**Datos de identificación del programa de formación**

| PROGRAMA DE FORMACIÓN | Desarrollo de aplicaciones web full stack |
| --- | --- |

| COMPETENCIA | 220501095. Diseño de los artefactos necesarios para la solución de *software.* | RESULTADOS DE APRENDIZAJE | 220501095-02. Seleccionar la arquitectura de *software* según características del proyecto a desarrollar. |
| --- | --- | --- | --- |

| NÚMERO DEL COMPONENTE FORMATIVO | 7 |
| --- | --- |
| NOMBRE DEL COMPONENTE FORMATIVO | Selección de la arquitectura de *software* |
| BREVE DESCRIPCIÓN | En el presente componente formativo se estudiarán diferentes conceptos como *networking,* modelos de arquitectura, lenguajes de programación, servidores web, protocolos de comunicación, entre otros, los cuales le brindarán las bases para su trabajo como desarrollador web *full stack.* |
| PALABRAS CLAVE | *Networking,* infraestructura, servidores, lenguajes de programación |

| ÁREA OCUPACIONAL | 6 - VENTAS Y SERVICIOS |
| --- | --- |
| IDIOMA | Español |

# **Tabla de contenido**

***1. Networking***

**2. Sistemas operativos de red e infraestructura**

**3. Modelos de arquitectura**

**4. Lenguajes de programación y *stack* tecnológico**

**5. Gestor de base de datos y servidores**

**6. Protocolos de comunicación**

**Introducción**

| Cuadro de texto |
| --- |
| Apreciado aprendiz, bienvenido a este componente formativo, donde se darán algunas bases para su futuro desempeño como desarrollador web *full stack.*  En el siguiente video conocerá, de forma general, la temática que se estudiará a lo largo del componente formativo. |

