

# Conceptos, tecnologías y arquitectura para el desarrollo web

## Breve descripción:

En este componente formativo se abordarán los conceptos relacionados con el funcionamiento de un sistema web, soportado por la arquitectura de internet. A continuación, se revisarán las tecnologías más importantes para el desarrollo web, tanto del lado del cliente como del servidor. Por último, se explorarán los conceptos básicos de una arquitectura orientada a servicios.

---

**Junio 2024**

## Tabla de contenido

Introducción .....	4
1. Internet y la web .....	5
1.1. Fundamentos de la internet .....	5
Historia de la internet.....	5
Hardware de red .....	6
Redes de área local LAN .....	10
Modelo de referencia TCP/IP .....	11
Proceso de comunicación .....	13
Funcionamiento del modelo.....	13
Cuatro capas del modelo TCP/IP.....	13
Dirección IP.....	15
Sistema de Nombres de Dominio (DNS) .....	16
Protocolo HTTP.....	16
1.2. Arquitectura web .....	17
Arquitectura cliente/servidor .....	17
Cliente web.....	18
Servidor web .....	18
2. Introducción a las tecnologías para el desarrollo web .....	22

2.1. Tecnologías del front-end .....	23
HTML5 .....	23
CSS3 .....	24
JavaScript (JS) .....	25
2.2. Tecnologías del back-end .....	28
Lenguajes del servidor .....	29
Frameworks del back-end .....	30
3. Introducción a la arquitectura orientada a servicios .....	32
SOAP .....	33
REST .....	34
Síntesis .....	35
Material complementario .....	36
Glosario .....	37
Referencias bibliográficas .....	38
Créditos .....	40

## Introducción

En este componente formativo, se abordarán los conceptos fundamentales relacionados con la arquitectura de sistemas web. Se examinará la naturaleza de su comportamiento, así como los componentes físicos y lógicos esenciales para su operación. Además, se identificarán y caracterizarán las tecnologías web del lado del cliente y del servidor, avanzando hacia la comprensión de las tecnologías que facilitan la comunicación y transferencia de información o la creación de servicios con otros sistemas.

Este componente tiene como objetivo fomentar un análisis crítico para la elección de tecnologías específicas, dependiendo de los requerimientos y el alcance de un proyecto estimado. Este análisis se sustenta en la descripción y rasgos distintivos que se detallan en cada sección de este documento.

Se profundizará en la arquitectura de un sistema web, explorando cómo su comportamiento depende tanto de los componentes físicos como lógicos, y cómo estos facilitan la identificación y caracterización de tecnologías web tanto del lado del cliente como del servidor. Estas tecnologías permiten la comunicación y el traspaso de información, así como la generación de servicios con otros sistemas.

En este componente formativo, se desarrollarán conocimientos sobre la arquitectura web y el funcionamiento de internet. También se introducirán las tecnologías para el desarrollo web, como el front-end y el back-end, y la arquitectura orientada a servicios, incluyendo SOAP, REST y API-REST.

## **1. Internet y la web**

El desarrollo de aplicaciones y sistemas web ha sido una tendencia predominante en las últimas tres décadas, y sus mejoras han evolucionado en consonancia con las innovaciones en la infraestructura tecnológica que las respalda. Actualmente, las aplicaciones son notablemente más interactivas, rápidas, seguras y fáciles de usar. Un ejemplo de esto es el acceso a sitios web desde dispositivos móviles, la visualización de videos en YouTube, las conversaciones a través de WhatsApp, o el streaming de series en Netflix. Todos estos avances son posibles gracias a una robusta infraestructura de hardware que permite el despliegue no solo de aplicaciones sino también de servicios que se utilizan diariamente. Por ello, es crucial comprender los componentes fundamentales que enriquecen y otorgan rigurosidad a la construcción de este tipo de sistemas.

### **1.1. Fundamentos de la internet**

Internet emergió ante la necesidad de comunicación e interacción con personas en cualquier parte del mundo. Esta tecnología se considera el pilar de la web y se define como la infraestructura técnica que la hace posible. En su forma más básica, se presenta como una vasta red de computadoras comunicándose simultáneamente.

### **Historia de la internet**

La historia de la internet se remonta a inicios de la década de 1960 con un proyecto de investigación denominado ARPANET, financiado por el ejército de los Estados Unidos. En la década de 1980, se transformó en una infraestructura pública con el apoyo de diversas instituciones del sector público y privado. Las tecnologías que

soportan la internet han evolucionado a lo largo del tiempo para satisfacer la creciente demanda de la población, aunque la forma en que funciona ha cambiado poco.

Internet se presenta como un medio de conectividad entre todos los equipos de cómputo y, debido a su distribución y organización, se busca continuamente la mejor manera de mantener conectados a los usuarios.

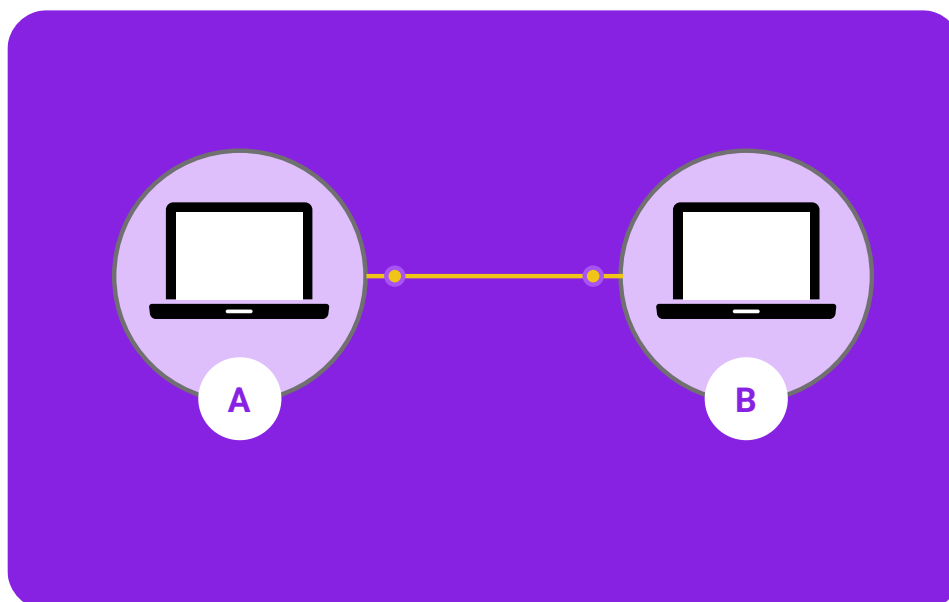
## Hardware de red

Cuando dos computadoras necesitan comunicarse, pueden vincularse mediante medios físicos como el cable Ethernet, o de manera inalámbrica utilizando tecnologías como WiFi, Bluetooth, ZigBee, entre otras.

- **Ejemplo de conexión entre dos equipos o terminales**

La mayoría de las computadoras de hoy en día pueden soportar cualquiera de este tipo de conexiones. A continuación, se presenta un ejemplo de conexión vía Ethernet entre dos equipos.

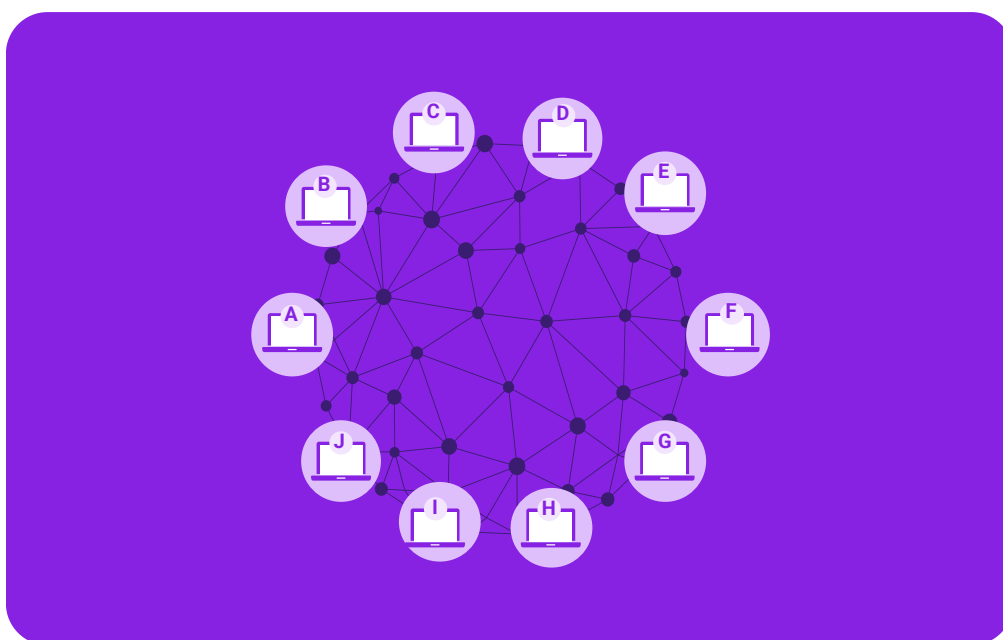
**Figura 1.** Conexión vía Ethernet entre dos equipos



- **Ejemplo de conexión entre diez equipos o terminales**

No se puede pensar que la red solo se limita a dos ordenadores, se pueden conectar tantos como se deseen, aunque con mayor complejidad. Por ejemplo, si se quieren conectar 10 computadoras, se necesitarían 45 cables y unos nueve conectores por computador, como se alcanza a apreciar en la siguiente figura.

