

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 bajo la arquitectura de la Internet. Seguidamente se revisarán las tecnologías más importantes para el desarrollo web, tanto del lado del cliente como del servidor y por último se abordan los conceptos básicos de una arquitectura orientada a servicios.

Tabla de contenido

Introducción	1
1. La Internet y la web	2
1.1. Funcionamiento de la Internet.....	2
Historia de la Internet.....	2
“Hardware” de red	3
Modelo de referencia TCP/IP	7
Dirección IP.....	11
Sistema de Nombres de Dominio, DNS	11
Protocolo HTTP.....	11
1.2. Arquitectura web	12
Arquitectura cliente/servidor	12
Cliente web.....	13
Servidor Web.....	13
2. Introducción a las tecnologías para el desarrollo web	15
2.1. Tecnología del “front-end”	15
HTML5	16
CSS3	17
JavaScript (JS)	18

“Frameworks” del “front-end”	19
2.2. Tecnologías del “back-end”	20
Lenguajes del servidos.....	20
“Frameworks” del “back-end”	21
3. Introducción a la arquitectura orientada a servicios	23
3.1. SOAP.....	24
3.2. REST.....	24
Síntesis	26
Material complementario	27
Glosario	28
Referencias bibliográficas	29
Créditos	31

Introducción

Este componente formativo se centra en los conceptos esenciales de la arquitectura de sistemas web, incluyendo tanto aspectos físicos como lógicos, tecnologías del lado del cliente y del servidor, así como la comunicación con otros sistemas. Su objetivo es proporcionar un análisis crítico para seleccionar tecnologías adecuadas según los requisitos del proyecto.

La evolución de Internet y la web en las últimas décadas ha impulsado aplicaciones más interactivas, rápidas y seguras, desde interfaces móviles hasta plataformas de contenido multimedia. Es crucial entender los fundamentos de esta compleja red para comprender su funcionamiento y desarrollo.

La arquitectura web abarca una amplia gama de tecnologías, desde el “front-end” que se ocupa de la experiencia del usuario hasta el “back-end” que gestiona la lógica del servidor. HTML5, CSS3, JavaScript y “frameworks” como React y Angular son elementos clave en la creación de contenido dinámico y en la evolución de la interacción en línea. En conjunto, estas tecnologías continúan transformando nuestra experiencia digital y la conectividad global.

1. La Internet y la web

El desarrollo de las aplicaciones y sistemas web, han sido una tendencia desde las últimas tres décadas y sus mejoras han ido evolucionando conforme a las innovaciones de la infraestructura tecnológica que las soporta. Hoy en día vemos que las aplicaciones son mucho más interactivas, más rápidas, más seguras y usables, un ejemplo de esto es cuando se ingresa a un sitio web desde un dispositivo móvil, en el cual se observa cómo la interfaz del sitio se ajusta o responde de una mejor manera a la pantalla del móvil, otro ejemplo es cuando vemos un video de YouTube, cuando conversamos por WhatsApp o vemos una serie en Netflix. Todas estas condiciones y aspectos se dan gracias a una robusta infraestructura de “hardware” que permite el despliegue no solo de aplicaciones sino de servicios que son usualmente utilizados día a día. En ese sentido, es importante comprender los primeros componentes que enriquecen y dan rigurosidad a la construcción de esta naturaleza de sistemas.

1.1. Funcionamiento de la Internet

La aparición de la Internet tuvo lugar cuando se dio la necesidad de comunicación y de interacción con distintas personas en cualquier lugar del mundo. De esta manera se ha considerado esta tecnología como una columna vertebral de la web, y se concibe como la infraestructura técnica que la hace posible. Desde lo más básico, se presenta como una gran red de computadoras que se comunican simultáneamente.

Historia de la Internet

Su historia se remonta al comienzo de la década de 1960, como un proyecto de investigación llamado ARPANET, financiado por el ejército de los Estados Unidos y más adelante se convirtió en una infraestructura pública en la década de 1980, con el apoyo

de muchas instituciones del sector público y privado. Las distintas tecnologías que soporta Internet han evolucionado con el tiempo, especialmente para satisfacer la gran demanda existente por parte de la población, no obstante, la forma en que funciona no ha cambiado mucho. En este sentido la Internet se muestra como una forma de conectividad entre todos los equipos de cómputo, y por su distribución y organización, propenderá a que se encuentre la mejor manera de mantenerse conectados.

“Hardware” de red

Cuando dos computadoras necesitan comunicarse, se pueden vincular a través de medios físicos, por lo regular el cable “Ethernet” o de forma inalámbrica con tecnologías como Wi-Fi, “Bluetooth”, “ZigBee”, entre otras.

