UNIDAD 1

Fundamentos de la Web

Son los principios básicos que sustentan el funcionamiento de Internet y las tecnologías asociadas que permiten la navegación, interacción y acceso a información a través de la World Wide Web.

* **URI (Uniform Resource Identifier):** Es una cadena de caracteres que identifica de manera única un recurso en la web.
* **CSS (Cascading Style Sheets):** Es un lenguaje utilizado para dar estilo y diseño a las páginas web creadas con HTML.
* **JavaScript**: Es un lenguaje de programación utilizado para agregar interactividad y dinamismo a las páginas web. Permite manipular el contenido de la página en tiempo real, responder a eventos del usuario y comunicarse con servidores web.
* **DOM (Document Object Model):** Es una representación en memoria de la estructura de una página web creada con HTML. Permite acceder y manipular los elementos de la página mediante programación

# **Protocolos de comunicación**

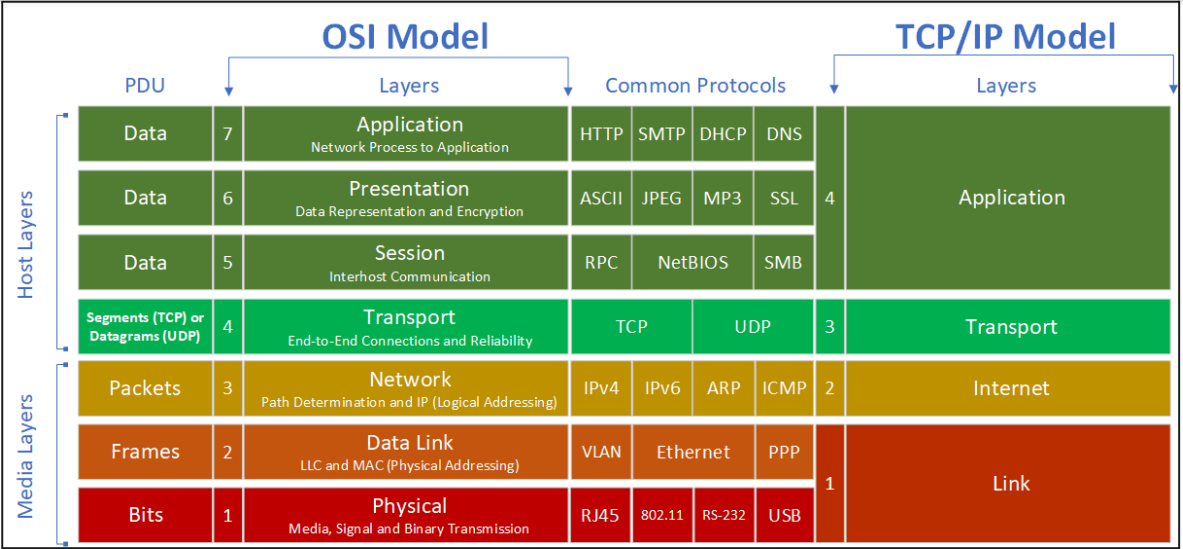
Los protocolos más comunes están: TCP, IP, ARP o HTTP. A un conjunto de protocolos que trabajan juntos se le llama pila de protocolos.

Los protocolos tienen que lidiar con los siguientes problemas:

* **Inicio de la conexión:** Inicia el servidor o el cliente, que se necesita
* **Negociación de las características de la conexión**: Cifrado y encriptado
* **Formato de los datos:** Formato del paquete y orden de lectura
* **Detección y corrección de errores:** Demasiado tiempo de respuesta o desconexión
* **Fin de la conexión:** Indicar fin de conexión

**El modelo OSI** o también llamado la pila de capas o niveles del modelo OSI, es un modelo de referencia ideal para los protocolos de la red, los cuales están separados de acuerdo a sus funciones. Está formado por siete capas o layers:

* Las Host layers son las responsables de proporcionar una entrega precisa de los datos entre los equipos.
* Las Media layers son las responsables de controlar la entrega física de los datos a través de la red.



