesta para mostrar el método http que se usa, así como los headers y el código de estado http obteniendo la siguiente salida por consola:



18-Seguridad en el Protocolo HTTP: Investiga y describe las vulnerabilidades de seguridad asociadas con el Protocolo HTTP. Explica cómo se pueden mitigar estas vulnerabilidades utilizando medidas como HTTPS y certificados SSL/TLS.

Principales vulnerabilidades del protocolo HTTP:

1. **Inyección SQL:** EL atacante utiliza la base de código de la aplicación y ejecuta código malicioso para corromper la base de datos.Las inyecciones SQL son uno de los tipos más frecuentes de vulnerabilidades de seguridad de aplicaciones web.
2. **El cross-site scripting:** consiste en dirigirse a la aplicación de un usuario e inyectar código malicioso, normalmente un script del lado del cliente, como JavaScript. Permite a los atacantes lanzar scripts en el navegador web del usuario y apoderarse de las sesiones del usuario, manipular sitios web y redirigir a los usuarios a sitios web no deseados.
3. **Autenticación y gestión de sesiones rotas:** Esta vulnerabilidad rodea muchos problemas de seguridad, todos ellos conducentes a la gestión de la identidad del usuario. Si las credenciales de validación y los identificadores de sesión no están protegidos en todo momento, el atacante puede asumir la identidad del usuario.
4. **Error de configuración de seguridad:** Se debe realizar una configuración segura para las aplicaciones web que garantice que se tomen medidas concretas para proteger el servidor de aplicaciones, el servidor de bases de datos y la plataforma. Las configuraciones de seguridad débiles pueden permitir que los secuestradores accedan a las aplicaciones web y comprometan el sistema en su conjunto.
5. **La falsificación de solicitudes entre sitios (CSRF):** Un sitio web externo enviará una solicitud al sitio web original de que el usuario ya ha sido autenticado e iniciado sesión. A continuación, el atacante puede acceder a la cuenta a través del navegador previamente validado por la víctima.

Para combatir estas vulnerabilidades, surge el protocolo HTTPS (Hypertext Transfer Protocol Secure), es la versión segura de HTTP. Incorpora una capa adicional de seguridad mediante el uso de certificados SSL (Secure Socket Layer) o TLS (Transport Layer Security) para cifrar las comunicaciones entre el navegador del usuario y el servidor web. Lo que asegura que cualquier dato transferido permanezca privado y protegido contra posibles incursiones externas con malas intenciones. Además autentifica las páginas, asegurando a los usuarios que están conectados a la web correcta y no a una réplica fraudulenta.

19-Protocolo FTP: Describe el Protocolo FTP (File Transfer Protocol) y cómo se utiliza para transferir archivos entre clientes y servidores. Explica los comandos básicos utilizados en una sesión FTP y cómo se establece la autenticación.

FTP (File Transfer Protocol) es un protocolo estándar utilizado para la transferencia de archivos entre un cliente y un servidor en la web. Este protocolo implica la conexión de un cliente FTP a un servidor FTP a través de una conexión de red. Una vez establecida la conexión, se puede enviar comandos al servidor para realizar operaciones, como listar los archivos en el servidor, descargar, cargar y eliminar archivos en el servidor.

Para establecer una conexión FTP, se necesita un programa cliente, una aplicación que nos permite conectarnos al servidor FTP, así también debemos disponer de todos los datos de configuración de la conexión: nombre o dirección IP del servidor, tipo, nombre de usuario y contraseña.

Principales comandos para trabajar en una sesión FTP:

* ***cd [directorio]:*** Cambia al directorio especificado.
* ***ls:*** Lista los archivos en el directorio actual.
* ***get [archivo]:***Descarga un archivo del servidor al cliente.
* ***put [archivo]:*** Sube un archivo del cliente al servidor.
* ***pwd:*** Muestra el nombre del directorio de trabajo remoto.
* ***mkdir:*** Crea un directorio en el sistema remoto.
* ***rmdir:*** Elimina un directorio en el sistema remoto.
* ***delete, mdelete:*** Elimina un archivo (o varios archivos) del directorio de trabajo remoto.