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

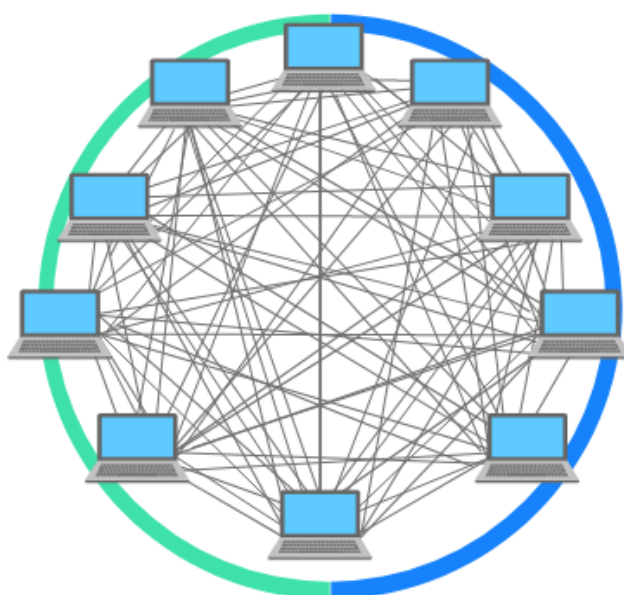
Figura 1. Ejemplo de conexión entre dos equipos o terminales



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 necesitan 45 cables y unos nueve conectores por computador, como se alcanza a apreciar en la siguiente figura.

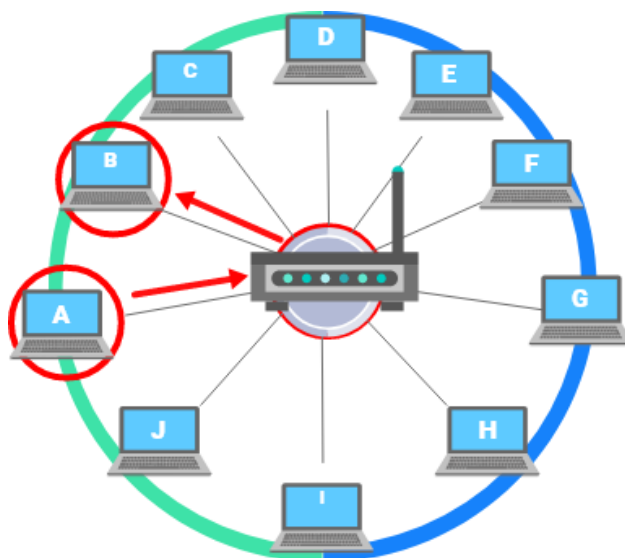
Figura 2. Ejemplo de conexión entre diez equipos o terminales



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. Ejemplo de conexión de diez equipos con un enrutador.



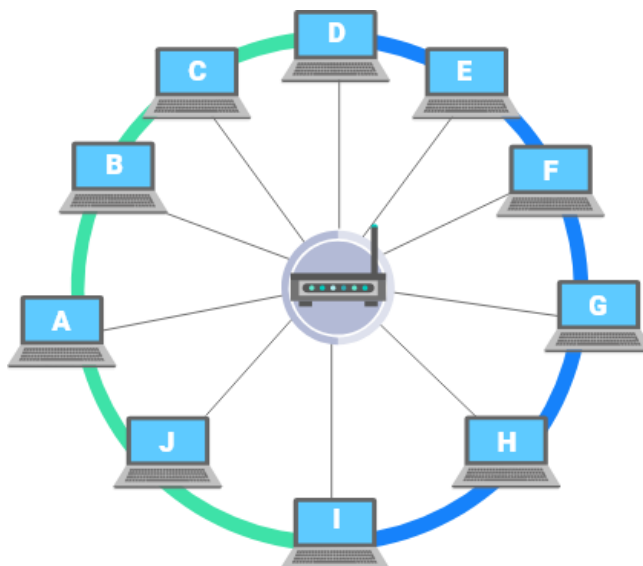
Teniendo en cuenta la gráfica anterior, ¿qué sucede al conectar cientos, miles, millones de ordenadores entre sí? Por supuesto un solo enrutador no puede escalar tanto, pero, anteriormente se ha indicado que el enrutador es como un pequeño computador, entonces ¿qué impide conectar dos enrutadores a la vez?

El proceso de conexión entre los diversos terminales demuestra que puede ser un ejercicio muy complejo y engorroso, por lo tanto, con el fin de solucionar esta organización, cada ordenador en una red está conectado a una pequeña computadora especial llamada enrutador o “router” (en inglés).

Este dispositivo cumple una función: tal como lo hace un señalizador en las estaciones del sistema integrado de transporte público, que indican cuál ruta necesitas para llegar a un destino. De esta manera el “router” se encarga de asegurar que el mensaje enviado desde un computador o terminal emisor llegue al destino correcto. 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 4. Red de diez terminales que solo requiere un enrutador con 10 conectores



En ese sentido, estas conexiones entre terminales generan redes de datos para la 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.

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 5. Ejemplo de red de área local inalámbrica y cableada



El diseño físico y lógico de la red llamado topología está basado en los enlaces de punto a punto comúnmente para redes alámbricas. El estándar IEEE 802.3, comúnmente conocido como “Ethernet”, es la manera más común de LAN alámbrica. La figura en su apartado b, muestra un ejemplo de topología de “Ethernet” conmutada. Cada computadora se comunica mediante el protocolo “Ethernet” y se conecta a una caja conocida como “switch” a través de un enlace de punto a punto.

En ese sentido, un “switch” tiene varios puertos, cada uno de los cuales se puede conectar a una computadora. La función que cumple el “switch” es transmitir paquetes entre las computadoras conectadas a él, y utiliza la dirección en cada paquete para determinar a qué computadora se lo debe enviar.