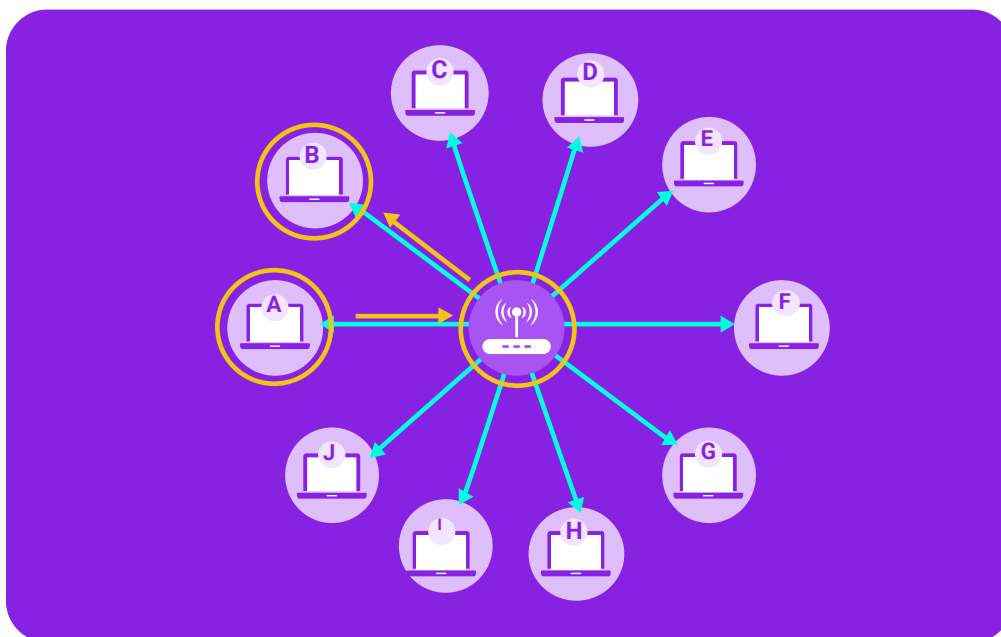
**Figura 2.** Conexión entre diez equipos o terminales



- **Ejemplo de conexión de diez equipos con un enrutador**

Por ejemplo, para que el terminal B reciba un mensaje desde el terminal A, este debe enviarlo primero al router, quien a su vez lo remite al ordenador B asegurándose que dicho mensaje no sea entregado a otro ordenador. Una vez incorporado un enrutador al sistema, la red de diez terminales solo requiere diez cables: un conector para cada ordenador y un enrutador con 10 conectores, como se aprecia en la siguiente gráfica.

**Figura 3.** Conexión de diez equipos con un enrutador

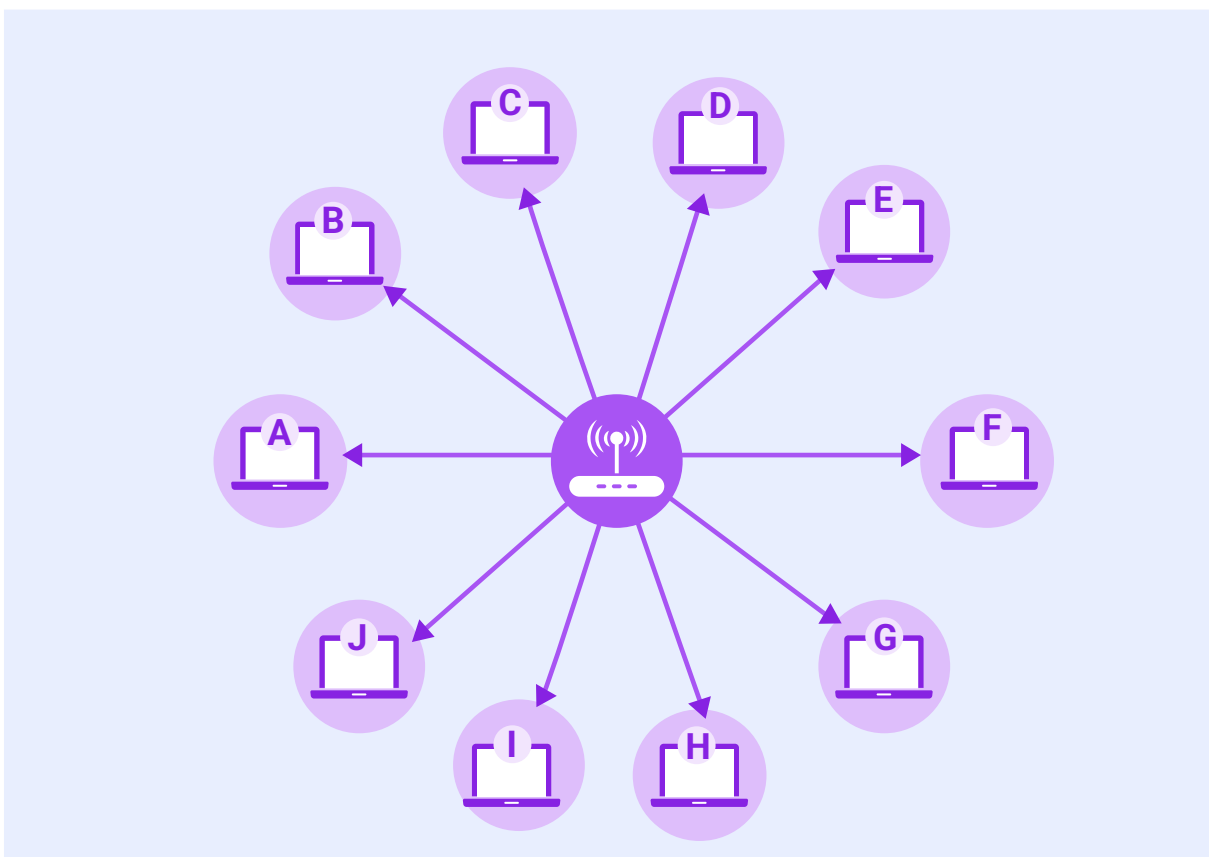


Reflexionando sobre el ejemplo anterior, surge la pregunta: ¿qué ocurre al conectar cientos, miles o millones de ordenadores entre sí? Claramente, un solo enrutador no puede gestionar una escala tan amplia. Aunque un enrutador funcione similarmente a un pequeño computador, esto no implica que conectar múltiples enrutadores sea una tarea sencilla.

El proceso de conexión entre varios terminales puede ser muy complejo y laborioso. Por ello, para organizar estas conexiones de manera eficiente, cada ordenador en una red está conectado a un dispositivo especializado denominado enrutador o router. Este dispositivo actúa de manera similar a los señalizadores en las estaciones del sistema integrado de transporte público, dirigiendo el tráfico para garantizar que llegues a tu destino correctamente. Así, el router asegura que los mensajes enviados desde un computador o terminal emisor alcancen el destino adecuado.