**Guion de video introductorio**

| **Tipo de recurso** | Video animación 2D | | | |
| --- | --- | --- | --- | --- |
| **NOTA** |  | | | |
| **Título** | Introducción | | | |
| **Escena** | **Imagen** | **Sonido** | **Narración** | **Texto** |
| **1** | Oficina diversa: Programador de TI trabajando en computadoras de escritorio. Especialista Masculino Creando Software Innovador Desarrollando Aplicación, Programa, Videojuego. Terminal con lenguaje de codificación. Más de hombro  Desarrollo web o de aplicaciones, concepto de negocios y tecnología. Pantalla de web que desarrolla código javascript. Desarrollo y diseño de aplicaciones de Internet. Aplicación de codificación por desarrollador de programadores  Vista de una interfaz de usuario de la interfaz de aplicación en un smartphone: representación 3d |  | Bienvenido a este componente formativo con el que inicia un recorrido por el importante mundo de la programación.  Antes de iniciar su desempeño como desarrollador web es necesario que obtenga el conocimiento de todos los conceptos básicos expuestos en el presente componente que le van a ayudar a entender mejor la construcción y el funcionamiento de las aplicaciones que desarrolle como profesional en este campo de la tecnología. | Programación  Desarrollador web  Construcción y funcionamiento de aplicaciones |
| **2** | Qué es el networking? - Loudroom.co  Sistemas operativos - Tipos, funcionamiento y Concepto |  | Estudiará algunos de los conceptos y temáticas sobre *networking* o redes, incluyendo características y tipos que forman parte de su estructura. Sistemas operativos como Windows, Linux, MacOS e infraestructura tecnológica sobre equipos de cómputo, los servidores más conocidos y utilizados en el mercado, memoria RAM, procesadores, almacenamiento. | *Networking*  Sistemas operativos  Servidores |
| **3** | La guía definitiva de la arquitectura de software | by Fernando Valdes |  Medium |  | También harán parte de este importante recorrido algunos patrones y modelos arquitectónicos de *software* para aplicaciones web*,* como: patrón cliente-servidor, patrón arquitectónico Modelo Vista Controlador (MVC), arquitectura orientada a servicios (SOA) y microservicios. | Patrones y modelos arquitectónicos de *software* |
| **4** | Buscar imágenes de los diferentes lenguajes de programación que se mencionan e irlas colocando a medida que se escuche el audio. |  | Para una mejor comprensión se abarcará el tema de lenguajes de programación, sus diferentes clasificaciones y algunos de los más importantes y demandados en la industria del *software* como: JavaScript, Python, PHP, Django, C#, ASP.NET, Ruby, Perl, C++, NodeJS, etc. Se estudiarán temáticas sobre *stack* tecnológicos y ejemplos de estos como: MEAN y MERN, LAMP, al igual que algunos gestores de bases de datos, tanto de base de datos relacionales como no relacionales. Entre ellos se encuentran: MySQL, PostgreSQL, Oracle, MongoDB, Microsoft SQL Server, Redis y Cassandra. | Lenguajes de programación |
| **5** | FTP, qué es y para qué sirve  bases-datos-sql-server-microsoft-gratis-mysql-postgreesql-mongodb | RINU -  Informática |  | Dentro de esta estructura de estudio también se abordan temáticas sobre servidores, tipos de servidores, como: servidores *web*, servidores de FTP, servidores de bases de datos con énfasis en algunos servidores web como: Apache, Microsoft IIS, Nginx, Google GWS y el tema de protocolos de comunicación y sus características: HTTP, FTP, XMPP, SMTP. | Servidores  Servidores *web*  Servidores de FTP  Servidores de bases de datos |
| **6** | Tips para contratar un programador web para tu proyecto |  | Estimado aprendiz: es durante el recorrido de este derrotero que logrará adquirir y afianzar los conocimientos que se han especificado con anterioridad. Esta ruta de aprendizaje le permitirá adquirir seguridad y conocimiento, fortalezas para el desempeño de sus funciones como programador web.  Usted cuenta con el apoyo y acompañamiento necesarios para ayudarle a aclarar dudas que puedan surgir durante el estudio de la temática. Recuerde leer cada uno de los ítems expuestos, al igual que el material de los enlaces recomendados que se suministran y que forman parte complementaria de este aprendizaje. |  |
| **7** | Lenguajes de programación: estos son los que mejor salario generan •  ENTER.CO |  | Recuerde que es sustancial la buena comunicación, en esta temática como en la vida cotidiana. Se espera que durante el recorrido utilice esta herramienta tan indispensable en todas las áreas. Muchos éxitos en esta fase de su estudio.  Son fundamentales su compromiso y la disciplina de estudio como ingredientes para culminar de manera satisfactoria el estudio de este componente. |  |
| **Nombre del archivo** | 228125\_v1 | | | |

**Desarrollo de contenido**

***1. Networking***

| Cuadro de texto |
| --- |
| Para comenzar, debemos tener claro que la conexión en terminales o estaciones de trabajo entre sí es fundamental para que las empresas o las personas lleven a cabo comunicaciones. Un ejemplo de una gran conexión de conexiones es el internet (del que luego hablaremos). Conexión bond connect networking unión concepto  El término *networking* en el presente contexto se refiere a: “redes”, redes de datos para comunicación e intercambio de información entre dispositivos o terminales. Básicamente una red de datos está compuesta por dispositivos, medios de comunicación y servicios. |