Modelo de referencia TCP/IP

Al igual que los humanos, es importante que todos los equipos de cómputo tengan un modo común de comunicarse entre ellos. Para la mayoría de los equipos actuales, este modo es TCP/IP. TCP/IP suele venir inmerso en los equipos y está automatizado en buena medida. No obstante, es necesario la comprensión del modelo

TCP/IP, sobre todo en el caso de que vaya a configurar un equipo para conectarlo a otro sistema. (Fisher, 2021)

Así, este protocolo TCP/IP, cuyas siglas definen: “Transmission Control Protocol” / Internet “Protocol” (Protocolo de control de transmisión / Protocolo de Internet), se presenta como un conjunto de reglas estandarizadas que permiten a los equipos comunicarse en una red como Internet. La potencialidad de las máquinas está en su velocidad de transmisión y comunicación con otras, de hecho, una de las tendencias de esta cuarta revolución industrial es la interacción de ellas de manera inteligente, sobre todo en contexto de producción. Muchas de las cosas para las que utilizamos los equipos (enviar mensajes de correo electrónico, ver Netflix u obtener indicaciones para llegar a un sitio) dependen de la comunicación entre ellos. Pueden ser equipos de distintos fabricantes e incluso encontrarse en zonas geográficas diferentes. De esta manera las personas y plataformas “software” que los utilizan pueden hablar distintos lenguajes humanos e informáticos.

El blog de “Avast Academy”, señala: “...Una interacción determinada puede darse entre dos sistemas informáticos o involucrar cientos de sistemas. Sin embargo, como sucede al pasar una carta o un paquete de mano en mano, cada transacción se produce entre solo dos equipos cada vez. Para que esto suceda, los dos equipos deben saber, por adelantado, cómo se espera que se comuniquen.

- ¿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

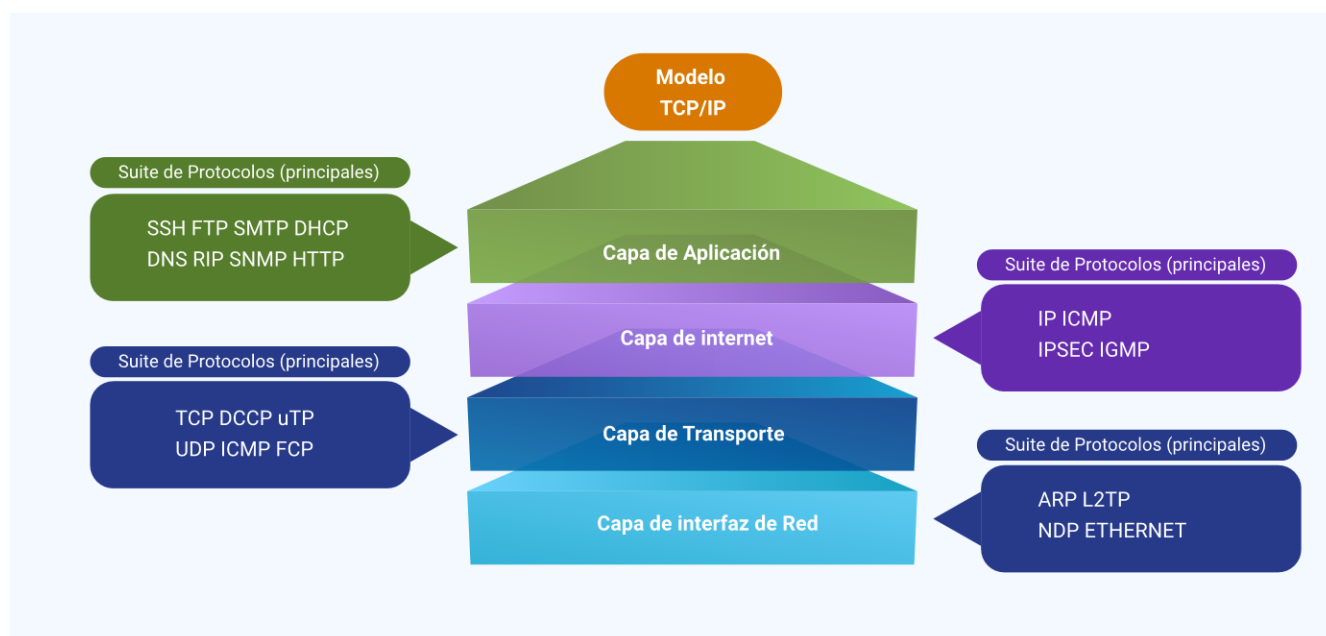
En la literatura, se encuentran un sinnúmero de definiciones alrededor de las capas de este modelo, entre las más importantes y haciendo un resumen para dar mayor claridad, se tienen:

- **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 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 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.
- **Capa de aplicaciones.** 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, las aplicaciones no tienen que preocuparse por ello.

En la siguiente gráfica, se muestran las capas y sus respectivos protocolos.

Figura 6. Principales protocolos de Modelo TCP/IP por capas



Nota. Tomado de Juncosa, 2020.