**Figura 4.** Conector para cada ordenador y enrutador con 10 conectores

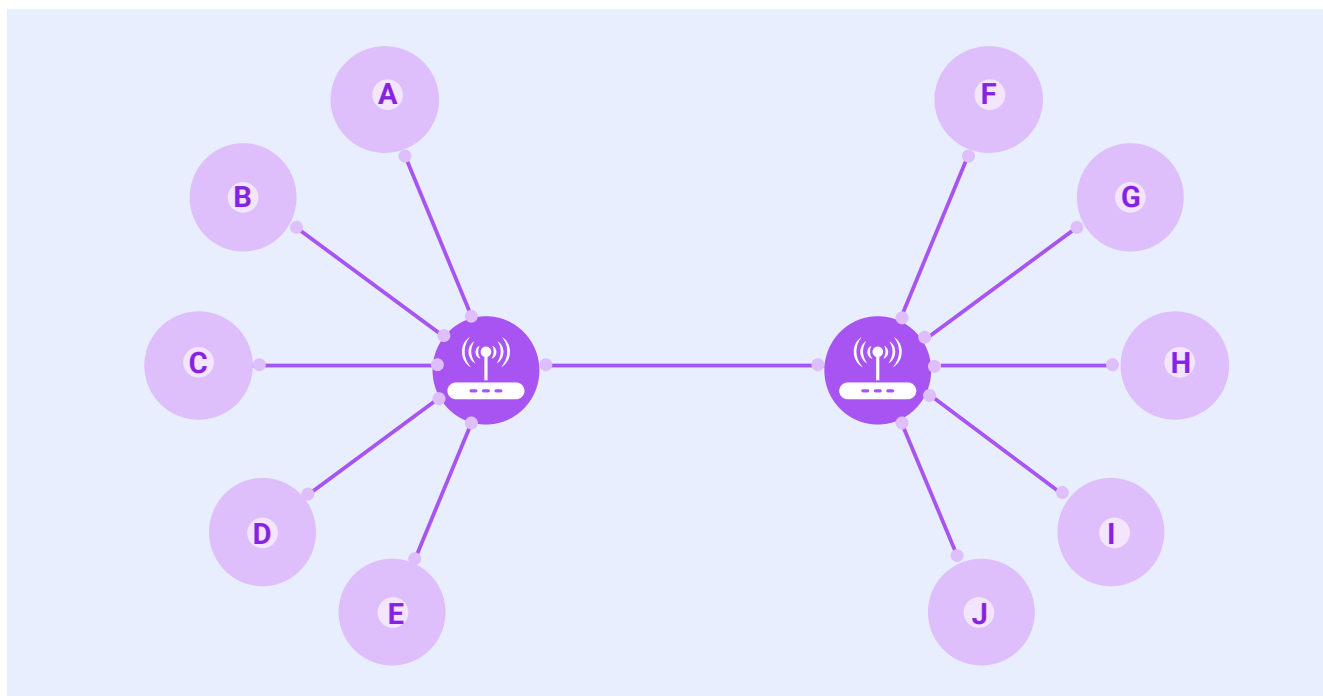


Al incorporar un enrutador al sistema, una red de diez terminales solo necesitaría diez cables: un conector para cada ordenador y un enrutador equipado con 10 conectores.

### **Ejemplo de conexión de diez equipos con dos enrutadores**

En ese sentido, estas conexiones entre terminales generan redes de datos de transmisión como la recepción de información, por lo que es necesario caracterizar estas redes, las cuales son importantes para la comprensión del funcionamiento de un sistema web. Entre ellas se encuentran las redes de área local, una red de área metropolitana y una red de área amplia.

**Figura 5.** Conexión de diez equipos con dos enrutadores

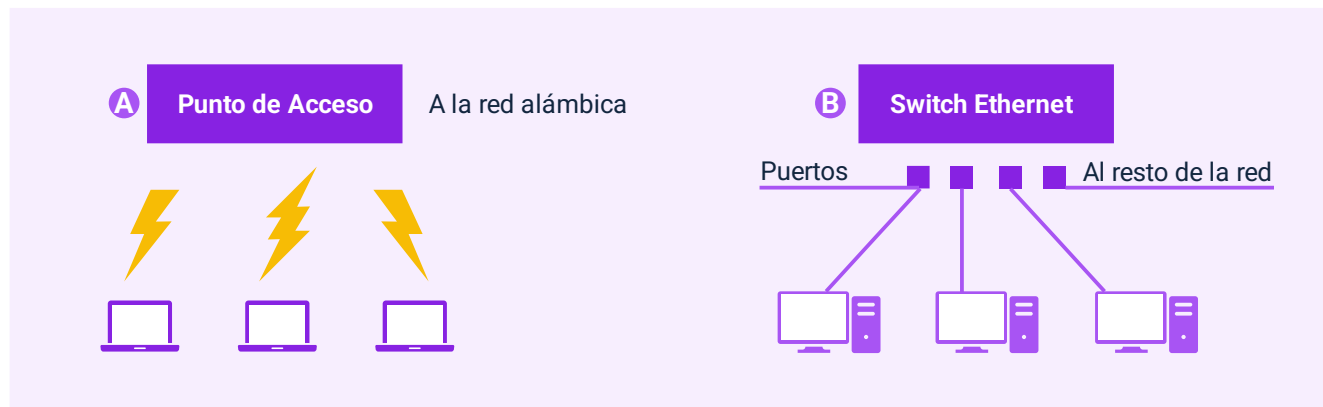


## Redes de área local LAN

Generalmente llamadas LAN (Local Area Networks), son redes de naturaleza privada que operan dentro de un solo establecimiento, como una casa, oficina o fábrica. Las redes LAN se utilizan ampliamente para conectar computadoras personales y electrodomésticos, con el fin de compartir recursos (por ejemplo, impresoras) e intercambiar información.

A continuación, se presenta un ejemplo de esta red:

**Figura 6.** Red de área local inalámbrica y cableada



El diseño físico y lógico de una red, conocido como topología, se basa frecuentemente en enlaces de punto a punto para redes alámbricas. El estándar IEEE 802.3, más conocido como Ethernet, es la forma más común de establecer una LAN (Red de Área Local) alámbrica. La figura en su apartado b ilustra un ejemplo de una topología de Ethernet conmutada, donde cada computadora se comunica a través del protocolo Ethernet y se conecta a un dispositivo conocido como switch mediante un enlace de punto a punto.

En este contexto, un switch dispone de varios puertos, cada uno de los cuales puede conectarse a una computadora. La función principal del switch es transmitir paquetes entre las computadoras que están conectadas a él. Utiliza la dirección incluida en cada paquete para determinar a qué computadora debe enviarlo, tal como se presenta en el ejemplo de la red de área cableada mencionado.

## Modelo de referencia TCP/IP

Al igual que los seres humanos requieren de un lenguaje común para comunicarse, es esencial que todos los equipos de cómputo dispongan de un modo

estandarizado para interactuar entre ellos. Para la mayoría de los equipos modernos, este estándar es el TCP/IP. Generalmente, el TCP/IP viene preinstalado en los dispositivos y opera de manera automatizada. Sin embargo, es crucial entender el modelo TCP/IP, especialmente si se necesita configurar un equipo para conectarlo a otro sistema. (Fisher, 2021).

El protocolo TCP/IP, cuyas siglas significan Transmission Control Protocol/Internet Protocol (Protocolo de Control de Transmisión/Protocolo de Internet), constituye un conjunto de reglas estandarizadas que facilitan la comunicación de los equipos en redes como Internet. La eficacia de estas máquinas radica en su velocidad de transmisión y en su capacidad para comunicarse con otras, un aspecto fundamental en la era de la cuarta revolución industrial, caracterizada por la interacción inteligente de equipos, especialmente en contextos de producción.

La dependencia de nuestras actividades diarias, como enviar correos electrónicos, ver Netflix o recibir indicaciones de navegación, se basa en la interacción continua entre distintos dispositivos. Estos equipos pueden ser de diferentes fabricantes y ubicarse en diversas zonas geográficas. Así, tanto las personas como las plataformas de software que los utilizan pueden emplear distintos lenguajes, tanto humanos como informáticos.

El blog de Avast Academy señala que una interacción determinada puede ocurrir entre dos sistemas informáticos o involucrar a cientos de ellos. Sin embargo, al igual que el proceso de pasar una carta o un paquete de mano en mano, cada transacción se realiza entre solo dos equipos a la vez. Para que esta comunicación sea efectiva, ambos equipos deben conocer de antemano cómo se espera que se comuniquen.

## **Proceso de comunicación**

Según (Fishre,2021), la comunicación nace a partir de las siguientes incógnitas:

- ¿Cómo inician la conversación?
- ¿A quién le toca comunicarse?
- ¿Cómo sabe un equipo si su mensaje se ha transmitido correctamente?
- ¿Cómo terminan la conversación?

De esta manera, es evidente que los equipos necesitan reglas, instrucciones o en su término técnico, protocolos para la comunicación y el comportamiento entre esas dos instancias.

## **Funcionamiento del modelo**

El funcionamiento de este protocolo radica en descomponer cada mensaje en paquetes. Porque el objetivo no es volver a enviar de nuevo y desde cero el mensaje completo, toda vez que haya un fallo. Así que estos paquetes se vuelven a ensamblar en el otro extremo. De hecho, cada paquete podría tomar una ruta diferente hasta el equipo de destino, toda vez sí la ruta deja de estar disponible o está muy congestionada. De manera adicional, TCP/IP divide las distintas tareas de comunicación en capas. Así, cada capa tiene una función diferente. Los datos pasan por cuatro capas independientes antes de recibirse en el otro extremo.

## **Cuatro capas del modelo TCP/IP**

En la literatura, se encuentran un sin número de definiciones alrededor de las capas de este modelo, entre las más importante y haciendo un resumen para dar mayor claridad se tienen:

- **Capa de internet**

La capa de internet (también denominada capa de red), es la capa encargada de controlar el movimiento (envío y recepción) de los paquetes alrededor de la red.

- **Capa de aplicación**

La capa de aplicaciones es el grupo de aplicaciones que requiere comunicación de red. Es con lo que el usuario o cliente suele interactuar, como el correo electrónico y la mensajería. Como la capa inferior es la encargada de gestionar los detalles de la comunicación, no tienen que preocuparse por ello.

- **Capa de enlace de datos**

La capa de enlace de datos (igualmente denominada como capa de enlace, capa de interfaz de red o capa física) es la encargada de manejar las partes físicas del envío y recepción de datos mediante el cable Ethernet, la tarjeta de interfaz de red, la red inalámbrica, el controlador del dispositivo en el equipo, entre otros.