| **Tipo de recurso** | Slider Presentación | |
| --- | --- | --- |
| **Introducción** | Conozcamos algunos aspectos importantes del *networking.* | |
| **Dispositivos de red**  Se refiere a las terminales de trabajo como servidores, computadoras convencionales, tabletas, móviles, teléfonos IP, etc.; al igual que aparatos que logran conexión con estos como son: *routers, switches, hubs, firewalls*, etc. | | computer network connections icons and topology eps10  **Imagen:** 228125\_i2 |
| **Medios de comunicación**  Son los canales a través de los cuales viaja la información: cables UTP, de fibra óptica o wifi. Dependiendo de las conexiones que tenga con los medios mencionados, tendrá una velocidad de transmisión. Por ejemplo, la conexión con fibra óptica es indicada para largas distancias de transmisión de datos, los cables UTP son recomendables para una conexión de máximo 100 metros, el wifi tiene menos alcance de conexión, y puede tener más interferencia, lo que implica menos velocidad de conexión. | | Twisting Cable Tool Twisted Pair Ethernet UTP Cat 5 |
| **Servicios**  Se refiere a los programas y aplicaciones que están a lo largo de la red de comunicaciones. Ejemplo: servicio de correo, servicios de páginas web, etc.  El proceso de comunicación en una red de datos se asemeja al proceso de comunicación común,en el que participan de forma básica: un **emisor,** un **canal de** comunicación, un **receptor,** un **mensaje** y un **código**; puede existir **retroalimentación**.  Imagine que está chateandopor WhatsApp con un compañero del programa para ponerse de acuerdo en realizar una actividad. El **emisor** sería usted y el **receptor** su compañero, quien responde el **mensaje**;entonces**,** hay una **retroalimentación**. Supongamos que el mensaje es un texto (**código**), este necesita un medio para viajar, pero resulta que está conectado al wifi de su casa; así, primero viajaría por ondas magnéticas, luego el dispositivo que emite dicho wifi está conectado a un cable y por allí también viajaría la información, ese sería entonces el **canal** o medio de **comunicación**. | | **Imagen:** 228125\_i4Mensajes de chat smartphone, Sms en pantalla de teléfono móvil. Hombre, pareja de mujeres charlando, Mensajería usando aplicación de chat o red social. Conversación de dos personas a través del teléfono celular enviando mensajes. ilustración vectorial. |

| Cuadro de texto |
| --- |
| Ahora, en los medios de comunicación, es importante tener en cuenta la velocidad y frecuencia de conexión con cables UTP dependen de algunas categorías que se detallan en la siguiente tabla: |

**Tabla 1**

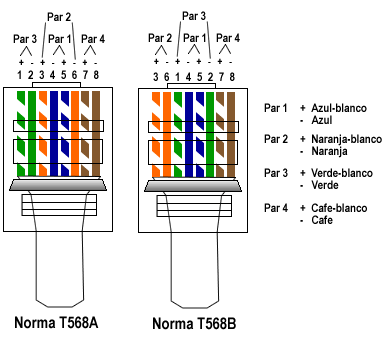
*Velocidad y frecuencia de conexión*

| **Categoría** | **Velocidad** | **Frecuencia** |
| --- | --- | --- |
| Cat 5 | 100 Mbits/s | 100 MHz |
| Cat 5E | 1000 Mbits/s | 100 MHz |
| Cat 6 | 1000 Mbits/s | 250 MHz |
| Cat 6ª | 10000 Mbits/s | 500 MHz |
| Cat 7 | 10000 Mbits/s | 600 MHz |

| Cuadro de texto |
| --- |
| Según la anterior tabla, debería configurar su cable con conectores RJ45, lo que en el común se escucha como “ponchar”, tal como lo muestra la siguiente imagen: |

**Figura 1**

*Configuración cable*



**Internet**

| Cuadro de texto |
| --- |
| El internet y la web son dos términos que se suelen confundir en el diario vivir, pero son distintos. La internet, es la llamada “red de redes”, no es una red única, y la web se ejecuta sobre internet. Es un “sistema distribuido”. No cabe duda de que ambos son de los avances tecnológicos más importantes en los últimos años, y de la informática en general.   Hoy en día basta con accesar a un navegador web (Chrome, Mozilla, Safari, etc.), ingresar la dirección de una página web y allí poder realizar un sinfín de operaciones: compras, apuestas, gestionar un negocio, ver películas.  Uno de sus objetivos como como desarrollador web *full stack* es que pueda construir aplicaciones mediante las que se realicen tareas como estas; y aún más. |

| Cuadro de texto |
| --- |
| El *networking* tiene unas características entre las que se encuentra la tipología y topología de redes. A continuación, conoceremos cada una de ellas. |

| **Tipo de recurso** | Infografía interactiva Punto caliente | |
| --- | --- | --- |
| **Texto introductorio** | En la tipología de redes se encuentran: | |
|  | | |
| **Código de la imagen** | 228125\_i6 | |
| **Punto caliente 1** | **LAN (Red de área local):** son redes privadas de escasos kilómetros de longitud que se encuentran en una misma vivienda, edificio o campus. | **LAN** |
| **Punto caliente 2** | **MAN (Red de área metropolitana):** es una red que abarca una ciudad; ejemplo: la televisión por cable. | **MAN** |
| **Punto caliente 3** | **WAN (Red de área amplia):** es aquella que abarca un área geográfica grande: un país o un continente. | **WAN** |