Dirección IP

Para enviar un mensaje a una computadora, se debe especificar a cuál. Es por ello por lo que toda computadora conectada a una red cuenta con una dirección única que la identifica de manera lógica y jerárquica, llamada “dirección IP” o Protocolo de Internet (IP de sus siglas en inglés Internet “Protocol”). Esta dirección se compone por una serie de cuatro números separados por puntos, por ejemplo: 192.168.2.19. Para las computadoras es un identificador simple, pero los humanos van a tener dificultad a la hora de recordar y memorizar este tipo de dirección. De esta manera, con el propósito de convertir esta serie numérica en algo que podamos asociar con mayor facilidad a la dirección IP, se utiliza lo que hoy en día conocemos como nombre de dominio.

Sistema de Nombres de Dominio, DNS

Son como una libreta de direcciones de sitios web. Cuando se escribe una dirección web en el navegador, el navegador busca los DNS antes de recuperar el sitio web. El navegador necesita averiguar en qué servidor vive el sitio web y así enviar los mensajes HTTP al lugar correcto (ver más abajo). Esto es como buscar la dirección de la tienda para que pueda llegar a ella. Por ejemplo, google.com es el nombre de dominio utilizado para sustituir la dirección IP 173.194.121.32. Así, usar un nombre de dominio es la manera más fácil para nosotros de identificar un ordenador a través de Internet.

Protocolo HTTP

El Protocolo de Transferencia de Hipertexto es un protocolo de aplicación que define un idioma para que los clientes y servidores se puedan comunicar. Esto es como el idioma que utilizas para ordenar sus compras. De esta forma, es el nombre de un protocolo el cual nos permite realizar una petición de datos y recursos, como pueden

ser documentos HTML. Es la base de cualquier intercambio de datos en la web, y un protocolo de estructura cliente-servidor, esto quiere decir que una petición de datos es iniciada por el elemento que recibirá los datos (el cliente), normalmente un navegador web.

En ese orden de ideas, se han abordado conceptos sobre los componentes que hacen parte de la arquitectura de la Internet. Como se ha expuesto, Internet es una infraestructura técnica que permite que miles de millones de computadoras estén conectadas entre sí. Algunas, también llamadas servidores web que son capaces de enviar mensajes inteligibles a los navegadores. Por tanto, Internet es una infraestructura, mientras que la web es un servicio construido sobre dicha infraestructura. Se resalta que existen otros servicios soportados por Internet, como es el correo electrónico. (Moz://a, 2021)

1.2. Arquitectura web

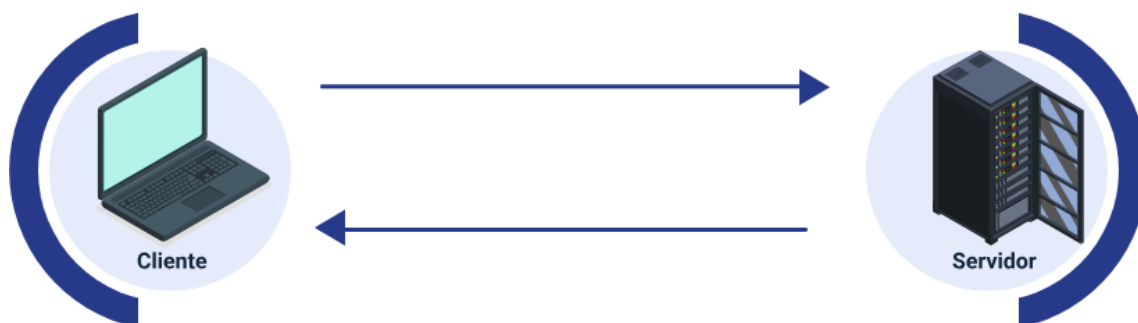
Para entender el funcionamiento de un sistema, plataforma y aplicación web, es necesario conocer los conceptos y el comportamiento de tecnologías que están inmersas en esta arquitectura. Por lo que se detalla a continuación.

Arquitectura cliente/servidor

Una aplicación web tiene una arquitectura simplificada, como lo es la arquitectura CLIENTE/SERVIDOR, en el que, por un lado, se encuentra al cliente que está constituido principalmente por un explorador web o web browser, que será el encargado tanto de mostrar como de solicitar información o documentos a través de

una red. Las computadoras conectadas a la web se llaman clientes y servidores. Un diagrama simplificado de cómo interactúan se vería de la siguiente forma:

Figura 7. Arquitectura cliente/servidor



Cliente web

Es una aplicación ejecutable que por lo regular se encuentra previamente instalada en el dispositivo del usuario, permitiendo el acceso a Internet y siendo capaces de mostrar y solicitar documentos sobre una red, en ese sentido se infiere que es el responsable de la capa de presentación.

Por tanto, los clientes web son dispositivos de los usuarios conectados a Internet (por ejemplo, tu ordenador conectado a la red Wi-Fi o el teléfono conectado a la red de telefonía móvil) y el “software” que se encuentra disponible y permite acceder a Internet en dichos dispositivos (normalmente, un navegador web como Firefox o Chrome).

Servidor Web

En el otro lado de la arquitectura web se tiene al servidor web, cuya actividad más importante es la de atender las solicitudes de los clientes a través de los navegadores web. Es en esta instancia donde se lleva a cabo todo el procesamiento de

las aplicaciones y la gestión de los datos. De manera adicional los servidores son computadores que almacenan páginas web, sitios o aplicaciones.