20-Protocolo SSL y TLS: Investiga y describe los protocolos SSL (Secure Sockets Layer) y TLS (Transport Layer Security), utilizados para proporcionar seguridad en las comunicaciones web. Explica cómo se establecen las conexiones seguras y cómo se validan los certificados digitales.

SSL es una tecnología estandarizada que permite cifrar el tráfico de datos entre un navegador web y un sitio web (o entre dos servidores web), protegiendo la conexión. Esto impide que un hacker pueda ver o interceptar la información que se transmite de un punto a otro.

TLS es una versión actualizada y más segura de SSL. Aunque seguimos refiriéndonos a nuestros certificados de seguridad como “certificados SSL” porque ese es el término más extendido.

El procedimiento o la lógica que se sigue a la hora de implementar estas tecnologías es el siguiente:

1. El cliente solicita acceso a un recurso protegido
2. El servidor responde enviando su certificado SSL, incluida la clave pública.
3. El cliente verifica que el certificado sea válido y confiable. Esto garantiza que el servidor sea auténtico.
4. El cliente genera una clave de sesión simétrica y la cifra con la clave pública del servidor. Esto transmite de forma segura la clave de sesión al servidor.
5. El servidor descifra la clave de sesión con su clave privada.
6. Ambas partes utilizan la clave de sesión simétrica para cifrar y descifrar todos los datos transmitidos.

Para el proceso de validación de certificados digitales sse realizan los siguientes pasos:

1. **Verificación de Firma:** El cliente utiliza la clave pública de la autoridad certificadora (CA) para verificar que la firma digital en el certificado es válida.
2. **Comprobación de Caducidad:** Verifica las fechas de validez, el certificado incluye una fecha de inicio y caducidad.
3. **Verificación de la Cadena de Confianza:** Asegura que el certificado esté firmado por una CA de confianza.
4. **Validación del Nombre del Servidor:** Se comprueba que el nombre del servidor en el certificado coincida con el nombre de dominio al que el cliente está intentando conectarse.
5. **Verificación de Revocación:** Se asegura de que el certificado no haya sido revocado por la CA antes de su fecha de expiración.
6. **Comprobación de Políticas:** Asegura que el certificado cumple con las políticas y restricciones establecidas por la CA y el entorno.

**Referencias bibliográficas**

[Unidad 1: Introducción a la Web y HTTP](https://docs.google.com/document/d/125WRTxBLf1b9aIvMKNjlzHcWb8KD5bZ42PIP7_-nOzY/edit#heading=h.u9nwu582xvn9)

[**Resumen de la historia de la Web. Páginas web HTML y hojas de estilo CSS.**](https://www.mclibre.org/consultar/htmlcss/otros/historia-resumen.html#:~:text=La%20primera%20guerra%20de%20los%20navegadores%3A%20Internet%20Explorer%20%2F%20Netscape&text=En%201995%20Microsoft%20incluy%C3%B3%20en,llamada%20guerra%20de%20los%20navegadores.)

[**Bartolomé Sintes Marco.**](https://www.mclibre.org/consultar/htmlcss/otros/historia-resumen.html#:~:text=La%20primera%20guerra%20de%20los%20navegadores%3A%20Internet%20Explorer%20%2F%20Netscape&text=En%201995%20Microsoft%20incluy%C3%B3%20en,llamada%20guerra%20de%20los%20navegadores.) [**www.mclibre.org**](http://www.mclibre.org)

[**EVOLUCION DE LA WEB 1.0 A WEB 5.0 timeline | Timetoast Timelines**](https://www.timetoast.com/timelines/evolucion-de-la-web-1-0-a-web-5-0)

[**World Wide Web - Wikipedia, la enciclopedia libre**](https://es.wikipedia.org/wiki/World_Wide_Web)

[**HTTP/2 vs. HTTP/1.1 | Cloudflare**](https://www.cloudflare.com/es-es/learning/performance/http2-vs-http1.1/)

