

Desarrollo Web

Segundo semestre de 2020

Docente: Dora Suárez



Temas

- Introducción y levantamiento de requerimientos
- Desarrollo de páginas web con HTML
- Elementos de diseño y CSS
- Fundamentos de Javascript
- Javascript del lado del cliente
- Javascript del lado del servidor

Modelos conceptuales (OSI)



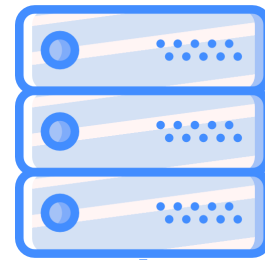
Protocolos de comunicación

Protocolo de red: las reglas que controlan el intercambio ordenado de datos en la red y los métodos de recuperación de errores

Los protocolos pueden ser implementados por software o hardware

Definen además la forma en la que se identifican diferentes dispositivos en la red y cómo la información circula por ella

Transmisor
Codificación

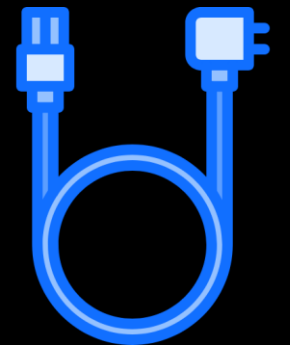


Receptor
Decodificación



Objetivo: Establecer un canal de comunicación para el envío de información.

- Detección de la conexión física subyacente (con cable o inalámbrica), o la existencia de otro punto final o nodo.
- Establecimiento de la comunicación.
- Negociación de varias características de la conexión.
- Cómo iniciar y finalizar un mensaje.
- Procedimientos en el formateo de un mensaje.
- Qué hacer con mensajes corruptos o formateados incorrectamente (corrección de errores).
- Cómo detectar una pérdida inesperada de la conexión, y qué hacer entonces.
- Terminación de la sesión y/o conexión.
- Estrategias para mejorar la seguridad (autenticación, cifrado).