- **Capa de transporte**

La capa de transporte es la capa encargada que proporciona una conexión de datos fiable entre dos dispositivos. Divide los datos en paquetes, hace acuse de recibo de los paquetes que recibe del otro dispositivo y además se asegura de que el otro dispositivo haga el acuse de recibo pertinente de los paquetes que recibe a su vez.

La siguiente tabla presenta las capas y sus respectivos protocolos.

**Tabla 1.** Principales protocolos del Modelo TCP/IP por capas

Modelo TCPS / IP	Suite de protocolos (Principales)	Suite de protocolos (Principales)	Suite de protocolos (Principales)	Suite de protocolos (Principales)
Capa de Aplicación	SSH	FTP	SMTP	DHCP
Capa de aplicación	DNS	RIP	SNMP	HTTP
Capa de transporte	TCP	DCCP	UTP	N/A
Capa de transporte	UDP	ICMP	FCP	N/A
Capa de Internet	IP	ICMP	N/A	N/A
Capa de Internet	IPSEC	IGMP	N/A	N/A
Capa de interfaz de Red	ARP	L2TP	N/A	N/A
Capa de interfaz de Red	NDP	ETHERNET	N/A	N/A

## Dirección IP

Para comunicarse con una computadora específica en una red, es necesario especificar su dirección. Por esta razón, cada computadora conectada a una red tiene una dirección única que la identifica de manera lógica y jerárquica, conocida como “dirección IP” (Internet Protocol). Esta dirección consta de una serie de cuatro números separados por puntos, por ejemplo: 192.168.2.19. Aunque para las computadoras este identificador es simple, los humanos pueden tener dificultades para recordarlo y memorizarlo. Para facilitar la asociación de esta serie numérica, se utilizan los nombres de dominio.

## **Sistema de Nombres de Dominio (DNS)**

Los DNS funcionan como una libreta de direcciones de sitios web. Cuando se introduce una dirección web en el navegador, esta consulta los DNS antes de recuperar el sitio web. El navegador necesita determinar en qué servidor se aloja el sitio web para enviar los mensajes HTTP correctamente. Significa que es análogo, el buscar la dirección de una tienda para poder visitarla.

Por ejemplo, google.com es el nombre de dominio que se utiliza para representar la dirección IP 173.194.121.32, facilitando la identificación de un ordenador en internet.

## **Protocolo HTTP**

El Protocolo de Transferencia de Hipertexto (HTTP) es un protocolo de aplicación que establece un lenguaje para la comunicación entre clientes y servidores. Es similar al idioma que se usa para hacer un pedido en una tienda. HTTP permite solicitar datos y recursos, como documentos HTML, y es la base de cualquier intercambio de datos en la web. Funciona con una estructura cliente-servidor, lo que significa que una petición de datos es iniciada por el elemento que recibirá los datos (el cliente), generalmente un navegador web.

Se han abordado conceptos clave sobre los componentes que conforman la arquitectura de internet. Como se ha explicado, internet es una infraestructura técnica que permite la conexión de miles de millones de computadoras. Algunas de estas, conocidas como servidores web, son capaces de enviar mensajes inteligibles a los navegadores. Así, internet constituye la infraestructura, mientras que la web es un



servicio construido sobre ella. Además, existen otros servicios soportados por internet, como el correo electrónico.

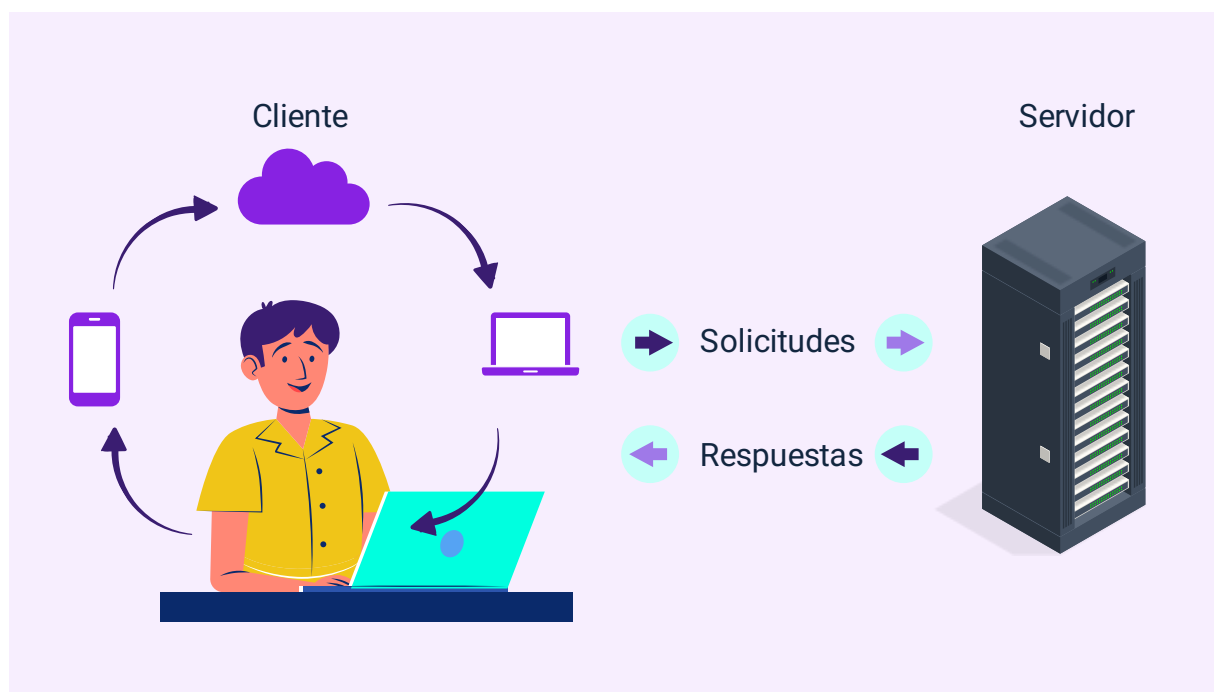
## **1.2. Arquitectura web**

Para comprender el funcionamiento de un sistema, plataforma o aplicación web, es fundamental conocer los conceptos y el comportamiento de las tecnologías que se integran en esta arquitectura. A continuación se detalla:

### **Arquitectura cliente/servidor**

Una aplicación web opera bajo una arquitectura simplificada conocida como cliente/servidor. En este modelo, el "cliente" generalmente se refiere a un navegador web instalado en un dispositivo del usuario, que se encarga de mostrar y solicitar información o documentos a través de una red. Las computadoras conectadas a la web se clasifican como clientes y servidores. Un diagrama simplificado de cómo interactúan se podría visualizar de la siguiente manera:

**Figura 7.** Arquitectura web



## Cliente web

El cliente web es una aplicación ejecutable, usualmente preinstalada en el dispositivo del usuario, que facilita el acceso a internet. Este cliente es capaz de mostrar y solicitar documentos sobre una red, siendo así responsable de la capa de presentación. Los clientes web son dispositivos de usuarios conectados a internet (por ejemplo, un ordenador conectado a la red Wi-Fi o un teléfono conectado a la red de telefonía móvil) y el software disponible que permite acceder a internet en dichos dispositivos (normalmente, un navegador web como Firefox o Chrome).