Cuando un dispositivo cliente quiere acceder a una página web, una copia de la página web se descarga desde el servidor en el equipo cliente y se muestra en el navegador web del usuario. Básicamente, un servidor es una gran computadora que guarda y transmite datos vía Internet.

2. Introducción a las tecnologías para el desarrollo web

Las tecnologías de desarrollo web se refieren a una multitud de lenguajes y herramientas de programación que se utilizan para producir sitios web, aplicaciones dinámicas y plataformas, con todas las funciones necesarias para dar alcance a un sinnúmero de necesidades y requerimientos por parte de los usuarios.

Este tipo de tecnologías, cada vez están en evolución, no es necesario hacer un estudio minucioso sobre el estado de transformación de estas tecnologías, solo basta con mirar el nuevo auge de diversos lenguajes de programación que permiten generar mejor y mayores acciones en sitios web. Por eso es por lo que las tecnologías web han sido consideradas como multiplataforma. Esta afirmación está asociada según el grupo de expertos de Ingenio Virtual donde indican:

“Los modelos y tecnologías de desarrollo web han evolucionado mucho en la última década, existen multitud de aplicaciones, “frameworks”, librerías, arquitecturas y sistemas de publicación en diferentes versiones que a su vez reciben cambios o mejoran con el tiempo. El progreso también ha tenido lugar en lo relacionado con la administración de sistemas, servicios de alojamiento, técnicas de escalabilidad, monitorización y gestión de centros de procesos de datos. Esta evolución ha dado lugar a la convergencia de una gran cantidad de tecnologías, herramientas y estilos arquitectónicos para desarrollar sitios web y aplicaciones.” (I, 2018)

2.1. Tecnología del “front-end”

Las tecnologías de “front-end” son usadas para el "lado del cliente", sea de un sitio web o aplicación. Se utilizan para desarrollar los componentes interactivos y producir los elementos que los usuarios ven e interactúan con ellos. Esto incluye

colores y estilos de texto, imágenes, botones y menús de navegación, entre otros objetos interactivos. Normalmente estas tecnologías y lenguajes vienen implementados en los diferentes navegadores.

HTML5

HTML es la sigla de “Hyper Text Markup Language”. Es una de las tecnologías fundamentales necesarias para el desarrollo de todo tipo de aplicaciones web. Proporciona la estructura base para una página web. El código HTML garantiza que todo el contenido de un sitio web tenga el formato adecuado y estructurado. Esto es para que cada navegador de Internet pueda mostrar el contenido según lo previsto. Sin HTML, un navegador no podría mostrar texto o cargar imágenes y otros elementos.

HTML5, la versión más actual de HTML, contiene un conjunto más amplio de tecnologías que permite a los sitios web y a las aplicaciones ser más diversas y de gran alcance. A este conjunto se le llama HTML5. En ese sentido, esta tecnología presenta una gran cantidad de recursos, como librerías, “plugins” y API que se puedan usar con un lenguaje de programación como lo es JavaScript para realizar un sitio web más interactivo y dinámico. Las tecnologías más usadas por HTML5 son las siguientes:

- **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.

- **WebWorkers.** 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.
- **WebSockets.** 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 (del inglés “Cascading Style Sheets”) o CSS es el lenguaje de estilos utilizado para describir la presentación de documentos HTML o XML (en-US) CSS, describe como debe ser renderizado el elemento estructurado, es decir, se encarga de la apariencia del sitio. De manera adicional, abordamos la definición descomponiendo sus siglas:

- “Cascading”: significa que los estilos que aplicamos a los elementos de una página web se propagan a los elementos que contiene, se aplican en cascada.
- “Style”: mediante la tecnología CSS se aplican estilos visuales a los distintos elementos de nuestra página web.
- “Sheets”: lo que traduce hojas, porque los estilos de una página web se añaden en archivos aparte. Estos archivos deben tener la extensión CSS de manera general.

Desde CSS3, el alcance de las especificaciones se incrementó de forma significativa y el progreso de los diferentes módulos de CSS comenzó a mostrar varias diferencias. De esta forma, con CSS3 se tiene mayor control de los elementos del sitio, con sus nuevas funciones y atributos sobre las etiquetas HTML, maximizan las acciones visuales del sitio o plataforma.

JavaScript (JS)

En la literatura hay un sinnúmero de definiciones con respecto a este lenguaje de programación, no obstante, hay una especial la cual se cita de parte del blog de Rafa Ramos de manera literal, él dice:

“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)

De manera adicional, JS si bien es más conocido como un lenguaje de “scripting” (secuencias de comandos) para páginas web, es usado también en muchos entornos fuera del navegador, tal como Node.js, Apache CouchDB y Adobe Acrobat.

Como conclusión a estas 3 principales tecnologías, se realiza una analogía con el cuerpo humano, de esta manera los huesos que son la estructura ósea es lo que a su vez hace el lenguaje de etiquetas HTML, los músculos que hacen que el cuerpo se mueva y tenga motricidad y dinámica es lo que a su vez hace JavaScript, y la piel que es la cubierta del cuerpo humano y se puede decir que es la que da apariencia, es lo que

hace a su vez el CSS. Con esta analogía se espera que se tenga más comprensión de este tipo de tecnologías del lado del cliente.