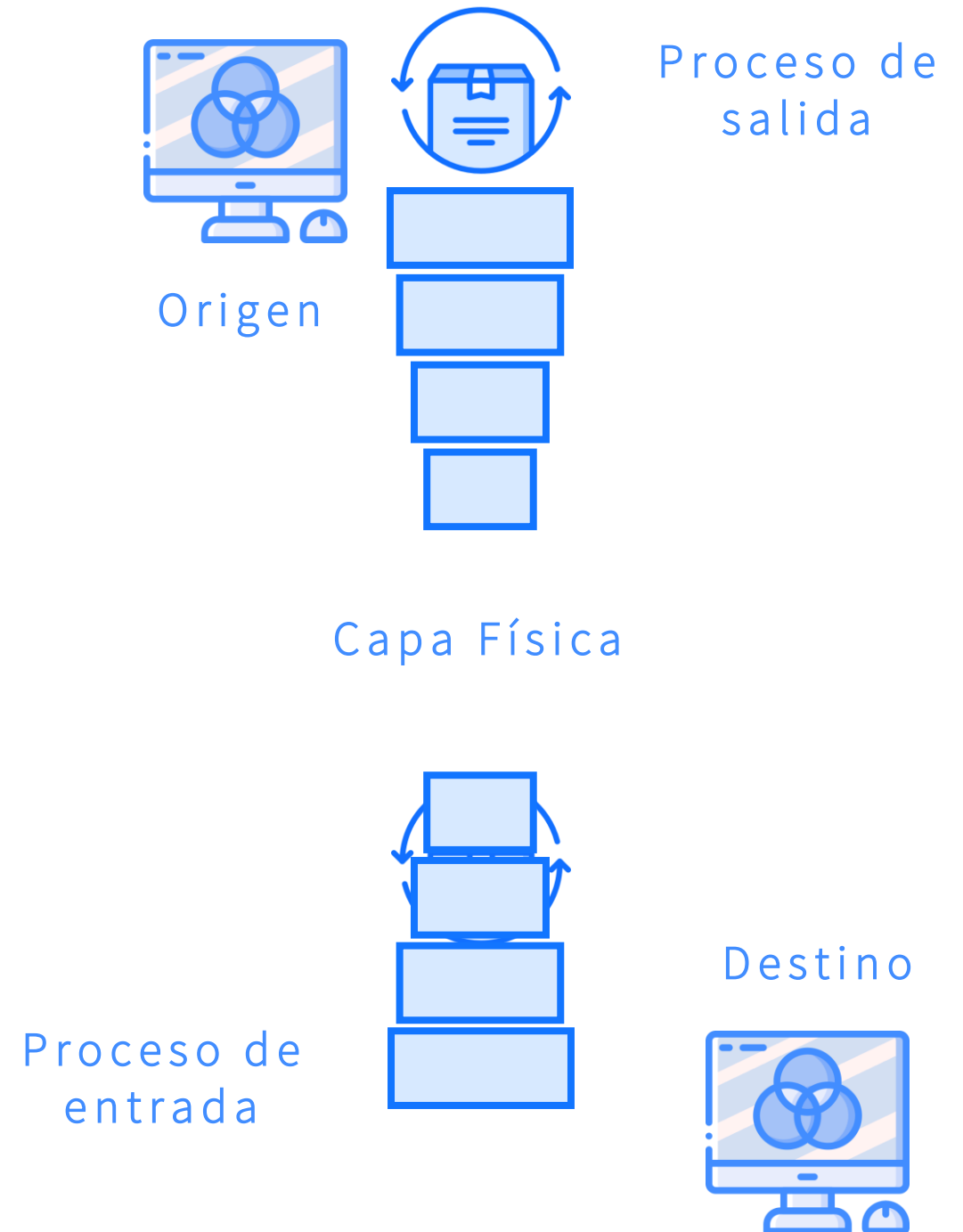


Modelo OSI

Interconexión de sistemas abiertos

Es un modelo de referencia para los protocolos de la red creada por ISO

Está basado en siete capas que definen las diferentes fases por las que deben pasar los datos



Modelo OSI

Interconexión de
sistemas abiertos

Es un modelo de referencia
para los protocolos de la red
creada por ISO

Está basado en siete capas que
definen las diferentes fases por
las que deben pasar los datos

Capa física

Capa de enlace de datos

Capa de red

Capa de transporte

Capa de sesión

Capa de presentación

Capa de aplicaciones

Capa de aplicación

Proporciona la interfaz para que las aplicaciones tengan acceso a la red

Se comunica en forma directa con el programa de aplicación del usuario

- Gestores de bases de datos
- FTP
- Correo electrónico





68 101 115 97
114 114 111 108
108 111 32 119
101 98

Capa de presentación

Convierte a los clientes
usar una sintaxis común

- Codificar (ASCII, EBCDIC)
- Cifrado de mensajes
- Compresión de mensajes

Capa de sesión

Mantiene y controla el enlace establecido entre los dos dispositivos de principio a fin de las operaciones a ser realizadas

Protocolo RCP

Llamada a procedimiento remoto

Protocolo SCP

Copia segura

Protocolo ASP

Protocolo de sesión APPLE TALK



	0 - 15	16 - 31
0	Puerto de origen	Puerto de destino
32	Longitud de mensaje	Suma de verificación
64	Datos	

Permite pérdida: Voz o video

No Permite pérdida: correos

Capa de transporte

Se elige un protocolo (confiable o no confiable para establecer la conexión)
Proporciona el tipo adecuado de conexión

Paquete IP (independiente del medio)

