**FORMATO PARA EL DESARROLLO DE COMPONENTE FORMATIVO**

|  |  |
| --- | --- |
| PROGRAMA DE FORMACIÓN | Programación de aplicaciones y servicios para la nube |

|  |  |  |  |
| --- | --- | --- | --- |
| COMPETENCIA | 220501123 – Construir sitios web según técnicas de interoperabilidad y protocolos técnicos. | RESULTADOS DE APRENDIZAJE | 220501123-1 Seleccionar las tecnologías para la construcción de aplicaciones web de acuerdo a los requisitos técnicos. |

|  |  |
| --- | --- |
| NÚMERO DEL COMPONENTE FORMATIVO | 001 |
| NOMBRE DEL COMPONENTE FORMATIVO | 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. |
| PALABRAS CLAVE | Web, API, internet, Arquitectura web, Servidor web |

|  |  |
| --- | --- |
| ÁREA OCUPACIONAL | 2 - CIENCIAS NATURALES, APLICADAS Y RELACIONADAS |
| IDIOMA | Español |

1. **TABLA DE CONTENIDOS:**

**Introducción**

1. **La internet y la web**
   1. Funcionamiento de la internet
      1. Historia de la internet
      2. *Hardware* de red
      3. Modelo de referencia TCP/IP
      4. Dirección IP
      5. Sistema de Nombres de Dominio, DNS
      6. Protocolo HTTP
   2. Arquitectura web
      1. ~~Arquitectura cliente/servidor~~
      2. ~~Cliente web~~
      3. ~~Servidor web~~
2. **Introducción a las tecnologías para el desarrollo web**
   1. Tecnologías del *front-end*
      1. HTML5
      2. CSS3
      3. JavaScript **(JS)**
      4. *Frameworks del front-end*
   2. Tecnologías del *back-end*
      1. Lenguajes del servidor
      2. *Frameworks del back-end*
3. **Introducción a la arquitectura orientada a servicios**
   1. SOAP
   2. REST
4. **DESARROLLO DE CONTENIDOS:**

**Introducción**

En este componente formativo se estudiarán los conceptos básicos sobre la arquitectura de un sistema web, desde la naturaleza de su comportamiento, comprendiendo los componentes físicos y lógicos para su funcionamiento, pasando por la identificación y caracterización de tecnologías web del lado del cliente y del servidor, hasta la identificación de tecnologías que permiten la comunicación y el traspaso de información o generación de servicios con otros sistemas.

En ese sentido, este componente busca generar un análisis crítico para la selección de una determinada tecnología, dependiendo de los requerimientos y alcance de un proyecto estimado, a partir de la fundamentación y rasgos distintivos que en cada apartado de este documento se detallan.

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. **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.

* + 1. **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.

* + 1. ***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 Wifi, Bluetooth, ZigBee, entre otras).

Imagen que contiene Gráfico

Descripción generada automáticamente

* + 1. **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 comunicar 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?”.(Fisher, 2021)

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 importantes y haciendo un resumen para dar mayor claridad, se tienen:

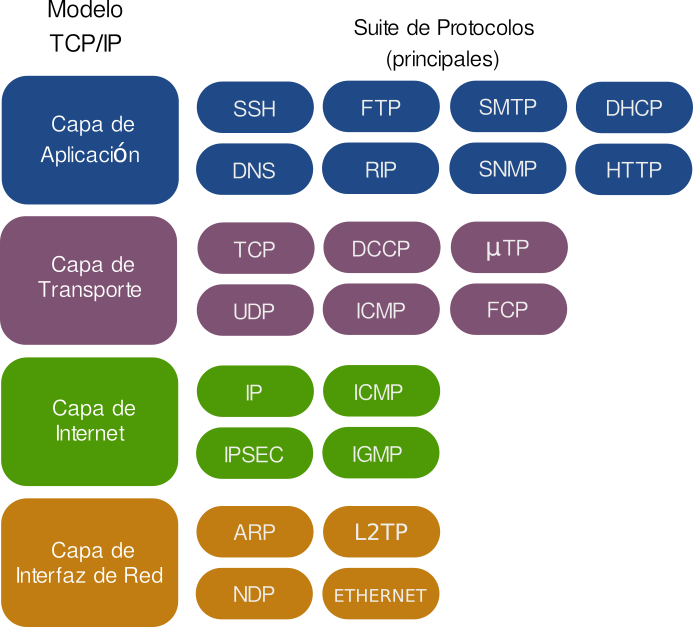
Imagen que contiene Gráfico de embudo

Descripción generada automáticamente

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

**Figura 1**

*Principales protocolos del Modelo TCP/IP por capas*



Nota. Tomado de Juncosa, 2020.