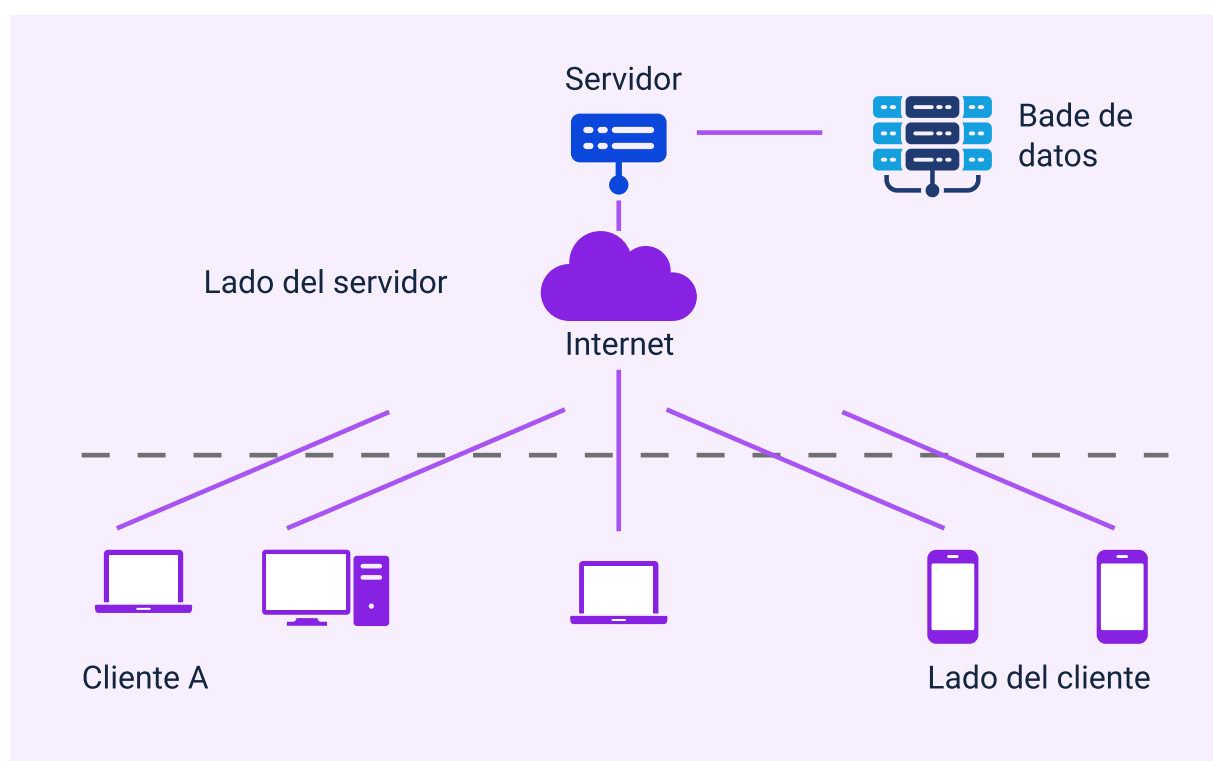
## Servidor web

En la arquitectura web, el servidor web juega un papel crucial, pues se encarga de atender las solicitudes enviadas por los clientes a través de los navegadores web. Es en el servidor donde se realiza todo el procesamiento de las aplicaciones y la gestión de los datos. Los servidores son computadoras que almacenan páginas web, sitios o

aplicaciones. Cuando un dispositivo cliente desea acceder a una página web, una copia de esta página se descarga desde el servidor al equipo cliente y se presenta en su navegador web. En esencia, un servidor es una computadora potente que almacena y transmite datos vía internet.

A continuación, se presenta un esquema que facilitará la comprensión de los diversos aspectos relacionados con esta arquitectura.

**Figura 8.** Arquitectura cliente servidor

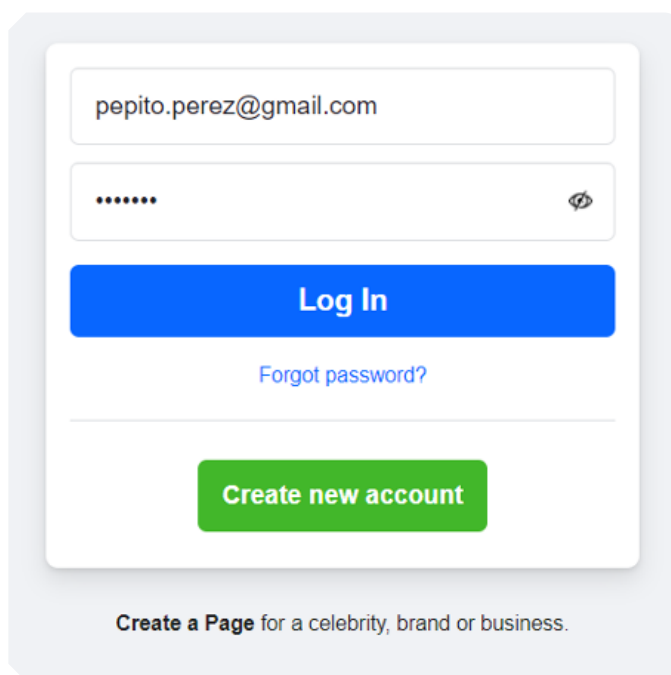


Se inicia con un ejemplo en el que se tiene al cliente A, conforme se presenta en el esquema anterior. Este cliente, que es una computadora o un ordenador de escritorio, cuenta con un explorador web instalado. El usuario, Pepito Pérez, necesita ingresar a Facebook. Para acceder a la red social, es necesario registrar las credenciales, como el login y la contraseña. Al hacer clic en iniciar sesión, se genera inmediatamente

una petición que viaja desde el cliente hasta el servidor, tal como se ilustra en el esquema de la arquitectura.

El servidor, por su parte, procesa los datos de Pepito y verifica la información con otros servidores y bases de datos, como se observa en la figura. Dependiendo de si la validación de los datos es afirmativa o no, el servidor enviará una respuesta al usuario, permitiéndole o negándole el acceso a su espacio en la red social. Esta respuesta del servidor se transmite en un lenguaje y con recursos que son interpretados y manejados por el explorador web para ser mostrados en pantalla. Esto indica que existen tecnologías tanto del lado del cliente como del lado del servidor, por lo que el servidor, además de procesar los datos, debe responder en el formato o lenguaje adecuado para que el cliente pueda interpretarlo correctamente.

**Figura 9.** Cuadro de login y password de Facebook



The image shows a screenshot of the Facebook login and registration interface. It features a white card with rounded corners on a light gray background. At the top, there is a text input field containing the email address 'pepito.perez@gmail.com'. Below this is a password input field represented by a series of dots, with an eye icon to its right for toggling visibility. A prominent blue 'Log In' button is centered below the password field. Underneath the button is a blue link that says 'Forgot password?'. A horizontal line separates the login section from the registration section, which contains a green 'Create new account' button. At the very bottom of the card, there is a link that says 'Create a Page for a celebrity, brand or business.'

Este ejemplo se puede realizar desde cualquier otro dispositivo, es por eso por lo que la arquitectura web es considerada como un ambiente multiplataforma y distribuido.

Finalmente, es importante tener en cuenta que una web sigue las siguientes pautas:

- **¿Cómo se realiza la transferencia de información?**

Cada solicitud y respuesta se envía utilizando el protocolo HTTP. Es mediante este protocolo que se transmiten los datos.

- **¿Cómo se describe la información apenas llega al cliente como respuesta?**

Para la presentación de la información web se utilizan tecnologías del lado del cliente, es decir, HTML, CSS y JavaScript en sus versiones más recientes. Estos lenguajes de programación y diseño permiten estructurar el contenido, definir su estilo y manejar comportamientos dinámicos en la página.

- **¿Cómo se localiza la información?**

La información se localiza mediante la dirección URL, la cual se compone de tres partes: primero el protocolo, segundo la ruta del servidor y tercero el dominio. La estructura queda de la siguiente forma:

<http://rutadelserveridor/ubicacion/dominio>. Esta estructura asegura que el navegador pueda identificar y acceder a la ubicación exacta del recurso solicitado en la web.

## **2. Introducción a las tecnologías para el desarrollo web**

Las tecnologías de desarrollo web abarcan una amplia gama de lenguajes y herramientas de programación que se utilizan para crear sitios web, aplicaciones dinámicas y plataformas, dotadas de todas las funciones necesarias para satisfacer una diversidad de necesidades y requerimientos de los usuarios. Estas tecnologías están en constante evolución; no es necesario realizar un análisis exhaustivo sobre su estado de transformación para notar el creciente auge de diversos lenguajes de programación que facilitan la realización de acciones más complejas y efectivas en los sitios web. Por esta razón, las tecnologías web son consideradas multiplataforma, una característica que se resalta en la afirmación del grupo de expertos de Ingenio Virtual.

... Los modelos y tecnologías de desarrollo web han evolucionado significativamente en la última década. Existen multitud de aplicaciones, frameworks, librerías, arquitecturas y sistemas de publicación que en sus diferentes versiones reciben actualizaciones y mejoras continuas. El progreso también se ha observado en la administración de sistemas, los servicios de alojamiento, técnicas de escalabilidad, monitorización y gestión de centros de procesos de datos. Esta evolución ha propiciado la convergencia de una gran cantidad de tecnologías, herramientas y estilos arquitectónicos para el desarrollo de sitios web y aplicaciones. (Ingenio Virtual, 2018).

Esta tendencia subraya la importancia de mantenerse actualizado con las últimas tecnologías y enfoques en el campo del desarrollo web, para poder aprovechar al máximo las capacidades que ofrecen estas herramientas modernas.

## 2.1. Tecnologías del front-end

Las tecnologías de front-end se utilizan para el desarrollo del "lado cliente" de un sitio web o aplicación. Estas tecnologías son responsables de desarrollar los componentes interactivos y producir los elementos visuales con los que los usuarios interactúan, como colores, estilos de texto, imágenes, botones y menús de navegación, entre otros objetos interactivos. Estas tecnologías y lenguajes están implementados en los diferentes navegadores, facilitando así su uso y compatibilidad.

### HTML5

HTML, abreviatura de Hyper Text Markup Language, es una de las tecnologías esenciales para el desarrollo de todo tipo de aplicaciones web. Proporciona la estructura base para una página web. El código HTML garantiza que el contenido de un sitio web esté adecuadamente formateado y estructurado, permitiendo que cada navegador de Internet muestre el contenido como está previsto. Sin HTML, los navegadores no podrían mostrar texto, cargar imágenes ni otros elementos esenciales.

