|  |  |  |
| --- | --- | --- |
| APLICACIONES WEB  ASIGNACIÓN 01 – Tecnologías para la creación de aplicaciones Web | | |
| **Objetivo:** | Que el alumno comprenda e identifique el funcionamiento de las tecnologías utilizadas para la creación de aplicaciones Web. | |
| **Tiempo:** | **Duración Total:** | 180 min. |
| **Lectura:** | 60 min. |
|  |  |
| **Realización del Ejercicio de clase:** | 90 min. |
| **Presentación de Resultados y envío:** | 30 min. |
| **Conformación de Equipos:** | | Por parejas |
| **INSTRUCCIONES** | | |
| Por parejas:  Desarrolle un Mapa Mental que tome como elemento central las Aplicaciones Web y abarque los contenidos relacionados.   * Guarda el documento con el siguiente formato A1\_ApellidosNombre.docx * Sigue los pasos que se te indican en el apartado Ejercicio Clase.   **TEORÍA**  **Internet y Web**  Es importante señalar la distinción entre estas dos palabras que se usan cada vez más de manera indistinta.   * **Internet**   Vallejos (2017) describe que el Internet “Es una colección de miles de redes de computadora, también conocida como la súper autopista de la información”. (pp. 2)  Al ser una red de computadoras conectadas que abarca todo el mundo, ninguna empresa es su dueña; es producto de un esfuerzo cooperativo regido por un sistema de normas y reglas, cuyo propósito principal es compartir información.    Ilustración 1. La web   * **La Web**   Por otra parte,Robbins (2018), establece que la Web **(***World Wide Web, o* **www***)* es solo una de las formas en que la información puede compartirse a través de Internet. Es la única en la que permite que los documentos se vinculen entre sí mediante enlaces de hipertexto, formando así una enorme "red" de información conectada. (pp. 21)  De esta forma podemos relacionar ambos conceptos, en donde la **Web***,* es un conjunto de documentos interconectados por enlaces de hipertexto, disponibles en **Internet** que se pueden comunicar a través de la tecnología digital. Entendiéndose por *“hipertexto”* la mezcla de textos, gráficos y archivos de todo tipo en un mismo documento. (Latorre, 2020)  En palabras simples, la **Web** es un subconjunto de **Internet** que contiene información a la que se puede acceder usando un navegador.  Latorre (2020), clasifica la Evolución de la Web de la siguiente forma:   * + 1.0 (1990) Sólo se podía consumir contenido, se trataban de páginas web en donde solo se podía acceder a la información, pero sin posibilidad de interactuar (unidireccional).   + 2.0 (2004) Permitió la participación de los usuarios, con el desarrollo de foros, blogs, comentarios y después las redes sociales, muy utilizada hoy en día.   + 3.0 (2010) Web semántica, se refiere al uso del lenguaje en la red (búsqueda de contenidos usando palabras clave).   + 4.0 (2016) Ofrece comportamiento inteligente y más predictivo, basta con hacer una afirmación o una llamada para obtener como resultado lo que buscamos.   **Aplicación Web**  Web App, o “Aplicación Web”, es una versión mejorada de la página web, es una interpretación optimizada y ajustable a cualquier dispositivo móvil o tableta e independiente de los navegadores y sistemas operativos. No requiere de instalación, debido a que se carga en el servidor web y se ejecuta en el navegador. (InternetPasoaPaso.com, 2019)  Así mismo, Gómez y Cervantes (2017), definen a las aplicaciones web como un “Conjunto de páginas que funcionan en internet que el usuario ve a través de un navegador de internet.”  Existen diferencias sustanciales entre las páginas web y las aplicaciones web:   * “- Las páginas o sitios web son estáticos, lo que quiere decir que no se actualizan a diario. En cambio, estas apps son mucho más dinámicas y están cambiando constantemente. * - La aplicación web depende de la interacción del usuario para lograr su objetivo, mientras que la página web solo proporciona información al mismo y no se relaciona con él concretamente. * - La página web se construyen utilizando HTML, CSS, y tal vez un poco de JavaScript. La Web App también pero a esto le añade el uso de lenguajes de programación como PHP, Ruby o Python, y marcos como Rails, Django y Scriptcase. * - Los sitios web son más baratos y estiman menos tiempo para desarrollarlos. Al contrario que sus contrarias, que generalmente cuesta más y necesita más tiempo para su desarrollo, sobre todo por el requerimiento de lenguajes de programación o bases de datos.” (InternetPasoaPaso.com, 2019)   **Desarrollo Web**  El desarrollo Web se divide en el desarrollo de dos partes complementarias: el frontend y el backend.  **Frontend**  El Frontend,“Es la parte visible, la que muestra el diseño, los contenidos y permite a los visitantes navegar por las diferentes páginas.” (NeoAttack, 2019)  Esta parte que engloba y muestra todo el trabajo de diseño web y, por lo general, hace uso de tres lenguajes diferentes: HTML, CSS y JavaScript, los cuales se describen brevemente a continuación:  **HTML (Hypertext Markup Languaje).** Lenguaje de marcas de hipertexto, es el lenguaje utilizado para crear documentos de páginas web. Hoy en día se usan algunas versiones de HTML: HTML 4.01 es el más firmemente establecido y el HTML5 más nuevo y robusto está ganando fuerza y ​​soporte para el navegador.  HTML no es un lenguaje de programación; es un lenguaje de marcado, lo que significa que es un sistema para identificar y describir los diversos componentes de un documento, como encabezados, párrafos y listas. El marcado indica la estructura subyacente del documento (puede considerarse como un esquema detallado y legible por máquina). Es muy sencillo de aprender y utilizar. (Robbins, 2018, pp. 12)    Agregar etiquetas descriptivas a un documento de texto se conoce como "marcar" el documento. HTML define docenas de elementos de texto que componen las páginas web, tales como encabezados, párrafos, texto enfatizado y, por supuesto, enlaces. También hay elementos que agregan información sobre el documento (como su título), medios como imágenes y videos, y widgets para entradas de formularios, solo por nombrar algunos.  (Ver Ilustraciones 2 y 3)    Ilustración 2. ejemplo de código html    Ilustración 3. imágenes invocada en el html  **CSS (Cascade Style Sheets).** Mientrasel HTML es usado para describir el contenido en una página web, son las Hojas de Estilo en Cascada (CSS) las que describen cómo debería verse ese contenido. En el negocio del diseño web, el aspecto de la página se conoce como su presentación. Eso significa fuentes, colores, imágenes de fondo, interlineado, diseño de página, etc. todo controlado con CSS. Con la versión más reciente (CSS3), se pueden agregar efectos especiales y animación básica a las páginas web. (Robbins, 2018, pp. 12)  CSS también proporciona métodos para controlar cómo se presentarán los documentos en contextos distintos del navegador tradicional, como en la impresión y / o en dispositivos con ancho de pantallas pequeñas.  Las hojas de estilo también son una gran herramienta para automatizar la producción porque puede cambiar la apariencia de un elemento en todas las páginas de su sitio editando un solo documento de hoja de estilo. Las hojas de estilo son compatibles hasta cierto punto por todos los navegadores modernos. (Ver ilustraciones 4 y 5)    Ilustración 4. ejemplo de hoja de estilo    Ilustración 5. visualización de la página aplicando el estilo indicado  **JavaScript.** JavaScript es un lenguaje de secuencias de comandos que se utiliza para agregar interactividad y comportamientos a las páginas web, incluidos estos (solo por nombrar algunos):  - Verificar entradas de formularios para entradas válidas;  - Intercambiar estilos para un elemento o un sitio completo;  - Hacer que el navegador recuerda la información sobre el usuario para la próxima vez que lo visita; y  - creación de widgets de interfaz, tales como expandir menús. (Robbins, 2018, pp. 13)  JavaScript se utiliza para manipular los elementos en la página web, los estilos aplicados a ellos, o incluso el navegador en sí. Hay otros lenguajes de secuencias de comandos web, pero JavaScript (también llamado ECMAScript) es el estándar y el más usado.  También es común escuchar el término scripting DOM utilizado en relación con JavaScript. DOM significa Modelo de Objetos del Documento, y se refiere a la lista estandarizada de elementos de la página web a los que se puede acceder y manipular usando JavaScript (u otro lenguaje de script). Permite el acceso dinámico a través de la programación para acceder, añadir y cambiar dinámicamente contenido estructurado de las páginas web.  **Backend**  El backend es la capa que accede a los datos y software en general para su comunicación; es la parte trasera de cualquier página web. “Se trata de todo el conjunto del desarrollo que se encarga de que una página funcione y de que lo haga como lo hace, pero al mismo tiempo es invisible para el usuario, que solo ve lo visual y gráfico”. (NeoAttack, 2019)  Es el complemento del Frontend, los cuales al trabajar en armonía forman, en conjunto construyen una applicación web que además de la usabilidad y el buen rendimiento deberá brinda comunicación hacia el interior.  