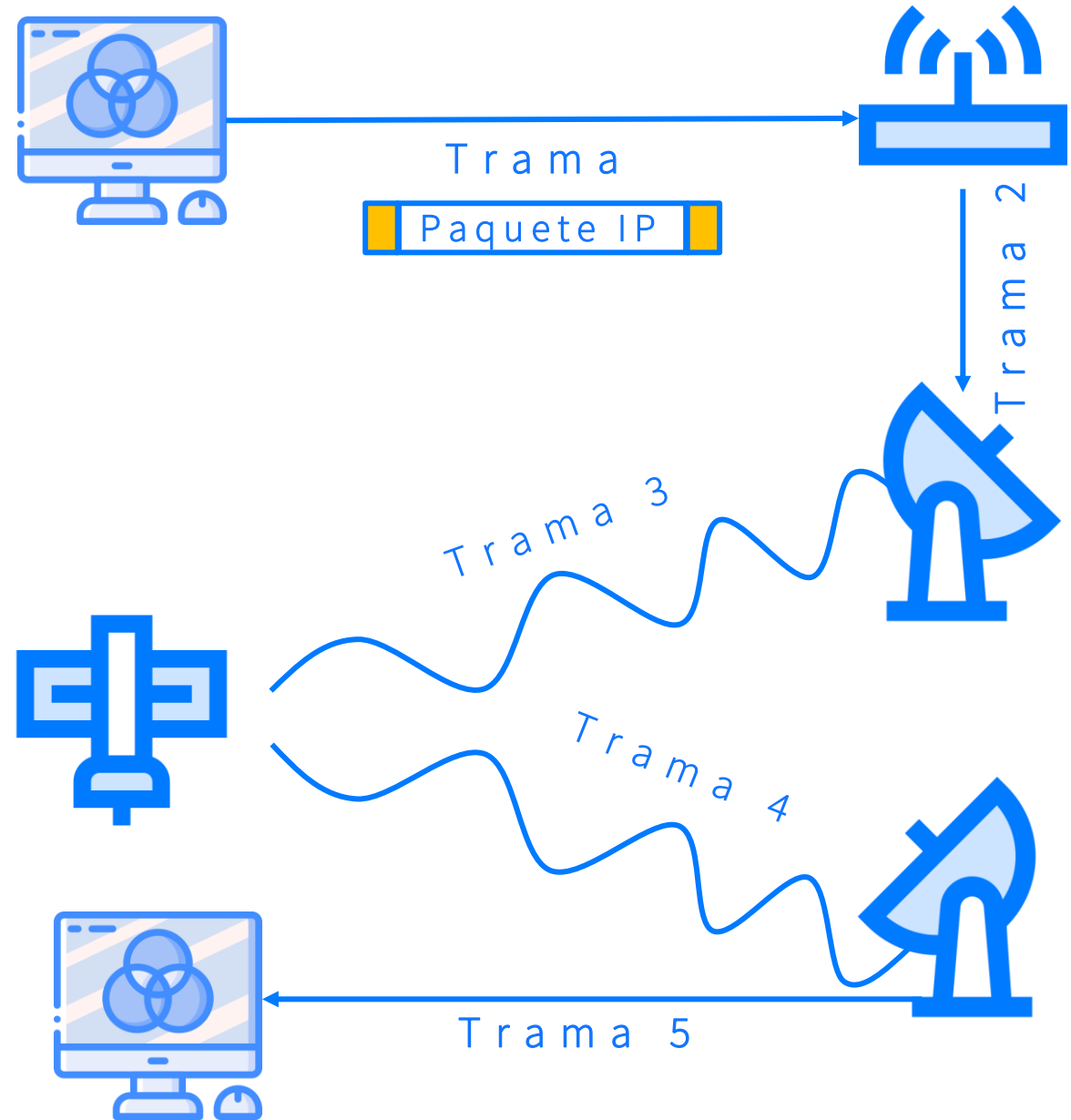
- UDP [User Datagram Protocol]
- TCP [Protocolo de control de transmisión]

Capa de enlace de datos

Es la encargada de traducir los datos de software al mundo de hardware

Encapsula el paquete IP en *tramas* para facilitar el acceso al medio

Las tramas son dependientes del medio



Capa de enlace de datos

Es la encargada de traducir los datos de software al mundo de hardware

La trama sirve como un encapsulador.