**Topología de redes**

| Cuadro de texto |
| --- |
| Representa la forma como se interconectan los dispositivos de una red. Se encuentran en los tipos mencionados anteriormente (LAN, WAN, MAN), y se clasifican en dos: física y lógica. El siguiente mapa conceptual nos presenta un resumen de la topología de redes. |

**Figura 2**

*Topología de redes*



| Cuadro de texto |
| --- |
| Ahora, en el *networking* encontramos dos modelos que debemos tener presente: el modelo OSI y el modelo TCP/IP. |

| **Tipo de recurso** | | Pestañas o tabs Verticales |
| --- | --- | --- |
| **Introducción** | | Conozcamos cada uno de ellos: |
| proxy server icon. Trendy linear proxy server logo concept on transparent background from Internet Security and Networking collection | | |
| **El modelo OSI** | El Modelo OSI (Interconexión de Sistemas Abiertos) se encuentra dentro de los **modelos de referencia:** son abstracciones para facilitar la comprensión de los protocolos de comunicación y la arquitectura de los sistemas utilizados para interrelacionar distintos programas y equipos*.*  Es un modelo propuesto por la Organización Internacional de Estándares (ISO) como un primer paso hacia la estandarización internacional de los protocolos utilizados en varias capas. | |
| **El modelo TCP/IP** | Es un modelo de referencia utilizado en **ARPANET** y en el **internet**.  Puede encontrar información sobre ARPANET en la página web oficial de **Mozilla**. | |

| Cuadro de texto |
| --- |
| Las capas de los modelos OSI y TCP/IP se resumen en la siguiente imagen: |

**Figura 3**

*Capas de los modelos OSI y TCP/IP*

Interfaz de usuario gráfica

Descripción generada automáticamente

**2. Sistemas operativos de red e infraestructura**

| Cuadro de texto |
| --- |
| Un sistema operativo (SO), en términos generales, es un *software* especial que “controla” el ***hardware***, el cual es aquella parte de un “sistema de cómputo” que se puede tocar: monitor, teclado, *mouse* o ratón, CPU, memoriaRAM; incluso, los mismos dispositivos de red. *Software* es aquello que no se puede tocar, por ejemplo, el paquete ofimático de Office, LibreOffice, un videojuego, una aplicación web, una aplicación de escritorio.  En el siguiente esquema se puede observar un resumen de las principales capas de un SO: |

**Figura 4**

*Principales capas de un SO*



**Tipos**

| Cuadro de texto |
| --- |
| Quizás conoce o ha escuchado sobre sistemas operativos para estaciones de trabajo convencionales, o quizás en estos momentos esté trabajando con alguno de ellos en su computadora: GNU/Linux, Windows, MacOS, Unix.   * GNU/Linux cuenta con distribuciones y versiones como Ubuntu (uno de los más usados) y Fedora, entre otros. * Windows, por ejemplo, tiene Windows 7, Windows 10, Windows 11. * Mac OS presenta Monterrey, Big Sur, Sierra.   Existen otros sistemas operativos, pero los mencionados son los más populares, conocidos y usados. |

| **Tipo de recurso** | Pestañas o *tabs* horizontales | |
| --- | --- | --- |
| **Introducción** | También existen versiones exclusivas de estos sistemas operativos para servidores y redes, pero con características especiales para trabajar en red. Estas versiones son: | |
| **Windows** | Propiedad de la compañía Microsoft Corporation. Entre las distribuciones y versiones más conocidas y usadas en el mercado actualmente se encuentra Microsoft Windows Server, que surgió a partir del año 2003; normalmente cada año se publica una versión, y su nombre está ligado con el año de distribución, por ejemplo: Windows Server 2012. Las versiones de Windows Server son de pago. | Descarga gratuita de Los Servidores De Microsoft, Windows Server, Los Servidores De Un Ordenador Imágen de Png  Imagen: 228125\_i10 |
| **GNU/Linux** | GNU es un sistema operativo de *software* libre, es decir, respeta la libertad de los usuarios. Linux es el Kernel o núcleo de GNU y la combinación de ambos, el sistema operativo GNU/Linux.  Existen versiones para servidores “gratuitos”: Ubuntu Server, CentOS, Fedora, y otras de pago como RedHat Enterprise Server. | Descarga gratuita de Gnulinux De Nomenclatura De La Controversia, Gnu, Linux Imágen de Png  Imagen: 228125\_i11 |
| **Mac OS X Server** | Son los menos usados en el mercado para propósito de servidores. | Apple anuncia el fin de macOS Server | iPhoneros  Imagen: 228125\_i12 |