* La capa 7 o de **aplicación** es el nivel más alto del modelo OSI, es la capa vista por los usuarios finales. (HTTP, SMPT, FTP, etc.)
* La capa 6 o de **presentación** se encarga de transformar los datos que recibe en un formato que pueda ser leído por la capa de aplicación o por las capas inferiores. (ASCII, JPEG, MP3)
* La capa 5 o de **sesión** controla el diálogo entre los dispositivos. Establece, gestiona, mantiene y termina las conexiones entre los equipos involucrados en la comunicación.
* La capa 4 o de **transporte** tiene como principal objetivo garantizar la correcta transferencia de los datos entre equipos (protocolos TCP y UDP)
* La capa 3 o de **red** es responsable de enrutar los datos entre las distintas redes físicas. Utiliza las direcciones lógicas de los dispositivos de red. También a veces se encarga de dividir los datos en fragmentos más pequeños y de la detección de errores. (IPv4, IPv6)
* La capa 2 o de **enlace** proporciona la transferencia de datos de nodo a nodo, un enlace o medio de transporte entre dos o más dispositivos directamente conectados. Generalmente se divide en dos subcapas: LLC que define la forma de transferencia de los datos y MAC que se encarga del direccionamiento. (Ethernet, VLAN, etc.)
* La capa 1 o **física** es el medio físico por el cual los datos de red son transferidos, la información se transfiere en bits. (cables como el RJ45 o el 802.11, USB, etc.)

# **Encapsulación de los datos**

Los protocolos de las capas del modelo OSI intercambian los datos entre ellas con la ayuda de la encapsulación. El proceso de la encapsulación da como resultado una unidad de protocolo de datos o **PDU**, el cual incluye los datos que se envían y además todas las headers o footers añadidos.

# **Introducción a la World Wide Web**

La web se compone de una intrincada red de servidores y clientes interconectados que se comunican a través del Protocolo de Transferencia de Hipertexto (HTTP)

* La primera generación de la web era un sistema unidireccional de publicación estático de sólo texto que no presentaba gráficos
* La segunda generación se basó en compartir e interactuar con la información de una manera sencilla.
* El objetivo de la tercera generación fue darle significado a la web y enriquecer la experiencia del usuario.
* La cuarta generación de la web se centró en la interactividad y la participación del usuario.
* La quinta generación de la web se caracteriza por la convergencia de la web y las tecnologías emergentes como la IA, el Internet de las cosas (IoT) y la realidad aumentada.

# **Arquitectura Cliente-Servidor y Protocolo HTTP**

La arquitectura cliente-servidor es un modelo de diseño de software que distribuye las funciones de una aplicación entre dos entidades principales: el cliente y el servidor.

* **Cliente**: Es la interfaz de usuario o la aplicación que solicita servicios y consume recursos proporcionados por el servidor. Su función principal es enviar solicitudes al servidor y mostrar los resultados al usuario.
* **Servidor**: Es el componente que almacena, procesa y gestiona los datos y recursos de la aplicación. Recibe las solicitudes del cliente, las procesa y envía las respuestas adecuadas.

Esta comunicación puede ser síncrona, donde el cliente espera una respuesta inmediata del servidor, o asíncrona, donde el cliente continúa su ejecución mientras espera la respuesta del servidor en segundo plano.

De forma básica, cuando un usuario se conecta a Internet con un dispositivo cualquiera, se le asigna un identificador único mediante los protocolos **TCP/IP**. El protocolo TCP proporciona el medio para crear las conexiones y el protocolo IP proporciona el mejor “camino” para alcanzar su destino.

Este identificador único, más conocido como **dirección IP**, suele estar compuesto por cuatro códigos de 8 bits y vinculado a un nombre único (dominio) como https://google.es.

Un **DNS** sirve cuando un usuario ingresa un nombre de dominio en su navegador, el DNS busca la dirección IP correspondiente a ese nombre.