Entre los lenguajes de programación empleados en el desarrollo del Backend nos encontramos con PHP, Java, y .Net, para llevar a cabo las tareas tales como la lógica de la web, las conexiones con bases de datos, el uso de librerías, e incluso las operaciones llevadas a cabo por distintas funciones disponibles.  De acuerdo a Robbins (2018), algunos sitios web simples son colecciones de documentos HTML estáticos y archivos de imágenes, pero la mayoría de los sitios comerciales tienen una funcionalidad más avanzada, como manejo de formularios, páginas generadas dinámicamente, carritos de compras, sistemas de administración de contenido, bases de datos, etc. Estas funciones son manejadas por aplicaciones web que se ejecutan en el servidor. Existen varios lenguajes de programación y frameworks(enumerados entre paréntesis) que se utilizan para crear aplicaciones web, que incluyen:  -PHP (Laravel, CodeIgniter, Symfony, CakePHP)  -Python (Django, TurboGears)  -Ruby (Ruby on Rails, Sinatra)  -JavaScript (Angular, Node.js, Rhino, SpiderMonkey)  -Java (Spring MVC, Java Server Faces, Spring Boot, Grails, Struts)  -ASP.Net (DotNetNuke, ASP.Net MVC)    **Hosting**  Robbins (2018), en su publicación *Learning Web Design*, recomienda que, si trabajamos en un sitio comercial o si tomamos muy en serio nuestra presencia personal en la web, debemos alquilar espacio en el servidor de un servicio de alojamiento profesional (“hosting”, en inglés). Para lo cual se estará pagando por un espacio en uno de sus servidores, una cantidad de ancho de banda por mes y soporte técnico. También se pueden incluir servicios adicionales como listas de correo, carritos de compras, etc. El servicio de alojamiento es responsable de asegurarse de que nuestro sitio esté en línea y disponible las 24 horas del día, los 7 días de la semana.”  Las empresas de alojamiento suelen ofrecer una amplia gama de paquetes de servidores, desde unos pocos megabytes (MB) de espacio y una dirección de correo electrónico hasta soluciones de comercio electrónico con todas las funciones y muchos detalles. Por supuesto, cuanto más espacio en el servidor y más funciones, mayor será el costo mensual.  De acuerdo a Robbins (2018), las siguientes son algunas preguntas que debe hacerse uno mismo o al cliente cuando se compren planes de alojamiento:  “-¿Es un sitio comercial o personal? Algunos servicios de alojamiento cobran tarifas más altas para sitios comerciales que para sitios personales.  -¿Necesita un nombre de dominio? Verifique si la compañía de hosting registrará un nombre de dominio para usted como parte del precio del paquete. Esto le ahorra un paso y el cargo adicional por el registro de dominio en otro lugar.  -¿Necesitas un servidor dedicado? La mayoría de los planes de alojamiento son para servidores compartidos, lo que, como parece, significa que su sitio compartirá espacio en una computadora con muchos otros sitios. Para la mayoría de los sitios, esto está bien, aunque es importante tener en cuenta que el tráfico excesivo a otro sitio en el servidor puede afectar el rendimiento de su sitio. Los sitios comerciales más grandes donde el rendimiento es crítico optan por un servidor dedicado para que puedan aprovechar la potencia de procesamiento total de esa máquina. Los planes de servidores dedicados tienden a ser significativamente más caros que los planes compartidos, pero puede ser un dinero bien gastado para sitios de procesamiento intensivo. Otra alternativa es el alojamiento en la nube.  -¿Cuánto tráfico obtendrás? Asegúrese de prestar atención a la cantidad de transferencia de datos o visitantes que le permiten por mes. Para mantenerse competitivos, muchas empresas de alojamiento ofrecen ancho de banda ilimitado para todos sus niveles de paquetes. Los planes menos costosos generalmente tienen límites en el ancho de banda o en la cantidad de visitantes, así que asegúrese de verificar dichos límites si anticipa mucho tráfico o si está sirviendo videos u otros archivos con uso intensivo de datos.  -¿Cuántas cuentas de correo electrónico necesitas? Tenga en cuenta cuántas personas querrán el correo electrónico en ese dominio cuando compre el paquete de servidor correcto. Si necesita muchas cuentas de correo electrónico, es posible que deba ir con un paquete más robusto y de mayor precio.  - ¿Necesitas funcionalidad extra? Muchos servicios de alojamiento ofrecen características especiales del sitio web, algunos vienen como parte de su servicio estándar y otros cuestan dinero extra. Se extienden desde bibliotecas de scripts elegantes (para formularios de correo electrónico o libros de visitas) hasta soluciones completas y seguras de comercio electrónico. Cuando compre espacio, considere si necesita funciones adicionales, como carritos de compra, servidores seguros (para transacciones con tarjeta de crédito), un servidor de transmisión de medios (para transmisión de audio y video), listas de correo, etc.  - ¿Te sientes cómodo con su nivel de soporte técnico? Eche un vistazo a las políticas de la empresa de alojamiento y registre el soporte técnico. ¿Proporcionan un número de teléfono (preferible), chat en vivo en línea con técnicos o simplemente una dirección de correo electrónico para servicio al cliente? Es importante saber que su empresa de alojamiento estará allí para responder sus preguntas con prontitud.  -¿Harán copias de seguridad periódicas? Pregunte si la empresa de alojamiento realiza copias de seguridad periódicas de sus datos en caso de que haya un problema con el servidor.  - ¿Necesitas ayuda con el diseño y desarrollo? Si necesita un sitio pero no está preparado para hacer un gran diseño, puede elegir uno de los servicios de alojamiento que se especializan en proporcionar plantillas e interfaces de arrastrar y soltar para crear páginas. Algunos también ofrecen comercio electrónico y otras funciones avanzadas. La desventaja de usar sitios que generan páginas para usted es que pueden encerrarlo en su alojamiento porque no puede simplemente empacar sus archivos HTML y salir. Esto se aplica a cualquier servicio de alojamiento que ofrezca herramientas de creación de páginas.”  **Protocolos(http, https)**  Hay muchas formas en que la información se puede pasar entre las computadoras, incluido el correo electrónico, la transferencia de archivos (FTP) y muchos modos más especializados sobre los que se construye Internet. Estos métodos estandarizados para transferir datos o documentos a través de una red se conocen como protocolos.  Como se mencionó anteriormente, la Web utiliza un protocolo llamado HTTP (Protocolo de transferencia de hipertexto) para comunicar y transferir documentos a través del Internet.  **http://**  Debido a que casi todas las páginas web usan el Protocolo de transferencia de hipertexto, la parte http: // a menudo solo está implícita. Este es el caso cuando los nombres de los sitios se anuncian en forma impresa o en la televisión, como una forma de mantener la URL fácil de recordar. Además, los navegadores están programados para agregar http: // automáticamente como una conveniencia para ahorrarle algunas pulsaciones de teclas. Puede parecer que lo está omitiendo, pero se está enviando al servidor detrás de escena.  Lo primero que hace una URL es definir el protocolo que se utilizará para esa transacción en particular. Las letras HTTP le permiten al servidor saber usar el Protocolo de transferencia de hipertexto o entrar en "modo web". (Robbins, 2018, pp. 24)  **https://**  A veces, cuando se utiliza una URL que comienza con https://, esto es una indicación de que es una transacción de servidor segura. Los servidores seguros tienen dispositivos de cifrado especiales que ocultan contenido delicado, como números de tarjetas de crédito, mientras se transfieren desde y hacia el navegador. Es recomendable hacer uso de éstos al realizar compras en línea para agregarle seguridad. (Robbins, 2018, pp. 24)  **Modelo Cliente-Servidor**  “Es un modelo de diseño de software en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados servidores. (Wikipedia, 2013)  En la ilustración 6, podemos observar como varios clientes realizan peticiones a un servidor a través de Internet, siendo el servidor, quien da respuestas a cada una de estas peticiones sin perder de vista quien las realizó.  Figura 01. Modelo Cliente Servidor  Ilustración 6. modelo cliente-servidor  **Servidor**  Las computadoras que componen Internet, debido a que "entregan" documentos a pedido, estas computadoras se conocen como servidores. Para que una computadora sea parte de la Web, debe estar ejecutando un software de servidor web especial que le permita manejar las transacciones del Protocolo de transferencia de hipertexto. Los servidores web también se denominan "servidores HTTP". La función del software del servidor es esperar una solicitud de información, luego recuperar y enviar esa información lo más rápido posible. (Robbins, 2018, pp. 22)  Existen muchas opciones de software de servidor, pero las dos más populares son Apache (software de código abierto) y Microsoft Internet Information Services (IIS). Apache está disponible gratuitamente para computadoras basadas en Unix y viene instalado en Macs con Mac OS X. También hay una versión de Windows. Microsoft IIS forma parte de la familia de soluciones de servidor de Microsoft.  