| Cuadro de texto |
| --- |
| El sistema operativo puede manipular los componentes de un equipo de cómputo, como: administración de recursos, acceso a dispositivos de entrada y salida, servidor la interfaz entre sistema y usuario.  A su vez un SO está conformado por varios programas, y sus componentes principales son la gestión de procesos, entradas y salidas, memoria, sistema de archivos. Una vez se tiene el sistema operativo instalado en un dispositivo, al *hardware* en particular o un *software* especial para que pueda ser manipulado; es lo que se llama ***drivers***. Muchas veces se conecta algún periférico: impresora, cámara, lector de huella, etc., pero el computador no es capaz de reconocerlos, surge entonces la necesidad de instalar dichos *drivers* o controladores para que se puedan realizar los trabajos con este periférico de manera adecuada. Los sistemas operativos modernos ya poseen la mayoría de los *drivers* para los dispositivos periféricos comunes comerciales, simplemente se conecta y, de manera casi inmediata, es reconocido. |

**Infraestructura**

| Cuadro de texto |
| --- |
| Los servidores físicos son equipos especiales, normalmente cuentan con recursos físicos mayores a un equipo de cómputo convencional de uso personal. |

| **Tipo de recurso** | Acordeón tipo 2 |
| --- | --- |
| **Introducción** | Normalmente se les instala un sistema operativo para servidor y poseen los siguientes recursos: |
| En el centro de datos moderno: El ingeniero de TI mantiene el nuevo disco duro de disco duro preparado para instalar el equipo de hardware en el rack del servidor. Especialista en TI haciendo mantenimiento y actualización de hardware.  **Imagen:** 228125\_i13 | |
| **Memoria RAM**  Es un dispositivo de *hardware*, normalmente en forma de tarjetas modulares que se insertan en la placa base o *Motherboard*. Se encarga del almacenamiento e intercambio de información de manera temporal. Mientras trabaja con su equipo de cómputo estos datos son volátiles y suelen borrarse cuando se apaga la computadora. Tiene capacidades de almacenamiento en *bytes* y las memorias modernas vienen con capacidades hasta de 128 GB y 256 GB, con frecuencia de intercambio de información hasta 5 200 MHz, tanto para computadores personales como para servidores. | |
| **CPU**  Es la unidad de almacenamiento central o procesador, llamado muchas veces como el “cerebro” de la computadora porque es el que lleva a cabo las principales funciones de gestión y cálculos que se realizan en esta. | |
| **Disco duro o rígido**  Es donde se almacenan y persisten los archivos de la computadora o servidor, incluso en él se instalan el sistema operativo y otro tipo de información importante. Aún existen los discos duros de lectura convencionales de plato, discos duros de estado sólido y discos M2. | |
| **Tarjeta de red**  Es necesaria para la comunicación por cable, o inalámbrica, entre una o más estaciones de trabajo. | |

| Cuadro de texto |
| --- |
| La infraestructura convencional para una empresa, en lo que se refiere a redes y servidores, anteriormente estaba conformada por equipos físicos armados como dispositivos de redes (*switch, routers*, cableado), servidores físicos, *racks*, UPS, termostatos, sensores de humo, temblor, etc.; organizados y almacenados en un cuarto climatizado y “asegurado”. La tendencia en la actualidad para estas infraestructuras es la basada en la nube o *cloud*, más conocida como infraestructura como servicio (IaaS). |