# **PROTOCOLO HTTP**

Una vez que se tiene el objetivo al que dirigirse, el cliente en términos de comunicaciones, abre una instancia de comunicación con el Servidor mediante el **protocolo HTTP**, que es quién dicta las normas la comunicación y es, además, quién define la sintaxis y semántica que se debe utilizar en cada conexión.

**HTTPS** es la versión segura de HTTP. Incorpora una capa adicional de seguridad mediante el uso de certificados SSL o TLS para cifrar las comunicaciones entre el navegador del usuario y el servidor web

El protocolo HTTP define una serie de **métodos** que especifican la acción que el cliente desea realizar sobre un recurso determinado en el servidor. Algunos de los métodos más comunes incluyen:

* **GET**: Solicita la recuperación de un recurso específico del servidor.
* **POST**: Envía datos al servidor para que sean procesados. Es utilizado comúnmente en formularios web
* **PUT**: Envía datos al servidor para ser almacenados en un recurso específico.
* **DELETE**: Solicita la eliminación de un recurso específico en el servidor.

Además de los métodos, el protocolo HTTP define **códigos de estado** que indican el resultado de la solicitud realizada por el cliente. Algunos de los códigos de estado más comunes son:

* **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.
* **500 Internal Server Error**: Indica que se ha producido un error interno en el servidor
* al procesar la solicitud del cliente.

Las **cabeceras HTTP** son componentes clave de las solicitudes y respuestas HTTP que se intercambian entre clientes y servidores. Estas cabeceras 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.

**Tipos de Cabeceras HTTP:**

* **Cabeceras de Solicitud:** Enviadas por el cliente al servidor para proporcionar información sobre la solicitud
* **Cabeceras de Respuesta**: Enviadas por el servidor al cliente para proporcionar información sobre la respuesta

**Funciones de las Cabeceras HTTP:**

* Control de Caché
* Negociación de Contenido
* Seguridad
* Autenticación y Autorización
* Control de Cookies

# **Protocolo FTP y SSL**

El Protocolo de Transferencia de Archivos (FTP) es un protocolo estándar utilizado para la transferencia de archivos entre un cliente y un servidor en una red de computadoras.

Algunas características importantes del Protocolo FTP incluyen:

* **Modos de Transferencia**: En el **modo activo**, el servidor inicia la conexión de datos con el cliente, mientras que, en el **modo pasivo**, el cliente inicia la conexión de datos con el servidor.
* **Autenticación**: FTP proporciona mecanismos de autenticación para garantizar la seguridad durante la transferencia de archivos
* **Seguridad:** Por defecto no viene cifrado, pero existen las versiones FTPS y SFTP (SSH)
* **Flexibilidad y Versatilidad:** Sirve para transferencia usuario-usuario como para grandes volúmenes de datos.

# **Protocolo SSH**

El **Protocolo SSH** es un protocolo de red que proporciona una forma segura de acceder y administrar servidores remotos a través de una conexión cifrada.

Algunas características clave del Protocolo SSH incluyen:

* **Cifrado de Datos:** Solo el cliente y servidor autorizados pueden acceder a la información.
* **Autenticación del Servidor y del Cliente:** La autenticación del servidor se realiza mediante el intercambio de claves públicas y privadas, mientras que la autenticación del cliente puede basarse en contraseñas, claves públicas, tokens, etc.
* **Túneles Seguros:** Esto es especialmente útil para acceder a servicios y recursos internos de una red privada desde ubicaciones externas.
* **Gestión de Sesiones Interactivas y No Interactivas:** SSH permite ejecutar comandos remotos para administrar servidores y sistemas.

**Técnicas de cifrado**

Hay tres tecnologías de cifrado diferentes utilizadas por SSH:

1. Cifrado simétrico: Se utiliza una clave secreta tanto para el cifrado como para el descifrado de un mensaje, tanto por el cliente como por el host.