HTML5, la versión más reciente de HTML, incluye un conjunto más amplio de tecnologías que permiten a los sitios web y a las aplicaciones ofrecer más diversidad y potencia. Este conjunto es conocido como HTML5 y ofrece una gran variedad de recursos, incluyendo librerías, plugins y APIs que se pueden utilizar junto con lenguajes de programación como JavaScript para crear sitios web más interactivos y dinámicos.

Algunas de las tecnologías más utilizadas en HTML5 incluyen:

- **Canvas**

Es un elemento HTML5 que se utiliza para dibujar imágenes y formas con el fin de manipularlas. También se puede utilizar para casos como gráficos y animaciones de juegos.

- **Almacenamiento web**

Se utiliza para almacenar información directamente en el navegador.

Algunos ejemplos de esto serían: almacenar la información de inicio de sesión del usuario y guardar las preferencias del usuario para un sitio web.

- **Web Workers**

Esta tecnología habilita un script que sigue ejecutándose en segundo plano cuando se abre una página web y se usa principalmente en sitios web con capacidades sin conexión. Hace que las páginas estén disponibles sin conexión y permite el uso de notificaciones como web Push. Puede enviar estas notificaciones incluso cuando su navegador no está abierto.

- **Web Sockets**

Permite una conexión bidireccional persistente entre el usuario y el servidor. Los casos de uso más comunes incluyen chats y notificaciones en aplicaciones web.

## **CSS3**

Hojas de Estilo en Cascada, conocidas por sus siglas en inglés como Cascading Style Sheets (CSS), son el lenguaje de estilos utilizado para describir la presentación de documentos HTML o XML. CSS especifica cómo deben ser renderizados los elementos estructurados, es decir, se encarga de la apariencia del sitio web.



- **Cascading**

Se refiere a que los estilos aplicados a los elementos de una página web se propagan a los elementos contenidos en estos, aplicándose en cascada.

- **Style**

A través de la tecnología CSS, se aplican estilos visuales a los distintos elementos de la página web.

- **Sheets**

Traducido como hojas, porque los estilos de una página web generalmente se añaden en archivos separados. Estos archivos deben tener la extensión .css.

Desde la introducción de CSS3, el alcance de las especificaciones se ha incrementado de forma significativa, y el desarrollo de los distintos módulos de CSS ha comenzado a mostrar varias diferencias. Con CSS3, se ofrece un mayor control sobre los elementos del sitio, gracias a sus nuevas funciones y atributos aplicables a las etiquetas HTML, lo que maximiza las acciones visuales del sitio o plataforma.

## **JavaScript (JS)**

Existen numerosas definiciones de este lenguaje de programación en la literatura. Sin embargo, una definición particularmente notable se encuentra en el blog de Rafa Ramos, que afirma literalmente:

JavaScript es el lenguaje de programación encargado de dotar de mayor interactividad y dinamismo a las páginas web. Cuando JavaScript se ejecuta en el navegador, no necesita de un compilador. El navegador lee directamente el código, sin necesidad de terceros. Por tanto, se le reconoce como uno de los tres lenguajes nativos

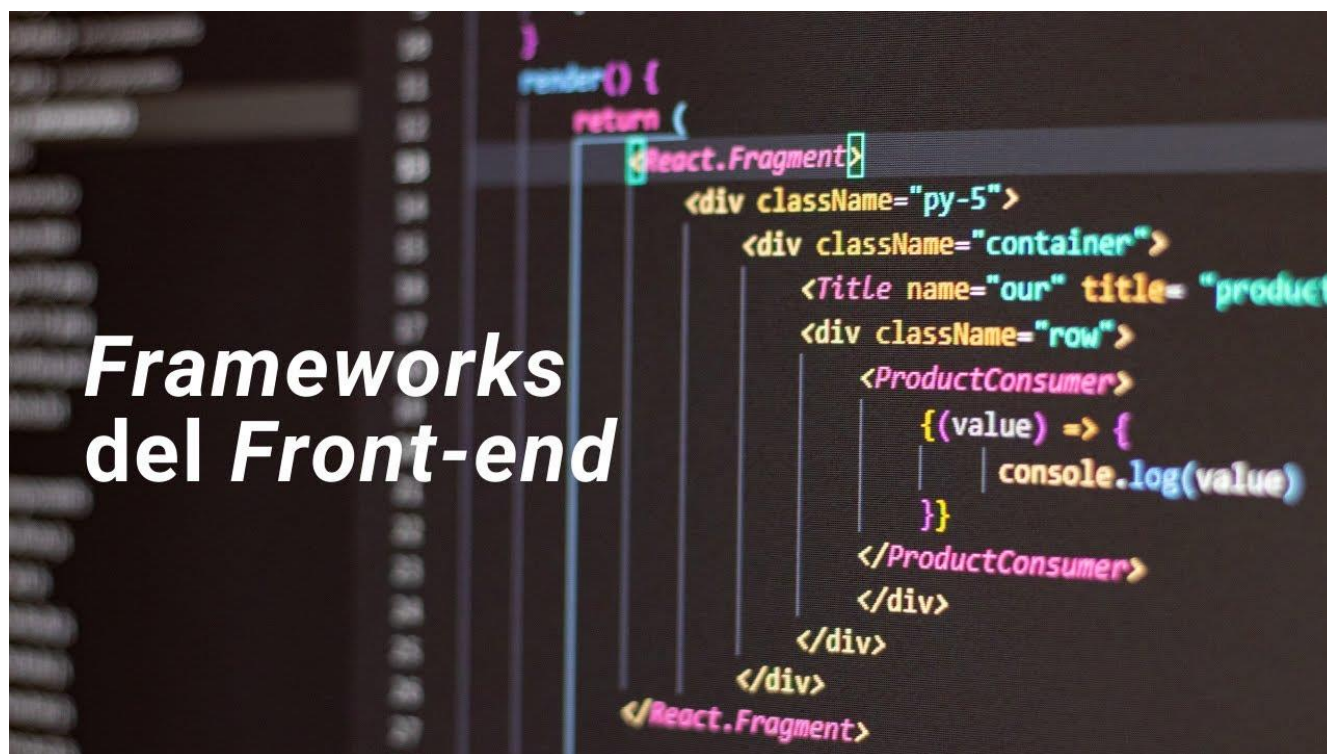
de la web junto a HTML (contenido y su estructura) y a CSS (diseño del contenido y su estructura). (Ramos, 2020).

Además, aunque JavaScript es más conocido como un lenguaje de scripting para páginas web, también se utiliza en muchos entornos fuera del navegador, como en Node.js, Apache CouchDB y Adobe Acrobat.

Para concluir sobre estas tres tecnologías principales, se realiza una analogía con el cuerpo humano: los huesos, que constituyen la estructura ósea, se comparan con HTML, que estructura el contenido de la web; los músculos, que permiten el movimiento y la dinámica del cuerpo, se comparan con JavaScript, que añade interactividad a las páginas web; y la piel, que cubre y da apariencia al cuerpo, se compara con CSS, que estiliza y diseña el contenido web. Esta analogía tiene como objetivo facilitar la comprensión de estas tecnologías del lado del cliente.

Para continuar, lo invitamos a explorar el siguiente video:

## Video 1. Frameworks del Front-end



[Enlace de reproducción del video](#)

### Síntesis del video: Frameworks del Front-end

Un framework es un tipo de herramienta o recurso que hace que el desarrollo de cualquier aplicativo sea del lado del cliente o servidor sea más rápido fácil y robusto. Este tipo de herramientas incorporan un conjunto de librerías, materiales y adoptan una estructura que van a permitir que todas las aplicaciones tengan un mejor rendimiento; en ese sentido se lista los frameworks más importantes a la fecha con el fin de que se tenga un reconocimiento para usos futuros; este listado es apoyado según la comunidad más grande de gestión de productos llamada Product Coalition:

React, está hecho por un desarrollador de Facebook Jordan Walke, la biblioteca se utilizó por primera vez en el servicio de noticias de Facebook en 2011, luego en instagram en 2012 y en 2013 el código se abrió para todos.

Vue Js, más de 36,000 aplicaciones están escritas en View, entre las empresas que lo eligieron como herramienta de desarrollo se encuentran: Stackoverflow y PlayStation; según lo define la comunidad de Product Coalition.

Angular, es en realidad un proyecto de Google que se utiliza principalmente para el desarrollo de aplicaciones móviles y de escritorio, hay muchas funciones útiles, el sitio oficial de frameworks dice: concéntrese en crear sitios impresionantes en lugar de hacer que su código funcione.

## **2.2. Tecnologías del back-end**

Las tecnologías del back-end se refieren a aquellas utilizadas en el lado del servidor. El servidor emplea estas tecnologías para gestionar y procesar las diferentes peticiones de información que recibe. Además, realiza la gestión de las bases de datos alojadas en el mismo. Una vez que la información es procesada, se envía de vuelta al dispositivo del usuario para que sea visualizada a través de las tecnologías front-end. Entre las tecnologías de back-end más conocidas y utilizadas se encuentran PHP, Java, .NET, Python y MySQL. Estas herramientas son esenciales para el desarrollo y la operación efectiva de aplicaciones web, ya que permiten la creación de funcionalidades complejas y manejo de grandes volúmenes de datos.