Encabezado de la trama

Inicio de trama

Dirección MAC
origen/destino

Tipo de protocolo
encapsulado

Datos

FCS
(frame check sequence)

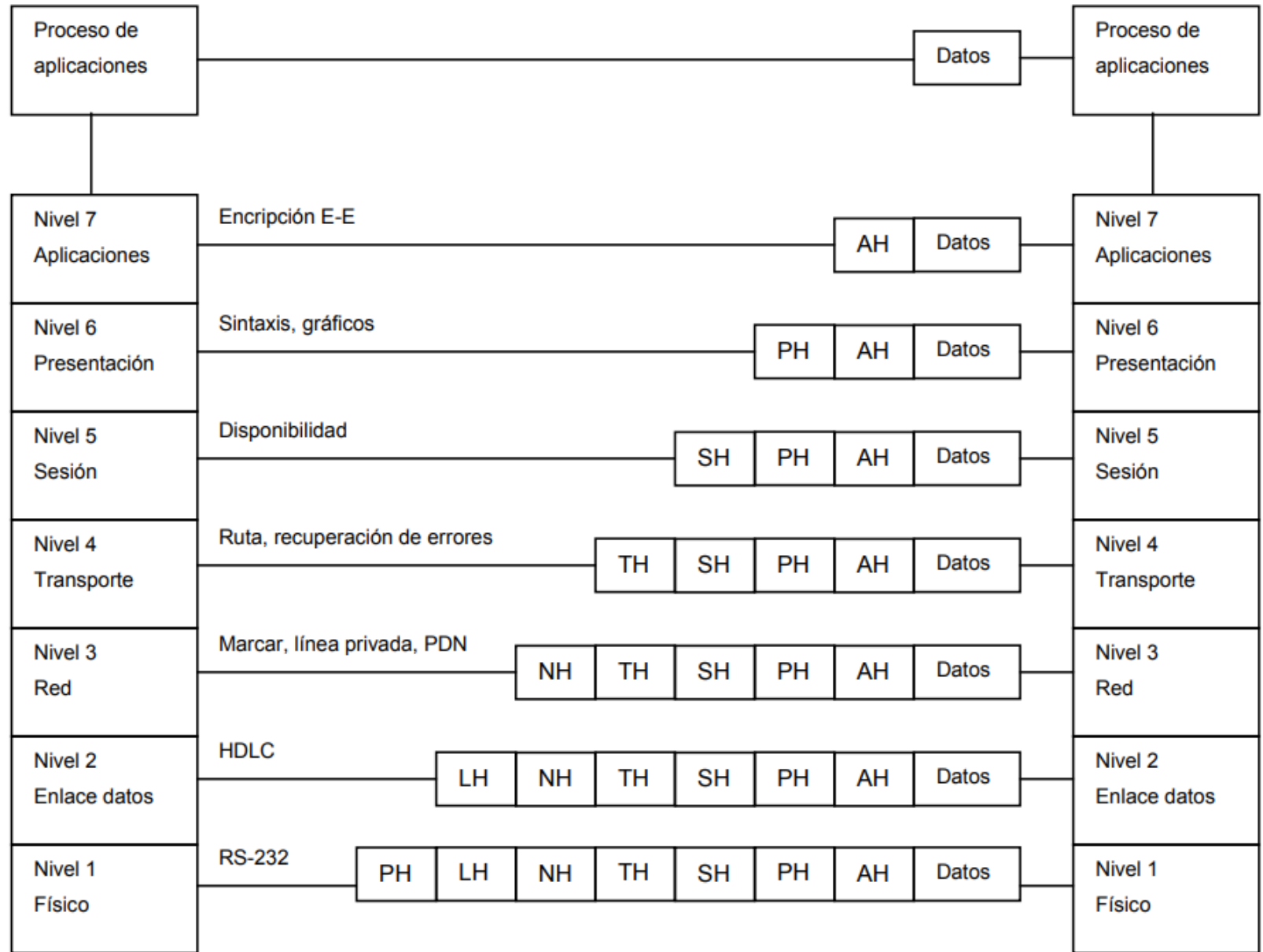
Capa física

Topología de red y
conexiones globales

Establece las normas físicas,
eléctricas, funcionales y de
procedimiento para entrar a
la red de comunicación de
datos