“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 listan los “frameworks” más importantes a la fecha, con el fin que se tenga un reconocimiento para usos futuros. Este listado es apoyado según la comunidad más grande de gestión de productos llamada “Product Coalition” (P, 2021):

- **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 Vue. Entre las empresas que lo eligieron como herramienta de desarrollo, se encuentran Stackoverflow y PlayStation, según lo define la comunidad de Product Coalition. (P, 2021)
- **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 “Framework” dice: 'Concéntrese en crear sitios impresionantes en lugar de hacer que su código funcione'. (Angular, 2021).

2.2. Tecnologías del “back-end”

Este tipo de tecnologías son aquellas que se utilizan en el lado Servidor. En ese sentido el Servidor las utiliza para gestionar y/o procesar las diferentes peticiones de información que le llegan, además realiza gestión sobre las bases de datos alojadas en los mismos. La información una vez tratada se envía devuelta al dispositivo para que sea visualizada en el dispositivo a través de las tecnologías “front-end”. Las tecnologías más conocidas y usadas en el “back-end” son PHP, Java, .NET, Python, MySQL, etc.

Lenguajes del servidores

En el lado del servidor, el “back-end” potencia el funcionamiento del sitio web debido a que es en esta instancia donde se realizan los procesos más robustos de información, por lo que se necesitan de igual manera lenguajes de programación con una diversidad de funciones para el tratamiento de esos datos. Además, los programas escritos aquí por los desarrolladores de “back-end” se utilizan para comunicar la información de la base de datos al navegador. La empresa de base tecnológica Third Rock Techkno indica que: “En 2021, el desarrollo de “back-end” será aún más vital a medida que las empresas busquen expandirse a un ritmo rápido. Además, esta empresa muestra los lenguajes de programación del servidor que están en tendencia”. (Third Rock Techkno, 2021)

- **Node JS.** Este lenguaje tiene su propio entorno de ejecución. Esta es la razón por la que es popular entre la comunidad de desarrolladores. Además, como Node.js en la literatura de esta disciplina del desarrollo “software” es comúnmente visto como uno de los lenguajes más eficientes, se prefiere

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, ya que cuenta con el soporte oficial de Google. Kotlin se muestra de cara como uno de los lenguajes que tiene una curva de aprendizaje muy fácil según Third Rock Techkno. Por lo tanto, el lenguaje es menos propenso a fallos y brinda a los desarrolladores la oportunidad de encontrar los errores fácilmente. (Third Rock Techkno, 2021).
- **Ruby.** Se estima que este lenguaje actualmente jugará un papel fundamental, ya que ayuda a reducir el tiempo dedicado a tareas estandarizadas.
- **PHP.** Es un lenguaje de programación de “back-end” conveniente que es muy flexible y tiene características impecables. Es uno de los programas donde su comunidad se ha venido actualizando y hay gran mayoría de “frameworks” basados en este lenguaje. Es por eso por lo que PHP se considera el futuro del desarrollo “back-end” en 2021. Además, la sintaxis de PHP es muy expresiva, lo que brinda libertad creativa a los desarrolladores.

“Frameworks” del “back-end”

Como se mencionó en el apartado de “frameworks” del lado del “front-end”, este tipo de recurso ayuda a los desarrolladores a realizar sus trabajos de manera más ágil, segura y robusta. En ese sentido se presentan los “frameworks” del lado del “back-end” que son tendencia. Según indica una de las empresas de desarrollo web más importantes de Canadá Monocubed. (Mistry, 2021)

- **Laravel.** Este “framework” tiene una excelente estructura del lenguaje, la capacidad de adaptarse a grupos gigantes y la efectividad de sus recursos como es la llamada “toolbox”. Basado en el lenguaje PHP, sigue una sintaxis elegante y expresiva. Laravel permite a los desarrolladores web ser flexibles y creativos mientras se encargan de los detalles de fondo a diferencia de otros “frameworks”.
- **Django.** Es un “framework” de alto nivel que se desarrolla con la noción de 'baterías incluidas'. Esto significa que casi todo lo que necesitaría cualquier desarrollador ya está incluido. Django es utilizado principalmente para el desarrollo de aplicaciones web interactivas a gran escala impulsadas por bases de datos. Es altamente personalizable y escalable. Tiene una amplia comunidad y una importante documentación.
- **Ruby on Rails.** Está escrito en el lenguaje de programación Ruby. Es fácil de probar y compilar, junto con el amplio apoyo de la comunidad y una gran cantidad de recursos disponibles, lo que lo convierte en una alternativa viable para los principiantes.