A cada computadora y dispositivo (módem, enrutador, teléfono inteligente, automóviles, etc.) conectado a Internet se le asigna una dirección IP numérica única (IP significa Protocolo de Internet). Por ejemplo, la computadora que aloja oreilly.com tiene la dirección IP 208.201.239.100. Todos esos números pueden ser vertiginosos, así que afortunadamente, el Sistema de nombres de dominio (DNS) se desarrolló para permitirnos referirnos a ese servidor por su nombre de dominio, "oreilly.com". La dirección IP numérica es útil para software de computadora, mientras que el nombre de dominio es más accesible para los humanos. Hacer coincidir los nombres de dominio de texto con sus respectivas direcciones IP numéricas es el trabajo de un servidor DNS separado.  Es posible configurar su servidor web para que más de un nombre de dominio se asigne a una sola dirección IP, lo que permite que varios sitios compartan un solo servidor.  **Cliente**  El software que realiza la solicitud se llama cliente. Las personas usan navegadores de escritorio, navegadores móviles y otras tecnologías de asistencia (como lectores de pantalla) como clientes para acceder a documentos en la Web. El servidor devuelve los documentos para que los muestre el navegador. (Robbins, 2018, pp. 23)  Las solicitudes y respuestas se manejan a través del protocolo HTTP, mencionado anteriormente. Aunque hemos estado hablando de "documentos", HTTP se puede utilizar para transferir imágenes, películas, archivos de audio, datos, secuencias de comandos y todos los demás recursos web que comúnmente componen sitios web y aplicaciones.  Es común pensar en un navegador como una ventana en el monitor de una computadora con una página web mostrada en él. Estos se conocen como navegadores gráficos o navegadores de escritorio y durante mucho tiempo, fueron el único juego de visualización web en la ciudad. Los navegadores de escritorio más populares a partir de este escrito incluyen Internet Explorer para Windows, Chrome, Firefox y Safari, con Opera en la parte posterior. Sin embargo, en la actualidad, cada vez más personas acceden a la Web mientras viajan utilizando clientes de navegación integrados en teléfonos móviles o tabletas, pantallas inteligentes, relojes inteligentes, etc.  También es importante tener en cuenta las experiencias web alternativas. Los usuarios con discapacidades visuales pueden estar escuchando una página web leída por un lector de pantalla (o simplemente haciendo que su texto sea extremadamente grande). Los usuarios con movilidad limitada pueden usar dispositivos de asistencia para acceder a los enlaces y escribir. Los sitios que creamos deben ser accesibles y utilizables para todos los usuarios, independientemente de sus experiencias de navegación.  Incluso en los navegadores de escritorio que nos presentaron por primera vez al amplio mundo de la Web, las páginas pueden verse y funcionar de manera diferente de un navegador a otro. Esto se debe a la compatibilidad variable para las tecnologías web y la capacidad de los usuarios para establecer sus propias preferencias de navegación.  A menudo, en el diseño web, escuchará referencias a aplicaciones "del lado del cliente" o "del lado del servidor". Estos términos se usan para indicar qué máquina está haciendo el procesamiento. Las aplicaciones del lado del cliente se ejecutan en la máquina del usuario, mientras que las aplicaciones y funciones del lado del servidor utilizan la potencia de procesamiento de la computadora del servidor.  **Microservicios**  Los microservicios son un sistema de desarrollo software que en los últimos años ha gozado de una gran popularidad por la manera en que influyen favorablemente en los proyectos, en aspectos como el tiempo, rendimiento y escalabilidad de los mismos.  Los microservicios o microservices proponen su propia arquitectura. Mientras que en una arquitectura monolítica la aplicación es desarrollada como una única unidad, una arquitectura de microservicios funciona con un conjunto de pequeños servicios que se ejecutan de manera independiente y autónoma. Incluso cada uno de ellos puede estar en un lenguaje de programación diferente. Este tipo de servicios nos permite contar con infraestructura IT más flexibles y adaptables, ya que para modificar un único servicio no es necesario alterar el resto de la infraestructura.  “Mientras que en una arquitectura monolítica la aplicación es desarrollada como una única unidad, una arquitectura de microservicios funciona con un conjunto de pequeños servicios que se ejecutan de manera independiente y autónoma.”  