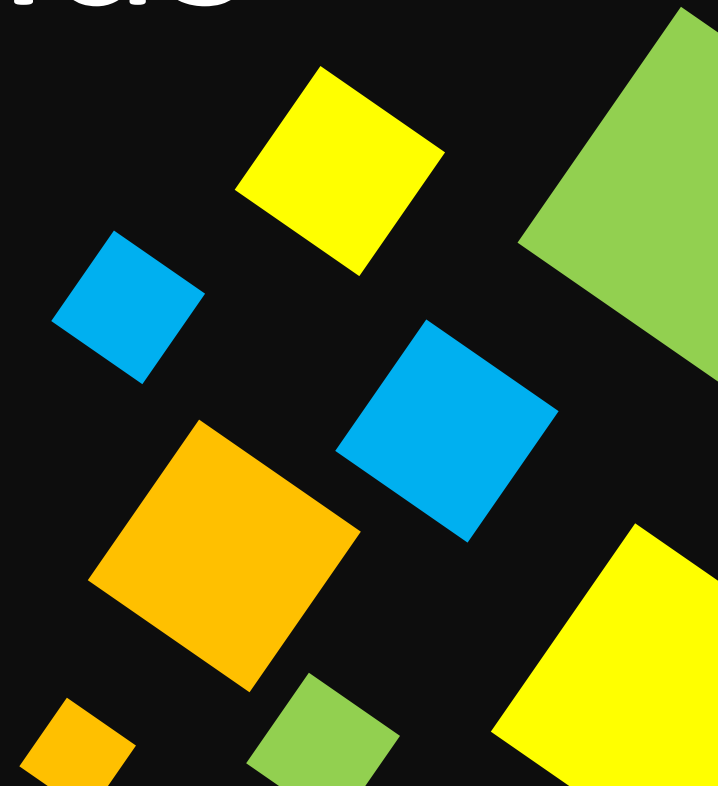
<https://www.submarinecablemap.com/>

Resumen

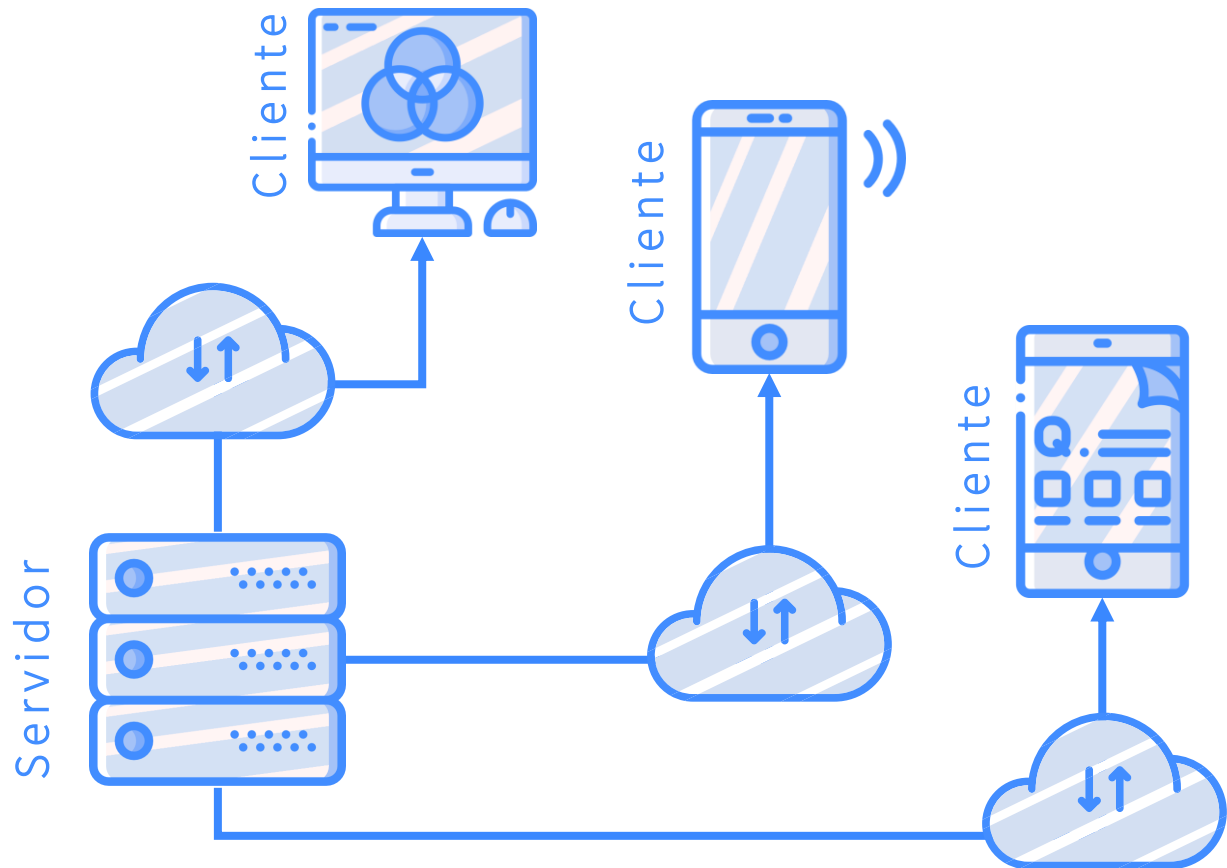


Jerarquías del protocolo internacional ISO. AH = encabezado de aplicaciones; PH = encabezado de presentación; SH = encabezado de sesión; TH = encabezado de transporte; NH = encabezado de red; LH = encabezado de enlace; PH = encabezado físico

¿Cómo funcionan las aplicaciones web?



Modelo Cliente Servidor



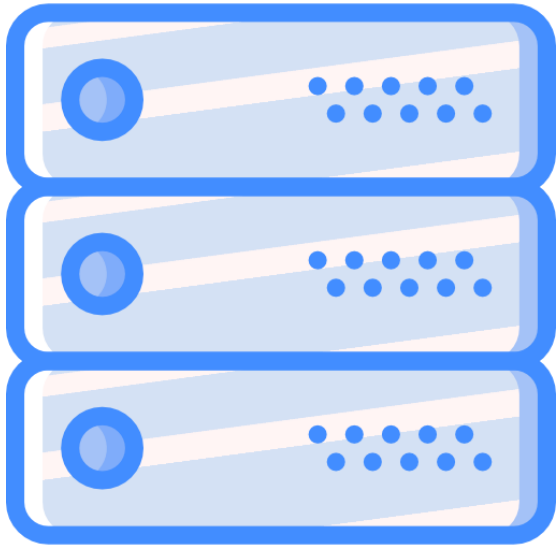
- Las tareas se reparten entre los proveedores de recursos o servicios [De forma distribuida]

- Los usuarios finales pueden estar dispersos en un área geográfica más o menos extensa

- El servidor y el cliente pueden ser aplicaciones diferentes en un mismo computador

- Facilita la integridad y el mantenimiento tanto de los datos como de los programas debido a que se encuentran centralizados en el servidor

Servidor



- Se limita a escuchar en espera de que lleguen las solicitudes de los clientes [dispositivo esclavo]

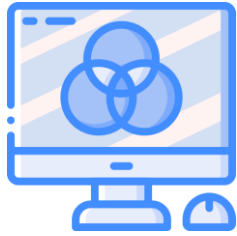
- Ofrece una serie de servicios encapsulados para que los clientes no conozcan su detalle

- Tras la recepción de una solicitud, la procesa y luego la envía al cliente

- Acepta conexiones de un gran número de cliente [Generalmente]

- Las respuestas de las instrucciones pueden ser datos, recursos, procedimientos, etc.