3. Introducción a la arquitectura orientada a servicios

Hasta el momento se han venido revisando las tecnologías para el desarrollo web, por lo que es necesario entender que aparte de crear aplicaciones también se pueden crear servicios. Para esto es necesario abordar de manera sucinta el concepto de la Arquitectura Orientada a Servicios llamada SOA. Al respecto es un tipo de arquitectura de “software”, la cual se basa en la integración de aplicaciones mediante servicios. Según IBM, se define SOA, como:

“...una manera de hacer que los componentes de “software” sean reutilizables a través de interfaces de servicio. Estas interfaces utilizan estándares de comunicación comunes entre sí, de tal manera que pueden incorporarse rápidamente a nuevas aplicaciones sin tener que realizar una integración profunda cada vez...” (SOA, 2021)

En ese sentido, es interesante el desarrollo de “software” en sí, la reutilización de componentes, y la comunicación con distintas aplicaciones sin tener la necesidad de crear un desarrollo a la medida para cada aplicación al cual se quiera conectar. Es por eso por lo que nace el concepto de servicio y de interoperabilidad.

Como una de las implementaciones de SOA existen las arquitecturas REST y SOAP, las dos siguen los mismos “principios” de SOA. Por tanto, REST y SOAP son dos vías distintas para la transmisión de datos en línea. Específicamente, ambas definen cómo diseñar interfaces de programación de aplicaciones (API), que no es más que una tecnología que sirve como un medio para que dos aplicaciones o plataformas se puedan comunicar entre sí y lo hacen exponiendo o publicando un conjunto de servicios y reglas para cómo acceder a ellos. La principal diferencia es que SOAP es un protocolo, y REST no lo es. Por lo general, alguno de ellos registrará a las API, es decir,

definirá una serie de reglas para la publicación de los servicios como también del acceso, según el caso práctico y las preferencias del desarrollador.

3.1. SOAP

Por sus siglas, protocolo simple de acceso a objetos. Se define como un protocolo o reglas estándar que se creó principalmente para permitir la comunicación entre las aplicaciones que estaban desarrolladas con diferentes lenguajes y además que reposaban en distintas plataformas tecnológicas. Por tener la naturaleza de un protocolo, existen reglas integradas que aumentan la complejidad y la sobrecarga en los servidores, lo cual causa que pueda haber retrasos en el tiempo que tardan las páginas en cargarse o dar alguna respuesta. No obstante, estos estándares por su estructura y seguridad son más demandados en el sector empresarial. (RedHat, 2021)

Adicionalmente, el envío de una solicitud de datos a una API de SOAP se puede administrar a través de cualquiera de los protocolos de la capa de la aplicación, por ejemplo, HTTP que es usado por los exploradores web, SMTP que es protocolo usado por los correos electrónicos, TCP, entre otros. No obstante, una vez que se recibe una solicitud, los mensajes SOAP de retorno devuelve por defecto documentos XML, que es un lenguaje de etiquetas comprensible tanto para las máquinas como para las personas.