## Lenguajes del servidor

En el lado del servidor, el back-end es fundamental para el funcionamiento de un sitio web, ya que es en esta instancia donde se realizan los procesos más robustos de manejo de información. Por ello, se requieren lenguajes de programación que ofrezcan una diversidad de funciones para el tratamiento adecuado de esos datos. Además, los programas escritos por desarrolladores de back-end se utilizan para comunicar la información desde la base de datos hasta el navegador. La empresa de base tecnológica Third Rock Techkno señala que:

“En 2021, el desarrollo de Back-End será aún más vital a medida que las empresas busquen expandirse a un ritmo rápido”. Esta empresa también destaca los lenguajes de programación del servidor que están en tendencia, subrayando la importancia de mantenerse actualizado con las tecnologías que lideran el mercado. (Third Rock Techkno, 2021)

Los lenguajes de programación son:

- **Node.JS**

Este lenguaje tiene su propio entorno de ejecución. Es popular entre la comunidad de desarrolladores y se considera uno de los lenguajes más eficientes en la literatura de desarrollo de software. Se prefiere para desarrollar aplicaciones de alta gama para el desarrollo de sitios web, así como para la transmisión de video.

- **Kotlin**

Se presenta como un lenguaje de programación back-end para Java Virtual Machine. Es muy preferido para el desarrollo de Android, contando con el soporte oficial de Google. Según Third Rock Techkno, tiene una

curva de aprendizaje muy fácil, es menos propenso a fallos y facilita la detección de errores.

- **Ruby**

Se estima que este lenguaje jugará un papel fundamental en el futuro, ya que ayuda a reducir el tiempo dedicado a tareas estandarizadas.

- **PHP**

Es un lenguaje de programación de back-end muy flexible con características impecables. Tiene una comunidad en constante actualización y numerosos frameworks basados en este lenguaje. Se considera el futuro del desarrollo back-end en 2021, destacando por una sintaxis muy expresiva que brinda libertad creativa a los desarrolladores.

## **Frameworks del back-end**

Al igual que los frameworks del front-end, los frameworks del back-end proporcionan a los desarrolladores herramientas para realizar su trabajo de manera más eficiente, segura y robusta. Estos frameworks ofrecen estructuras predefinidas y componentes reutilizables que aceleran el proceso de desarrollo, ayudando a manejar aspectos complejos como la conexión con bases de datos, la gestión de sesiones, el procesamiento de transacciones y la seguridad.

Una de las empresas de desarrollo web más destacadas de Canadá, Monocubed, señala la importancia de estos frameworks en el desarrollo actual. Según Mistry (2021), los frameworks de back-end que están marcando tendencia facilitan significativamente la creación de aplicaciones complejas y escalables. Estos frameworks no solo mejoran la eficiencia del desarrollo sino que también aseguran

que las aplicaciones puedan adaptarse y responder a las exigencias del mercado moderno.

- **Laravel**

Este framework tiene una excelente estructura y es capaz de adaptarse a grupos grandes debido a la efectividad de sus recursos, como la llamada toolbox. Basado en PHP, sigue una sintaxis elegante y expresiva. Permite a los desarrolladores web ser flexibles y creativos, manejando los detalles de fondo a diferencia de otros frameworks.

- **Django**

Framework de alto nivel conocido por su enfoque de ‘baterías incluidas’, lo que significa que incluye casi todo lo que necesitaría cualquier desarrollador. Se utiliza principalmente para el desarrollo de aplicaciones web interactivas a gran escala impulsadas por bases de datos. Es altamente personalizable y escalable, con una amplia comunidad y una importante documentación.

- **Ruby on Rails**

Escrito en Ruby, este framework es fácil de probar y compilar. Cuenta con el amplio apoyo de una comunidad activa y una gran cantidad de recursos disponibles, lo que lo convierte en una alternativa viable para los principiantes en desarrollo.

### **3. Introducción a la arquitectura orientada a servicios**

Hasta ahora, se han explorado las tecnologías para el desarrollo web, pero es crucial entender que, además de crear aplicaciones, también es posible desarrollar servicios. Para ello, es esencial abordar el concepto de la Arquitectura Orientada a Servicios, conocida como SOA. SOA es un tipo de arquitectura de software que se fundamenta en la integración de aplicaciones a través de servicios. Según IBM, SOA se define como:

... una manera de hacer que los componentes de software sean reutilizables mediante interfaces de servicio. Estas interfaces utilizan estándares de comunicación que les permiten incorporarse rápidamente a nuevas aplicaciones sin necesidad de una integración profunda cada vez. (IBM, 2021).

En este contexto, es destacable el desarrollo de software en sí, la reutilización de componentes y la comunicación con distintas aplicaciones sin la necesidad de crear desarrollos a medida para cada aplicación con la que se desee conectar. De aquí surge el concepto de servicio y de interoperabilidad.

Dentro de las implementaciones de SOA, se encuentran las arquitecturas REST y SOAP, las cuales siguen los mismos principios de SOA. REST y SOAP son dos enfoques diferentes para la transmisión de datos en línea. Ambos definen cómo diseñar interfaces de programación de aplicaciones (APIS), que actúan como medios para que dos aplicaciones o plataformas se comuniquen entre sí, exponiendo o publicando un conjunto de servicios y reglas sobre cómo acceder a ellos. La principal diferencia entre ellos es que SOAP es un protocolo estructurado, mientras que REST no lo es.



Generalmente, uno de estos dos enfoques registrará las APIS, definiendo así una serie de reglas tanto para la publicación de servicios como para el acceso a los mismos, según el caso práctico y las preferencias del desarrollador.

## **SOAP**

SOAP, acrónimo de Simple Object Access Protocol (Protocolo Simple de Acceso a Objetos), se define como un protocolo o conjunto de reglas estándar diseñado principalmente para facilitar la comunicación entre aplicaciones desarrolladas en diferentes lenguajes de programación y que operan en distintas plataformas tecnológicas.

Debido a su naturaleza de protocolo, SOAP incorpora reglas integradas que, aunque aumentan la seguridad y la estructuración de los mensajes, también pueden incrementar la complejidad y la carga en los servidores, lo que a veces resulta en retrasos en los tiempos de carga de las páginas o en las respuestas. Sin embargo, debido a su robustez en estructura y seguridad, SOAP es ampliamente utilizado en el sector empresarial, donde estos factores son críticos. (RedHat, 2021).

Adicionalmente, el envío de una solicitud de datos a una API de SOAP puede gestionarse a través de diversos protocolos de la capa de aplicación, tales como HTTP, utilizado comúnmente por los navegadores web; SMTP, el protocolo para el envío de correos electrónicos; y TCP, entre otros. Una vez recibida una solicitud, los mensajes SOAP retornan por defecto documentos XML, un lenguaje de etiquetas que es comprensible tanto para máquinas como para personas, facilitando así la interoperabilidad entre sistemas dispares.