| **Tipo de recurso** | Tarjetas avatar |
| --- | --- |
| **Introducción** | Los proveedores más conocidos en la actualidad para estos servicios son los gigantes de: |
| Amazon (Amazon Web Service) | Descarga gratuita de Amazoncom, Amazon Web Services, Logotipo Imágen de Png  **Imagen** 228125\_i14 |
| Microsoft (Azure) | Descarga gratuita de  Imágen de Png  **Imagen:** 228125\_i15 |
| Google (Google Cloud) | Descarga gratuita de Google Cloud Platform, Computación En La Nube, Google Imágen de Png  **Imagen:** 228125\_i16 |

| Cuadro de texto |
| --- |
| También se puede optar por una infraestructura híbrida, es decir, en donde se incluya infraestructura convencional e infraestructura en la nube. Las pequeñas, medianas y grandes empresas pueden construir su propia infraestructura de acuerdo con sus necesidades, ventajas y desventajas, costo / beneficio, y dependiendo de las variables tiempo y cantidad de clientes, entre otras. |

**3. Modelos de arquitectura**

| Cuadro de texto |
| --- |
| La mayoría de los conceptos e ideas de **arquitectura de *software*** provienen de arquitectura de edificios. Quizá no somos arquitectos, pero tenemos la idea de que un arquitecto, antes de llevar a cabo sus proyectos, diseña los planos de sus casas o edificios y presenta sus **estilos arquitectónicos**, puede presentar unos tiempos y materiales necesarios para la construcción.Industria 4.0: Arquitecto profesional moderno con audífonos de realidad virtual usa gestos para mover, diseñar, manipular edificios para la ciudad 3D. Software De Realidad Aumentada Mixta. Efecto visual especial de VFX  Existen estilos arquitectónicos propuestos y llevados a cabo por autores que han sido probados en grandes organizaciones y han funcionado con éxito, y por eso se usan aún; algunos no,o llegan nuevos que los sustituyen. Incluso, el arquitecto puede crear su propio estilo o patrón.  Arquitectura de *software* es una definición algo relativa. Se podría decir que no hay concepto exacto de ello, de hecho, puede ser complejo. A continuación presentamos una definición que parece bastante acertada y comprensible: |

| Tipo de recurso | Cajón de texto de color |
| --- | --- |
| “La arquitectura de *software* de un programa o computación sistema, es la estructura o estructuras del sistema que comprenden componentes de *software*, el externo de las propiedades visibles de esos componentes, y las relaciones entre ellos [sic]” (Garlan, D., s.f.). | |

| **Tipo de recurso** | Tarjetas Avatar | |
| --- | --- | --- |
| **Introducción** | Existen tres conceptos que tienden a confundirse: | |
| Patrón arquitectónico | |  |
| Patrón de diseño | |  |
| Estilo arquitectónico | |  |

| Cuadro de texto |
| --- |
| Un estilo arquitectónico, tal como lo hablamos en la comparación con edificios, es el cómo está construido: la forma como está hecho, la manera como se construye. Con respecto a los patrones arquitectónicos un estilo puede tener uno o varios. Entonces, un estilo arquitectónico de *software* se refiere a ese conjunto de decisiones para construir ese *software* en determinado contexto. |

| **Tipo de recurso** | Tarjetas Avatar | |
| --- | --- | --- |
| **Introducción** | Es así que cuando vamos a construir un *software* no debemos ir al grano sin antes analizar bien y responder las siguientes preguntas: | |
| ¿Qué tipo de *software* o aplicación vamos a construir? | |  |
| ¿Para quién la vamos a construir? | |  |
| ¿Para cuándo debe estar lista? | |  |

| Cuadro de texto |
| --- |
| Icono de concepto turquesa del ciclo de vida del desarrollo de software. Programar habilidades abstractas de la idea en línea delgada. Dibujo de contorno aislado. Trazo editable. Fuentes Arial, Myriad Pro-Bold utilizadas  En la fase de la arquitectura, donde vamos a representar el sistema con un alto nivel de abstracción, debemos tener claros todos los **requerimientos del sistema**. Esta fase se realiza antes de la fase de diseño del **Ciclo de vida de desarrollo del *software***.  Un patrón en general es algo que se sigue para solucionar un problema en particular; entonces, un patrón arquitectónico brinda soluciones a problemas en desarrollo de *software*, pero debemos ver los patrones arquitectónicos como una abstracción. Muchas veces se suelen mirar como ilustraciones particulares o líneas de código de un lenguaje de programación que solucionan algo, pero esto es un concepto errado, al igual que para los patrones de diseño.  Los patrones y estilos arquitectónicos nos van a permitir organizar los componentes del *software* final que se requiere. También aportan a sus atributos de calidad, que resultan de los **requerimientos no funcionales** del sistema.  La arquitectura puede ilustrarse, representarse o describirse con lenguajes y estándares basados en el Lenguaje de Modelado Unificado (UML). |