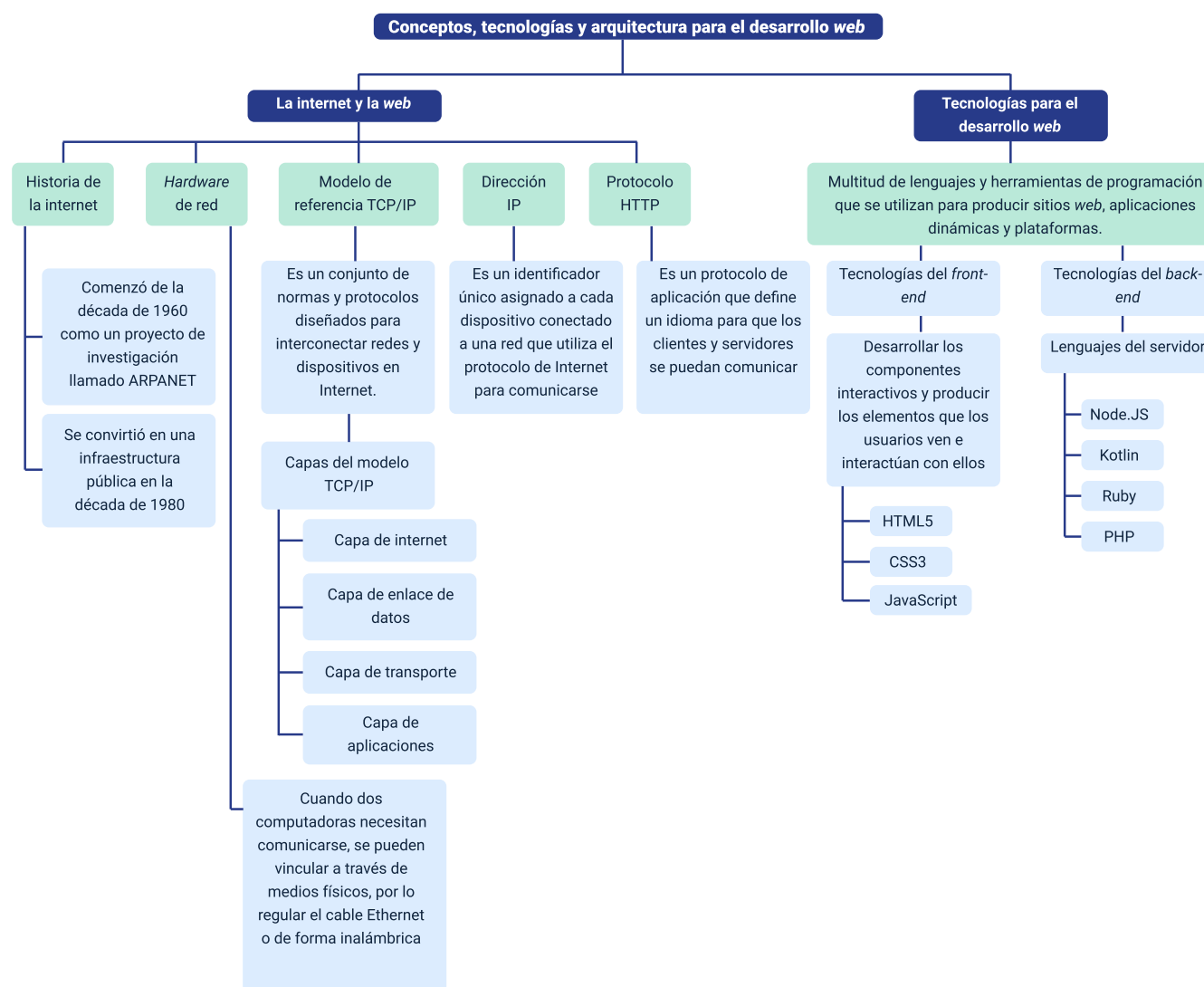
3.2. REST

Como conclusión se tiene una comparación de estas dos arquitecturas: las API de REST son livianas, así que son ideales para los escenarios tecnológicos más nuevos, como lo son el Internet de las cosas (IoT), el desarrollo de aplicaciones móviles y la informática sin servidor. Los servicios de SOAP ofrecen seguridad y cumplimiento de las

operaciones que están asociadas a la dinámica operativa de una empresa, sin embargo, al tener la rigurosidad de abarcar tantas reglas puede ser también muy pesada la gestión 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
Funcionamiento de la Internet	ThePRBridge. (2011). Cómo funciona el Internet.	Video	https://www.youtube.com/watch?v=31LE0bPLrhM
Funcionamiento de la Internet	Aaron. (2009). How the Internet Works in 5 Minutes (video). YouTube.	Video	https://www.youtube.com/watch?v=7 LPdttKXPc
Arquitectura web	saklar. (2017). Topologías de Red (video). YouTube.	Video	https://www.youtube.com/watch?v=awLJkNHBoms
Introducción a las tecnologías para el desarrollo web	Code War. (2020). 2. Frontend y backend, en que consisten, sus tecnologías, APIs - Aprende a programar sistemas web (video). YouTube.	Video	https://www.youtube.com/watch?v=rQX8D4dy1PM

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: una computadora de grandes especificaciones “hardware” que aloja un sitio web o plataformas con todos sus recursos en la Internet.

Sitio 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".

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

ISECOM. (2021). OSSTMM. <https://www.isecom.org/research.html#content5-9d>

Angular. (2021). Angular. <https://angular.io/>

Fisher, S. (2021, 19 mayo). ¿Qué es TCP/IP y cómo funciona? Avast Academy. <https://www.avast.com/es-es/c-what-is-tcp-ip>

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

Juncosa, M. (2020, 6 agosto). El modelo TCP/IP capa a capa. aprendederedes.com. <https://aprenderedes.com/2019/06/el-modelo-tcpip/>

Mistry, J. (2021, 29 mayo). 8 Best Back-end Frameworks to Use for web Development 2021. Monocubed. <https://www.monocubed.com/best-backend-frameworks/>

Moz://a. (2021, 13 mayo). How does the Internet work? - Learn web development | MDN. MDN web Docs. https://developer.mozilla.org/en-US/docs/Learn/Common_questions/How_does_the_Internet_work

P. (2021, 15 marzo). Front End Development Trends That Will Be with Us in 2021. Medium. <https://productcoalition.com/front-end-development-trends-that-will-be-with-us-in-2021-a0ad023922ea>

RedHat. (2021). REST vs. SOAP. <https://www.redhat.com/en/topics/integration/whats-the-difference-between-soap-rest>

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

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

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

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

Créditos

Nombre	Cargo	Regional y Centro de Formación
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
Paula Andrea Taborda Ortiz	Diseñador Instruccional	Centro de Diseño y Metrología - Regional Distrito Capital
Vilma Lucia Perilla Méndez	Revisora Metodológica y Pedagógica	Centro de Gestión Industrial - Regional Distrito Capital
Rafael Neftalí Lizcano Reyes	Asesor Pedagógico	Centro Industrial del Diseño y la Manufactura - Regional Santander
Jhana Johanna Bustillo Ardial	Revisión de Estilo	Centro Industrial del Diseño y la Manufactura - Regional Santander
Andrés Felipe Herrera Roldan	Diseñador de Contenidos Digitales	Centro de Servicios de Salud - Regional Antioquia
Jhon Jairo Urueta Alvarez	Desarrollador Fullstack	Centro de Servicios de Salud - Regional Antioquia
Edgar Mauricio Cortes Garcia	Actividad Didáctica	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

Nombre	Cargo	Regional y Centro de Formación
Luis Gabriel Urueta Álvarez	Validador de Recursos Educativos Digitales	Centro de Servicios de Salud - Regional Antioquia
Jaime Hernán Tejada Llano	Validador de Recursos Educativos Digitales	Centro de Servicios de Salud - Regional Antioquia