<https://www.vhv.rs/dpng/d/606-6061147_suite-de-protocolos-tcpip-protocolos-del-modelo-tcp.png>

* + 1. **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.**

* + 1. **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.

* + 1. **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. **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, como se detalla a continuación.

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

1. **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 sin número 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)

* 1. **Tecnologías 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.

* + 1. **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.
  + 1. **CSS3**

Hojas de Estilo en Cascada (del inglés *Cascading Style Sheets*) o CSS, es el lenguaje de [estilos](https://wiki.developer.mozilla.org/es/docs/Web/API/StyleSheet) utilizado para describir la presentación de documentos [HTML](https://developer.mozilla.org/es/docs/Web/HTML) o XML (en-US). CSS describe cómo 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.

* + 1. **JavaScript (JS)**

En la literatura hay un sin número 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](https://en.wikipedia.org/wiki/JavaScript#Uses_outside_Web_pages), tal como [Node.js](https://developer.mozilla.org/es/docs/Glossary/Node.js), [Apache CouchDB](https://couchdb.apache.org/) y [Adobe Acrobat](https://www.adobe.com/devnet/acrobat/javascript.html).

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.

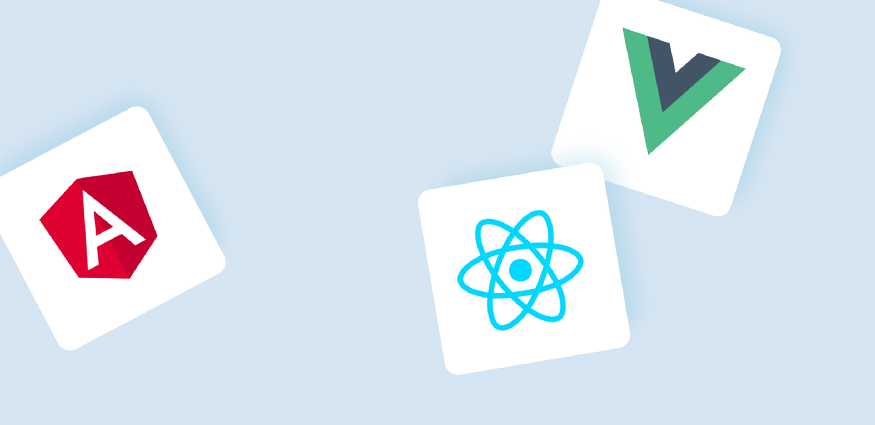
* + 1. ***Frameworks del front-end***

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).*

~~Figura 10. Logos de las principales tecnologías del Front-end~~

~~~~

~~Fuente: Tomado de (P, 2021)~~

* 1. **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-endson PHP, Java, .NET, Python, MySQL, etc.

* + 1. **Lenguajes del servidor**

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-endserá 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, debido a 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, porque 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.
  + 1. ***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, 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.

1. **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)](https://www.redhat.com/es/topics/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 regirá 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.

* 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.

* 1. **REST**

Por sus siglas en inglés, *Representational State Transfer,* traduce transferencia de estado representacional. Se define como un compendio de lineamientos que se ajustan a las necesidades de los [servicios web](https://www.redhat.com/es/topics/cloud-computing/what-are-cloud-services) y especialmente a las [aplicaciones móviles](https://www.redhat.com/es/topics/mobile) por la velocidad de transmisión y por lo ligero de la información. Cuando se envía una solicitud de datos a una API de REST, se hace a través de un protocolo de transferencia de hipertexto, comúnmente denominado HTTP. Una vez que reciben la solicitud, las API diseñadas para REST (conocidas como API o servicios web de RESTful) dan como respuesta mensajes en diversos formatos como pueden ser HTML, XML, JSON, entre otros. El formato más usado para los mensajes es la notación de objetos JavaScript (JSON). Este formato puede leerlo cualquier lenguaje de programación, es liviano y es comprensible para las máquinas como para las personas. De esta forma, las API de RESTful son más flexibles y se pueden configurar con mayor facilidad. (RedHat, 2021).