Los microservices se comunican entre sí a través de peticiones, por ejemplo, de HTTP a sus API. Para ello, debe de haber un número de mínimo de microservicios cuya función sea gestionar elementos comunes.  Por ello para aplicar la nueva arquitectura que proponen los microservicios, sobre todo en soluciones IT ya desarrollada, deberemos actuar con una arquitectura en dos capas. Una capa que actúe de manera interna y la segunda capa de manera externa.  **Cloud Computing**  Para las soluciones basadas en microservicios es necesario abordar los temas de Cloud Computing (Cómputo en la Nube)  La computación en la nube (del inglés cloud computing), es un paradigma que permite ofrecer servicios de computación a través de una red, que usualmente es internet.  En otras palabras, las nubes son entornos de TI que extraen, agrupan y comparten recursos escalables en una red. Suelen crearse para habilitar el cloud computing, que consiste en ejecutar cargas de trabajo dentro del sistema. Sin embargo, las nubes y el cloud computing no son tecnologías en sí mismas.  Las nubes son entornos: sitios donde se ejecutan las aplicaciones.  El cloud computing es una acción: la función que se encarga de ejecutar cierta carga de trabajo en una nube.  Las tecnologías son elementos: sistemas de software y hardware que se utilizan para diseñar y usar las nubes.  Existen tres tipos de entrega o servicio proporcionado en la nube:   * **SaaS.** El Software como servicio (SaaS) ofrece a los usuarios una aplicación en completo funcionamiento, así como la plataforma en la que se ejecuta y la su infraestructura de TI que necesita, a través de una conexión de red. Normalmente esta es la forma en que se ofrecen las aplicaciones de nube. * **PaaS.** La Plataforma como servicio (PaaS) es una modalidad del cloud computing en la cual un tercero brinda el sistema de hardware y una plataforma de software de aplicaciones. La PaaS es ideal principalmente para los desarrolladores y los programadores, ya que permite que el usuario desarrolle, ejecute y gestione sus propias aplicaciones sin tener que diseñar ni mantener la infraestructura ni la plataforma que suelen estar relacionadas con el proceso. El proveedor aloja el hardware y el software en su propia infraestructura, y ofrece la plataforma al usuario como una solución integrada, una pila de soluciones o un servicio a través de Internet. * **IaaS.** La infraestructura como servicio, también conocida como servicios de infraestructura en la nube, es una forma de cloud computing que ofrece a los usuarios finales una infraestructura de TI a través de Internet. Por lo general, se la asocia con la informática sin servidor.   **Diferencias entre PaaS, IaaS y SaaS**  En el caso de la IaaS, el proveedor gestiona la infraestructura por usted (es decir, los servidores, la red, la [virtualización](https://www.redhat.com/es/topics/virtualization) y [el almacenamiento](https://www.redhat.com/es/topics/data-storage)) a través de una nube. La infraestructura se alquila, y el usuario accede a ella con una [interfaz de programación de aplicaciones (API)](https://www.redhat.com/es/topics/api) o un panel. El usuario gestiona el sistema operativo, las aplicaciones y el [middleware](https://www.redhat.com/es/topics/middleware), mientras que el proveedor se encarga de los sistemas de hardware, las redes, los discos duros, el almacenamiento y los servidores. Además, es el responsable de prevenir las interrupciones, hacer reparaciones y solucionar los problemas de hardware.  En el SaaS, el proveedor gestiona una aplicación por usted. Este se encarga de las actualizaciones, las correcciones de errores y demás tareas de mantenimiento general del software, mientras que usted se conecta a él a través de un explorador web o una API. También se elimina la necesidad de tener una aplicación instalada en cada computadora individual.  La PaaS es una muy buena opción para los desarrolladores y los programadores que tienen ideas y escriben el código para hacerlas realidad, pero que no tienen o no desean tener los equipos y las complicaciones de mantenerlos en sus propias instalaciones.  Pueden sincronizar el código con una PaaS y ejecutar su aplicación usando el hardware y el software del proveedor, quien se encarga de su mantenimiento. Esto despeja el camino y permite un mayor desarrollo e innovación con menos distracciones, además de que reduce la cantidad de ajustes y codificación de la infraestructura. Dado que la PaaS se encuentra en la nube, también ofrece escalabilidad y fácil migración.  