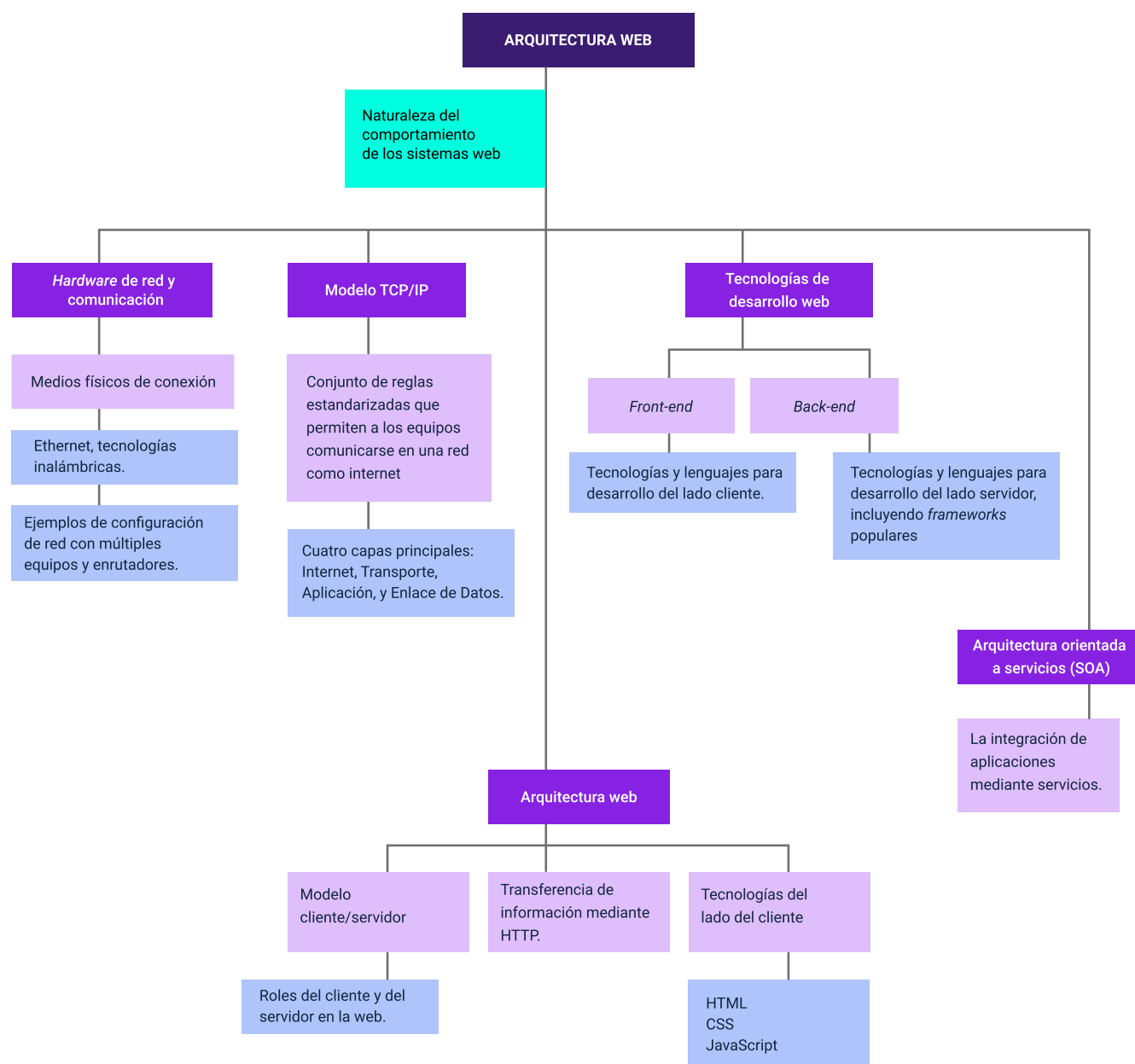
## REST

REST, acrónimo de Representational State Transfer (Transferencia de Estado Representacional), se define como un conjunto de principios arquitectónicos que se adaptan a las necesidades de los servicios web, y es especialmente valorado en aplicaciones móviles por la velocidad de transmisión y la ligereza de la información. Al enviar una solicitud de datos a una API REST, se utiliza el protocolo de transferencia de hipertexto, comúnmente conocido como HTTP. Las API diseñadas según los principios de REST (conocidas como API o servicios web RESTFUL) responden a estas solicitudes con mensajes en formatos como HTML, XML, JSON, entre otros. El formato más utilizado para los mensajes es la notación de objetos de JavaScript (JSON), que puede ser interpretado por cualquier lenguaje de programación, es liviano y comprensible tanto para máquinas como para personas. Así, las API RESTFUL son más flexibles y fáciles de configurar. (RedHat, 2021).

En conclusión, al comparar estas dos arquitecturas, las API REST son ideales para los escenarios tecnológicos más modernos, como el internet de las cosas (IoT), el desarrollo de aplicaciones móviles y la computación sin servidor, debido a su ligereza. Por otro lado, los servicios web SOAP ofrecen niveles superiores de seguridad y cumplimiento para operaciones asociadas con la dinámica operativa de las empresas; sin embargo, la rigurosidad de sus reglas puede resultar en una gestión más pesada de la información.

## Síntesis

A continuación, se presenta una síntesis de la temática estudiada en el componente formativo.



## Material complementario

Tema	Referencia	Tipo de material	Enlace del recurso
La internet y la web	ThePRBridge. (2011, 13 octubre). Cómo funciona el internet [Video]. YouTube.	Video	<a href="https://www.youtube.com/watch?v=31LE0bPLrhM">https://www.youtube.com/watch?v=31LE0bPLrhM</a>
La internet y la web	Aaron. (2009, 19 febrero). How the internet Works in 5 Minutes [Video]. YouTube.	Video	<a href="https://www.youtube.com/watch?v=7LPdttKXPc">https://www.youtube.com/watch?v=7LPdttKXPc</a>
Fundamento de la internet	Saklar. (2017, 25 junio). Topologías de Red [Video]. YouTube.	Video	<a href="https://www.youtube.com/watch?v=awLJkNHBoms">https://www.youtube.com/watch?v=awLJkNHBoms</a>
Introducción a las tecnologías para el desarrollo web	Code War. (2020, 17 mayo). 2. Frontend y backend, en qué consisten, sus tecnologías, APIs - Aprende a programar sistemas web [Video]. YouTube.	Video	<a href="https://www.youtube.com/watch?v=rQX8D4dy1PM">https://www.youtube.com/watch?v=rQX8D4dy1PM</a>

## Glosario

**Arquitectura software:** es una metodología que se refiere a la estructuración de un sistema software que, idealmente, se crea en etapas tempranas de la construcción del sistema.

**Página web:** es un documento que se puede mostrar en un navegador web como Firefox, Google Chrome, Microsoft Internet Explorer o Edge, o Safari de Apple.

**Servidor web:** es una colección o conjunto de páginas web que se agrupan y normalmente se conectan de varias maneras. Por lo regular se le conoce como "sitio web" o simplemente "sitio".

**Sitio web:** conjunto de programas y rutinas que permiten a la computadora realizar determinadas tareas.

**W3C:** por sus siglas World Wide Web Consortium. Es una comunidad internacional que genera recomendaciones y estándares que aseguran el crecimiento de la WWW a largo plazo.

**Web push:** son notificaciones que el usuario recibe en su navegador (móvil o de escritorio) sin necesidad de tener una aplicación descargada en su equipo.

## Referencias bibliográficas

Facebook. (2021). Facebook - Meld je an of registreer je.

Fisher, S. (2021, 19 de mayo). ¿Qué es TCP/IP y cómo funciona? Avast Academy.

<https://www.avast.com/es-es/c-what-is-tcp-ip>

IBM. (2021, 7 de mayo). SOA (arquitectura orientada a servicios).

<https://www.ibm.com/co-es/cloud/learn/soa>

Ingenio Virtual. (2018a, 11 de febrero). Conceptos básicos sobre tecnologías de desarrollo web. <https://www.ingeniovirtual.com/conceptos-basicos-sobre-tecnologias-de-desarrollo-web/>

Juncosa, M. (2020, 6 de agosto). El modelo TCP/IP capa a capa. Aprende de Redes.

Medium. (2021, 15 de marzo). Front end development trends that will be with us in 2021. <https://productcoalition.com/front-end-development-trends-that-will-be-with-us-in-2021-a0ad023922ea>

Mistry, J. (2021, 29 de mayo). 8 best backend frameworks to use for web development 2021. Monocubed. <https://www.monocubed.com/best-backend-frameworks/>

Mozilla. (2021, 13 de mayo). How does the internet work? - Learn web development. MDN Web Docs. [https://developer.mozilla.org/en-US/docs/Learn/Common\\_questions/How\\_does\\_the\\_internet\\_work](https://developer.mozilla.org/en-US/docs/Learn/Common_questions/How_does_the_internet_work)

Ramos, R. (2020, 18 de noviembre). ¿Qué es JavaScript y para qué sirve? Agencia de Marketing Digital. <https://soyrafaramos.com/que-es-javascript-para-que-sirve/>

RedHat. (2021). REST vs. SOAP.

<https://www.redhat.com/en/topics/integration/whats-the-difference-between-soap-rest>

Tanenbaum, A. S., Wetherall, D. J., & Elizondo, A. V. R. (2012). Redes de computadoras. Pearson Educación.

Third Rock Techkno. (2021, 24 de marzo). Top 5 picks for backend development in 2021. <https://www.thirdrocktechkno.com/blog/top-5-picks-for-backend-development-in-2021/>

## Créditos

Nombre	Cargo	Centro de Formación y Regional
Milady Tatiana Villamil Castellanos	Responsable del Ecosistema	Dirección General
Olga Constanza Bermúdez Jaimes	Responsable de Línea de Producción	Centro de Servicios de Salud - Regional Antioquia
Santiago Muñoz de la Rosa	Experto Temático	Centro de Teleinformática y Producción Industrial - Regional Cauca
Paola Alexandra Moya Peralta	Evaluadora Instruccional	Centro de Servicios de Salud - Regional Antioquia
Carlos Julián Ramírez Benítez	Diseñador de Contenidos Digitales	Centro de Servicios de Salud - Regional Antioquia
Luis Jesús Pérez Madariaga	Desarrollador Fullstack	Centro de Servicios de Salud - Regional Antioquia
Edgar Mauricio Cortés García	Actividad Didáctica	Centro de Servicios de Salud - Regional Antioquia
Daniela Muñoz Bedoya	Animador y Productor Multimedia	Centro de Servicios de Salud - Regional Antioquia
Luis Gabriel Urueta Álvarez	Validador de Recursos Educativos Digitales	Centro de Servicios de Salud - Regional Antioquia
Margarita Marcela Medrano Gómez	Evaluador para Contenidos Inclusivos y Accesibles	Centro de Servicios de Salud - Regional Antioquia
Daniel Ricardo Mutis Gómez	Evaluador para Contenidos Inclusivos y Accesibles	Centro de Servicios de Salud - Regional Antioquia