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)](https://www.redhat.com/es/topics/internet-of-things), el desarrollo de aplicaciones móviles y la [informática sin servidor](https://www.redhat.com/es/topics/cloud-native-apps/what-is-serverless). Los servicios web 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.

1. **ACTIVIDADES DIDÁCTICAS (OPCIONALES SI SON SUGERIDAS)**

|  |  |
| --- | --- |
| DESCRIPCIÓN DE ACTIVIDAD DIDÁCTICA | |
| Nombre de la Actividad | Tecnologías del desarrollo web |
| Objetivo de la actividad | Afianzar algunos de los conceptos más importantes asociados a la identificación de tecnologías del desarrollo web. |
| Tipo de actividad sugerida | Arrastrar y soltar |
| Archivo de la actividad  (Anexo donde se describe la actividad propuesta) | Actividad didactica CF1.docx |

1. **MATERIAL COMPLEMENTARIO:**

|  |  |  |  |
| --- | --- | --- | --- |
| Tema | Referencia APA del Material | Tipo de material  (Video, capítulo de libro, artículo, otro) | Enlace del Recurso o  Archivo del documento o material |
| Cómo funciona la internet | ThePRBridge. (2011). *Cómo funciona el internet* (video). YouTube. | Video | <https://www.youtube.com/watch?v=31LE0bPLrhM> |
| La internet en 5 minutos | Aaron. (2009). How the internet Works in 5 Minutes (video). YouTube. | Video | <https://www.youtube.com/watch?v=7_LPdttKXPc> |
| Topología de redes | saklar. (2017). Topologías de Red (video). YouTube. | Video | <https://www.youtube.com/watch?v=awLJkNHBoms> |
| Front-end y backend, en qué consisten, sus tecnologías, API | Code War. (2020). 2. Front-end y backend, en qué consisten, sus tecnologías, APIs - Aprende a programar sistemas web (video). YouTube. | Video | <https://www.youtube.com/watch?v=rQX8D4dy1PM> |

1. **GLOSARIO:**

|  |  |
| --- | --- |
| TÉRMINO | SIGNIFICADO |
| 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. |

1. **REFERENCIAS BIBLIOGRÁFICAS:**

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

~~Facebook - Meld je an of registered je. (2021). Facebook.~~ [~~https://www.facebook.com/unsupportedbrowser~~](https://www.facebook.com/unsupportedbrowser)

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://aprendederedes.com/redes/introduccion/modelo-tcp-ip/~~](https://aprendederedes.com/redes/introduccion/modelo-tcp-ip/)

Mistry, J. (2021, 29 mayo). 8 Best Back-endFrameworks 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>

Ramos, R. (2020, 18 noviembre). ¿Qué es JavaScript y para qué sirve? Agencia de Marketing Digital | Rafa Ramos. <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>

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-enddevelopment in 2021. <https://www.thirdrocktechkno.com/blog/top-5-picks-for-backend-development-in-2021/>

1. **CONTROL DEL DOCUMENTO**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Nombre | Cargo | Dependencia  *(Para el SENA indicar Regional y Centro de Formación)* | Fecha |
| Autor | Santiago Muñoz de la Rosa | Experto Temático | Regional Cauca, Centro de Teleinformática y Producción Industrial | Junio de 2021 |
| Paula Andrea Taborda Ortiz | Diseñadora instruccional | Regional Distrito Capital - Centro de diseño y metrología. | Junio de 2021 |
| Vilma Lucia Perilla Méndez | Revisora metodológica y pedagógica | Regional Distrito Capital – Centro de Gestión Industrial | Junio de 2021 |
| Rafael Neftalí Lizcano Reyes | Asesor pedagógico | Regional Santander - Centro Industrial del Diseño y la Manufactura | Junio de 2021 |

1. **CONTROL DE CAMBIOS**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Nombre | Cargo | Dependencia | Fecha | Razón del Cambio |
| Autor (es) | Jhana Johanna Bustillo Ardial | Revisión de estilo | Regional Santander - Centro Industrial del Diseño y la Manufactura | Junio 2021 |  |