Es muy seguro porque la clave nunca se transmite entre el cliente y el host. Los dos equipos comparten datos públicos y luego los manipulan para calcular de forma independiente la clave secreta. Incluso si otra máquina captura los datos públicamente compartidos, no será capaz de calcular la clave porque el algoritmo de intercambio de clave no se conoce.

1. Cifrado asimétrico: Se utiliza dos claves separadas para el cifrado y el descifrado. Estas dos claves se conocen como la clave pública y la clave privada. La clave pública se distribuye abiertamente. Un mensaje cifrado por la clave pública de una máquina, sólo puede ser descifrado por la misma clave privada de la máquina. Sólo se utiliza durante el algoritmo de intercambio de claves de cifrado simétrico
2. Hashing: Se usa como forma de verificar si una entrada es correcta, si un cliente tiene la entrada correcta, pueden generar el hash criptográfico y comparar su valor para verificar si poseen la entrada correcta.  
   Mientras se selecciona el algoritmo de cifrado simétrico, también se selecciona un algoritmo de autenticación de mensajes adecuado.

**Negociación de cifrado de sesión**

Cuando un cliente intenta conectarse al servidor a través de TCP, el servidor presenta los protocolos de cifrado y las versiones respectivas que soporta. Si el cliente tiene un par similar de protocolo y versión, se alcanza un acuerdo y se inicia la conexión con el protocolo aceptado.  
Una vez que esto se establece se ejecuta el Algoritmo de Intercambio de Claves:

1. Tanto el cliente como el servidor coinciden en un número primo muy grande (semilla), que por supuesto no tiene ningún factor en común

2. Luego, las dos partes acuerdan un mecanismo de cifrado común para generar otro conjunto de valores manipulando los valores semilla de una manera algorítmica específica.

3. Ambas partes generan independientemente otro número primo. Esto se utiliza como una clave **privada secreta para la interacción**

4. Esta clave privada recién generada, con el número compartido y el algoritmo de cifrado, se utiliza para calcular una **clave pública** que se distribuye a la otra computadora.

5. Las partes utilizan su clave privada personal, la clave pública compartida de la otra máquina y el número primo original para crear una clave compartida final, calculada de forma independiente por ambos equipos.

6. Ahora que ambas partes tienen una clave compartida, pueden cifrar simétricamente toda la sesión SSH.

# **Uniform Resource Locators - URL**

Una **URL** es una dirección que se utiliza para identificar de manera única un recurso en internet.

* **Protocolo**: El protocolo es la parte inicial de una URL y define cómo se debe acceder al recurso. HTTP, HTTPS, FTP y mailto.
* **Dominio:** También conocido como nombre de host, identifica la ubicación específica en la web donde se encuentra el recurso.
* **Ruta**: La ruta especifica la ubicación exacta del recurso dentro del servidor.
* **Parámetros**: Los parámetros son datos adicionales que se pueden incluir en la URL para proporcionar información adicional al servidor.
* **Fragmento**: El fragmento, también conocido como ancla, identifica una sección específica dentro de un recurso más grande, como una página web

https://www.ejemplo.com/ruta/pagina.html?parametro1=valor1&parametro2=valor2#seccion

Desglose de los componentes:

* Protocolo: https://
* Dominio: [www.ejemplo.com](http://www.ejemplo.com)
* Ruta: /ruta/pagina.html
* Parámetros: ?parametro1=valor1&parametro2=valor2
* Fragmento: #seccion

Las URL solo se pueden enviar a través de Internet utilizando el juego de caracteres ASCII. **Si una URL contiene caracteres fuera del conjunto ASCII, la URL debe convertirse**.

UNIDAD 2

# **HTML (HyperText Markup Language)**

Es un lenguaje de marcado que permite a los desarrolladores web organizar y presentar información en forma de documentos hipertextuales, los cuales pueden contener texto, imágenes, enlaces, formularios, videos y mucho más. HTML proporciona una forma estructurada de organizar la información, utilizando etiquetas que definen el significado y la función de cada elemento en una página web.