Cliente



- El cliente es una aplicación o un computador que envía instrucciones de forma remota al servidor e interactúa con el usuario

- Es quien inicia solicitudes o peticiones, Tiene un papel activo en la comunicación

- Espera y recibe las respuestas del servidor

- Por lo general, puede conectarse a varios servidores a la vez

- Normalmente interactúa con los usuarios mediante una interfaz gráfica

URL

Protocolo (por defecto http)

Puerto (por defecto 80)

http://www.desarrollo2020.com:80/inicio

Servidor

Recurso

Solicitud HTTP

La solicitud se empaqueta y se envía al servidor



Aquí el servidor toma la solicitud y la procesa usando una aplicación escrita en algún lenguaje de programación

Php,
ASP,
Ruby,
Python,
Java,
Perl,
Javascript

Es común que en este proceso el servidor tenga que conectarse con bases de datos

MySQL
PostgreSQL
Oracle
Microsoft SQL
server

Front-end y Back-end



Front-end

Es la parte de un sitio web que interactúa con los usuarios. Se ejecuta del lado del cliente y cualquier persona puede verlo (abierto)

HTML, CSS y
Javascript

Back-end

Lo que se ejecuta desde el Backend y que el usuario no ve, está oculto y generalmente se basa en lenguajes interpretados

Javascript, Php,
Ruby, Python

HTML, CSS & Javascript

HTML -> Lenguaje de marcas de hipertexto
HyperText Markup Language

HTML sólo sirve para indicar cómo va ordenado el contenido de una página web a través de etiquetas (tags)

CSS -> Hojas de estilo en cascada
Cascading Style Sheets

Es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en algún lenguaje de marcado como HTML

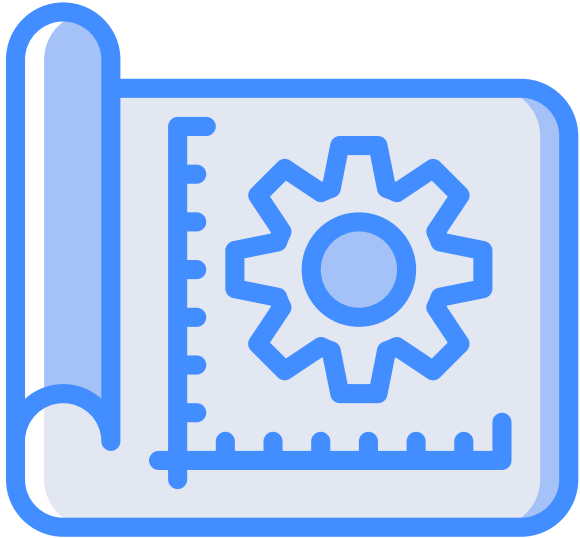
Modifica el diseño de forma que se ajusta al tamaño de la pantalla (responsive design – Media queries)

Javascript -> Lenguaje de programación interpretado (traducen lenguajes compilados)

Se implementa como parte del navegador desde el lado del cliente permitiendo mejoras en la interfaz de usuario y páginas web dinámicas

Manipula los componentes del DOM (DOM Manipulation)

Herramientas de diseño gráfico



Se utilizan muchas herramientas para incorporar imágenes, iconos e ilustraciones

Photoshop
Illustrator
Sketch

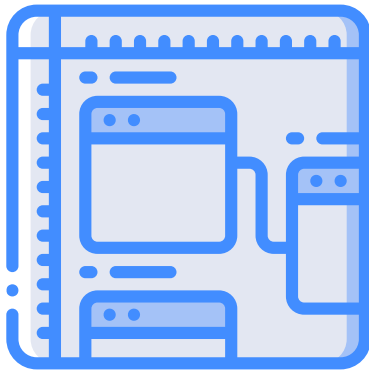
Existen también sistemas de diseño que integran elementos y hacen que CSS sea más fácil y rápido

Bootstrap
Foundation
Materialize

Arquitectura

Modelo - Vista - Controlador MVC

Base – Interfaz de usuario (GUI) – Contenido dinámico



Es la parte de la aplicación que contiene la lógica de la aplicación que maneja los datos de un software o un dispositivo y no es directamente accesible por los usuarios

Modelo de cómo los datos y los diferentes usuarios de la aplicación interactúan entre sí

Sistemas manejadores de bases de datos y modelo CRUD

Solicitudes: postman

Frameworks Moviles

Similar al desarrollo de aplicaciones web. A menudo son escritas específicamente para aprovechar las características únicas de los dispositivos móviles (Nativas)

Las híbridas aprovechan el navegador y son agnósticas al dispositivo

React Native
Angular
Ionic
VueJs