[**Ciberseguridad: Tipos de ataques y en qué consisten (iebschool.com)**](https://www.iebschool.com/blog/ciberseguridad-ataques-tecnologia/)

[**¿Qué son los ataques DDoS y cómo evitarlos? (kaspersky.com)**](https://latam.kaspersky.com/resource-center/threats/ddos-attacks?srsltid=AfmBOopIZ5eqC9xDoHTdSUJVkc3_E9quPS0U1DfzJxbks1kU3O8M_AHP)

[**Comparación entre Opera y Edge | Elige el mejor navegador | Opera**](https://www.opera.com/es/compare/opera-vs-edge)

[**Comparativa detallada: Opera GX vs Edge - Descubre cuál es el mejor navegador para ti. (byronvargas.com)**](https://www.byronvargas.com/web/que-es-mejor-opera-gx-o-edge/)

[**WebVR API - Referencia de la API Web | MDN (mozilla.org)**](https://developer.mozilla.org/es/docs/Web/API/WebVR_API)

[**WebVR - Bringing Virtual Reality to the Web**](https://webvr.info/)

[**Diseño web: 6 casos de éxito que debes conocer (kiwop.com)**](https://www.kiwop.com/blog/6-casos-de-exito-en-diseno-web)

[**¿Qué es la arquitectura cliente-servidor? | by NUCBA | Medium**](https://nucba.medium.com/qu%C3%A9-es-la-arquitectura-cliente-servidor-eb9f402506cc)

[**Arquitectura cliente servidor: qué es, tipos y ejemplos | Blog de Arsys**](https://www.arsys.es/blog/todo-sobre-la-arquitectura-cliente-servidor)

[**Redes Peer-to-Peer vs Cliente Servidor | Todas las Comparativas (aprendeinformaticas.com)**](https://aprendeinformaticas.com/redes-peer-to-peer-vs-cliente-servidor/)

[**Qué es el protocolo HTTP y cómo funciona - Platzi**](https://platzi.com/clases/1638-api-rest/21614-que-es-y-como-funciona-el-protocolo-http/)

[**Generalidades del protocolo HTTP - HTTP | MDN (mozilla.org)**](https://developer.mozilla.org/es/docs/Web/HTTP/Overview#arquitectura_de_los_sistemas_basados_en_http)

[**Los 10 códigos de estado HTTP más comunes - Dotcom-Monitor Web Performance Blog**](https://www.dotcom-monitor.com/blog/es/los-10-most-common-http-status-codes/)

[**HTTP headers - HTTP | MDN (mozilla.org)**](https://developer.mozilla.org/es/docs/Web/HTTP/Headers)

[**¿Qué es el control de caché? | La caché explicada | Cloudflare**](https://www.cloudflare.com/es-es/learning/cdn/glossary/what-is-cache-control/)

[**Política de seguridad de contenido (CSP) - HTTP | MDN (en inglés) (mozilla.org)**](https://developer.mozilla.org/en-US/docs/Web/HTTP/CSP)

[**HTTP Cookies - Win32 apps | Microsoft Learn**](https://learn.microsoft.com/es-es/windows/win32/wininet/http-cookies)

[**¿Qué es la vulnerabilidad HTTP y sus tipos? (purevpn.com)**](https://www.purevpn.com/ddos/http-vulnerability)

[**Conexión FTP: qué es y programas populares - Cloud Center Andalucía (cloudcenterandalucia.es)**](https://www.cloudcenterandalucia.es/blog/conexion-ftp-que-es-que-necesitas-y-programas-populares/#Que_necesito_para_establecer_una_conexion_FTP)

[**Inicio de sesión en un sistema remoto (ftp) - Guía de administración del sistema: servicios de red (oracle.com)**](https://docs.oracle.com/cd/E24842_01/html/E22524/remotehowtoaccess-14.html)

[**¿Qué son SSL, TLS y HTTPS? | DigiCert**](https://www.digicert.com/es/what-is-ssl-tls-and-https)

[**Que es SSL /TLS: Una guía detallada - SSL.com**](https://www.ssl.com/es/art%C3%ADculo/que-es-ssl-tls-an-in-depth-guide/)