En la siguiente imagen puede observarse el nivel de responsabilidad en la administración de las soluciones de TI que implica cada uno de los servicios de Cloud Computing comparado con el tradicional On-Site.    **Amazon Web Services**  Amazon Web Services (AWS) es la plataforma en la nube más adoptada y completa en el mundo, que ofrece más de 200 servicios integrales de centros de datos a nivel global. Millones de clientes, incluso las empresas emergentes que crecen más rápido, las compañías más grandes y los organismos gubernamentales líderes, están usando AWS para reducir los costos, aumentar su agilidad e innovar de forma más rápida.  Ver video What is AWS[**https://www.youtube.com/watch?v=a9\_\_D53WsUs**](https://www.youtube.com/watch?v=a9__D53WsUs)  A continuación, se describen algunos de los servicios de AWS:  **AWS API Gateway**  Amazon API Gateway es un servicio completamente administrado que facilita a los desarrolladores la creación, la publicación, el mantenimiento, el monitoreo y la protección de API a cualquier escala. Las API actúan como la "puerta de entrada" para que las aplicaciones accedan a los datos, la lógica empresarial o la funcionalidad de sus servicios de backend. Con API Gateway, puede crear API RESTful y API WebSocket que permiten aplicaciones de comunicación bidireccional en tiempo real. API Gateway admite cargas de trabajo en contenedores y sin servidor, así como aplicaciones web.  API Gateway gestiona todas las tareas implicadas en la aceptación y el procesamiento de hasta cientos de miles de llamadas a API simultáneas, entre ellas, la administración del tráfico, compatibilidad con CORS, el control de autorizaciones y acceso, la limitación controlada, el monitoreo y la administración de versiones de API. API Gateway no requiere pagos mínimos ni costos iniciales. Se paga por las llamadas a las API que se reciben y por la cantidad de datos salientes transferidos; además, con el modelo de precios por niveles de API Gateway, puede reducir sus costos a medida que cambie la escala de uso de las API.  **AWS Lambda**  AWS Lambda es un servicio de informática sin servidor que ejecuta código en respuesta a eventos y administra automáticamente los recursos informáticos subyacentes. Puede usar AWS Lambda para ampliar la funcionalidad de otros productos de AWS con lógica personalizada o bien crear servicios back-end propios que funcionen con el nivel de seguridad, rendimiento y escala de AWS. AWS Lambda puede ejecutar código automáticamente en respuesta a varios eventos, como solicitudes HTTP a través de Amazon API Gateway, modificaciones realizadas en objetos en buckets de Amazon S3, actualizaciones de tablas en Amazon DynamoDB y transiciones de estado en AWS Step Functions.  Lambda ejecuta el código en una infraestructura informática de alta disponibilidad y se encarga de la administración integral de los recursos informáticos, incluido el mantenimiento del servidor y del sistema operativo, el aprovisionamiento de capacidad y el escalado automático, la implementación de parches de seguridad y código, así como la monitorización de código y los registros. Lo único que tiene que hacer es proporcionar el código.  **AWS S3**  Amazon Simple Storage Service (Amazon S3) es un servicio de almacenamiento de objetos que ofrece escalabilidad, disponibilidad de datos, seguridad y rendimiento líderes en el sector. Gracias a Amazon S3, clientes de todos los tipos y sectores pueden almacenar y proteger cualquier volumen de datos para los más variados fines, como usarlos en lagos de datos, sitios web, aplicaciones móviles, procesos de copia de seguridad y restauración, operaciones de archivado, aplicaciones empresariales, dispositivos IoT y análisis de big data. Amazon S3 proporciona características de administración fáciles de utilizar que le permiten organizar los datos y configurar sofisticados controles de acceso con objeto de satisfacer sus requisitos empresariales, organizativos y de conformidad. Amazon S3 está diseñado para ofrecer una durabilidad del 99,999999999 % (11 nueves) y almacena datos de millones de aplicaciones para empresas de todo el mundo.  **AWS SQS**  Amazon Simple Queue Service (SQS) es un servicio de colas de mensajes completamente administrado que permite desacoplar y ajustar la escala de microservicios, sistemas distribuidos y aplicaciones sin servidor. SQS elimina la complejidad y los gastos generales asociados con la gestión y el funcionamiento del middleware orientado a mensajes, y permite a los desarrolladores centrarse en la diferenciación del trabajo. Con SQS, puede enviar, almacenar y recibir mensajes entre componentes de software de cualquier volumen, sin pérdida de mensajes ni la necesidad de que otros servicios estén disponibles. Comience a usar SQS en minutos con la consola, la interfaz de línea de comandos o el SDK de AWS de su elección, y tres comandos simples.  **AWS SNS**  Amazon Simple Notification Service (Amazon SNS) es un servicio de mensajería completamente administrado para la comunicación aplicación a aplicación (A2A) y aplicación a persona (A2P).  La funcionalidad de publicación/suscripción A2A brinda temas para la mensajería de alto rendimiento, de muchos a muchos, basada en push entre sistemas distribuidos, microservicios y aplicaciones sin servidores controladas por eventos. Mediante el uso de temas de Amazon SNS, los sistemas de publicadores pueden distribuir los mensajes a una gran cantidad de sistemas de suscriptores, entre otros, colas de Amazon SQS, funciones de AWS Lambda y puntos de enlace HTTPS, para procesamiento paralelo y Amazon Kinesis Data Firehose. La funcionalidad A2P permite que envíe mensajes a usuarios a escala a través de SMS, push móviles y correo electrónico.  **AWS CloudFormation**  AWS CloudFormation le ofrece una forma sencilla de modelar un conjunto de recursos relacionados de AWS y de terceros, aprovisionarlos de manera rápida y consistente y administrarlos a lo largo de sus ciclos de vida tratando la infraestructura como un código. La plantilla de CloudFormation describe los recursos que desea y sus dependencias para que los pueda lanzar y configurar juntos como una pila. Puede usar la plantilla para crear, actualizar y eliminar toda una pila como una única unidad, tantas veces como sea necesario, en lugar de administrar los recursos de manera individual. Puede administrar y aprovisionar pilas en varias cuentas y regiones de AWS.  **AWS IAM**  Con AWS Identity and Access Management (IAM) puede administrar el acceso a los servicios y recursos de AWS de manera segura. Además, puede crear y administrar usuarios y grupos de AWS, así como utilizar permisos para conceder o negar el acceso de estos a los recursos de AWS.  IAM es una característica de su cuenta de AWS que se ofrece sin cargos adicionales. Solo se le cobrará por la utilización de los demás servicios de AWS por parte de sus usuarios.  **EJERCICIO EN CLASE**  Nota: Los trabajos copia se anulan (tanto el original como la copia)   * Guarda el documento con el siguiente formato Ensayo1\_ApellidosNombre.docx, suba su trabajo a la liga correspondiente en la plataforma educativa. | | |
| **Referencias**  AWS. (2021). *Informática en la nube con AWS.* Recuperado de <https://aws.amazon.com/es/what-is-aws/?nc1=f_cc>  AWS. (2021). Amazon API Gateway. Recuperado de <https://aws.amazon.com/es/api-gateway/>  AWS. (2021). Características de AWS Lambda. Recuperado de <https://aws.amazon.com/es/lambda/features/>  AWS. (2021). Amazon S3. Recuperado de <https://aws.amazon.com/es/s3/>  AWS. (2021). Amazon Simple Queue Service. Recuparado de <https://aws.amazon.com/es/sqs/>  AWS. (2021). Amazon Simple Notification Service. Recuperado de <https://aws.amazon.com/es/sns/?whats-new-cards.sort-by=item.additionalFields.postDateTime&whats-new-cards.sort-order=desc>  AWS. (2021). Amazon CloudFormation. Recuperado de <https://aws.amazon.com/es/cloudformation/>  AWS. (2021). AWS Identity and Access Management (IAM). Recuperado de https://aws.amazon.com/es/iam/  Gómez, M. C. y Cervantes, J. (2017). *Introducción a la Programación Web con Java: JSP, JavaServer Faces.* Recuperado de <http://www.cua.uam.mx/pdfs/revistas_electronicas/libros-electronicos/2017/java/Java.pdf>  Internerpasoapaso.com. (2019). *¿Qué es, para que sirve y como funciona una Web App? Definición, tipos y ejemplos.* Recuperado de <https://internetpasoapaso.com/web-app/>  Latorre, M. (2020). *Historia de la Web 1.0, 2.0, 3.0 y 4.0.* Recuperado de <http://marinolatorre.umch.edu.pe/historia-de-la-web-1-0-2-0-3-0-y-4-0/>  NeoAttack. (2019). *Diccionario de Marketing.* Recuperado de <https://neoattack.com/neowiki/>  Redhat. (2021). *Computación en la nube: ¿Qué es el cloud computing?.* Recuperado de <https://www.redhat.com/es/topics/cloud>  Robbins, J. (2018). *Learning Web Design: A Beginner's Guide to Html, Css, Javascript, and Web Graphics.* Recuperado de <https://books.google.com.mx/books?id=uN7kAT9FidcC&printsec=frontcover&hl=es&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false>  Vallejos, O. (2017). *Introducción a Internet.* Recuperado de <http://ing.unne.edu.ar/pub/internet.pdf>  Wikipedia. (2013). *Cliente-servidor*. Recuperado de <https://es.wikipedia.org/wiki/Cliente-servidor> | | |