| Cuadro de texto |
| --- |
| Los estilos arquitectónicos se seleccionan dependiendo del tipo de aplicación a construir. Pueden ser: |

1. **Cliente-Servidor**

| Cuadro de texto |
| --- |
| Arquitectura Cliente-Servidor  Se cataloga como un estilo arquitectónico; en esta arquitectura están basadas la mayoría de las aplicaciones que usan internet. Como su nombre lo dice, tiene dos componentes: cliente y servidor, además del medio de comunicación entre estos. |

| Cuadro de texto |
| --- |
| A su vez, esta arquitectura tiene varios tipos: 2-tier (2 niveles o capas).  En esta arquitectura el cliente y el servidor se comunican entre sí sin ningún punto o nodo intermedio; aunque proporciona un servicio rápido, puede ocasionar agujeros de seguridad y problemas de rendimiento. Ejemplo: Internet Explorer utiliza una arquitectura de servidor de cliente de 2 niveles, 3-tier (3 niveles o capas); en esta, un nodo más ―llamado nivel medio― se encuentra entre el cliente y el servidor. El nivel medio toma la solicitud del cliente, lo autentica, aprueba y luego lo pasa al servidor. También se habla de n-tier (multinivel). |

1. **Arquitectura por capas (Layered)**

| Cuadro de texto |
| --- |
| También es catalogada como un estilo arquitectónico. En ejemplos de arquitectura por capas están el modelo OSI y el modelo TCP/IP, en donde hay una interacción entre los componentes de una misma capa o con las capas adyacentes. Todas trabajan por una tarea en común.  Esta arquitectura agrupa las responsabilidades del *software* en varias capas débilmente acopladas. Las capas pueden comunicarse entre sí a través de interfaces bien definidas. Dado que las capas están débilmente acopladas, pueden residir en una sola computadora o pueden estar distribuidas en varios ordenadores. |

| **Tipo de recurso** | Infografía interactiva Punto caliente | |
| --- | --- | --- |
| **Texto introductorio** | Por lo general un sistema de *software* se divide en tres capas: | |
|  | | |
| **Código de la imagen** | 228125\_i20 | |
| **Punto caliente 1** | Esta capa incorpora funcionalidades de la interfaz de usuario. A través de esta capa el usuario puede interactuar con el sistema. | **Capa de presentación** |
| **Punto caliente 2** | Esta capa incorpora la lógica de negocio del sistema. | **Capa de negocios** |
| **Punto caliente 3** | Esta capa incorpora servicios de datos, servicios de red y otros servicios de infraestructura. | **Capa de infraestructura** |

| Tipo de recurso | Cajón de texto de color |
| --- | --- |
| **Observación**: cuando se llevan al español los términos *tier y layer* se tienden a confundir, por eso es mejor hablar de *tier* como nivel y de *layer* como capa. *Tier* para referirse a lo físico, *layer* como parte del código o parte lógica. | |

1. **Arquitectura orientada a servicios (SOA)**

| Cuadro de texto |
| --- |
| SOA - Acrónimo de arquitectura orientada a servicios. fondo de concepto de negocio. concepto de ilustración vectorial con palabras clave e iconos. ilustración con iconos para banner web, volante, aterrizaje  Es un estilo arquitectónico para sistemas distribuidos. Al igual que la arquitectura cliente – servidor, SOA, examina cómo construir, usar y combinar servicios. En lugar de crear grandes paquetes de *software* que lo hagan todo…**divide y vencerás.** SOA alcanza los objetivos del *software* mediante la creación y el uso de servicios y el diseño de una arquitectura que admita su uso. |

1. **Microservicios**

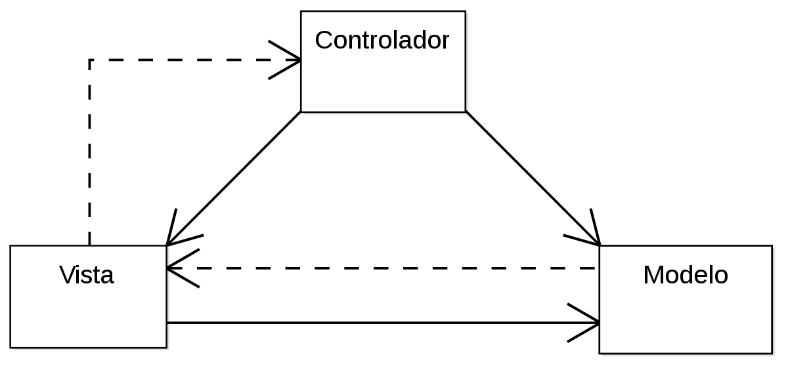
| Cuadro de texto |
| --- |
| Engranajes con un vector de icono de esquema de cubos. Símbolo de arquitectura de microservicios. Signo de diagrama de flujo del modelo de programación de software.  Las pioneras de esta arquitectura fueron importantes plataformas de entretenimiento, servicio de *streaming* y de comercio electrónico. Fue tan bueno el resultado, que otras empresas decidieron adoptar el modelo y seguirlo. Tales plataformas se consideran también un estilo arquitectónico consistente en que las aplicaciones se dividen en sus elementos más pequeños e independientes entre sí. A diferencia del enfoque tradicional y monolítico de las aplicaciones, en el que todo se compila en una sola pieza, los microservicios son elementos independientes que funcionan en conjunto para llevar a cabo las mismas tareas. Cada uno de esos elementos o procesos es un microservicio. |

1. **El Modelo Vista Controlador (MVC)**

| Cuadro de texto |
| --- |
| Según el concepto de patrón arquitectónico, el MVC se puede considerar así, un patrón de arquitectura, que, de hecho, es bastante común en el diseño de *software*. Cabe resaltar que otros autores lo pueden considerar como una estructura, y otros, de manera errada, mencionan que es un patrón de diseño. |

**Figura 5**

*Representación del MVC*



| Cuadro de texto |
| --- |
| La idea de este patrón es separar la vista (interfaz de usuario) con la lógica de negocios y los datos:   * Modelo: mantiene el estado de una aplicación. * Vista: muestra la interfaz de usuario a dicho usuario. * Controlador: procesa las acciones del usuario (enviadas desde la vista) para actualizar el modelo y reenviar esas actualizaciones a la vista.   Con MVC puede lograr el diseño de buenas prácticas de código, creando estructuras de *namespaces*, paquetes o directorios para tener un proyecto organizado a nivel de código que puede ser más fácil de mantener y entender. |

| Tipo de recurso | Cajón de texto de color |
| --- | --- |
| Un patrón arquitectónico es solo una abstracción. El MVC, por ejemplo, se puede ver o comparar con un sinnúmero de actividades o sucesos de la vida real. Imagínese llegar a una cafetería, sentarse en la terraza y pedir una bebida (usted sería la vista) a la persona que atiende (controlador), esta persona va a la cocina y les dice a los encargados (modelo) que preparen su pedido; ellos lo preparan, lo entregan al mesero, y el mesero lleva su pedido y usted lo recibe. | |

**4. Lenguajes de programación y *stack* tecnológico**

| Cuadro de texto |
| --- |
| Un lenguaje de programación es aquel lenguaje computacional que sirve para crear *software* en general, sean programas de consola, de escritorio, orientados a la *web*, móviles, dispositivos embebidos; incluso, con un lenguaje de programación se pueden crear otros lenguajes de programación. La creación de este *software* se realiza mediante algoritmos o instrucciones dadas, propias de cada lenguaje (sintaxis) y una lógica. |

| Cuadro de texto |
| --- |
| Según su ejecución se pueden encontrar tres clases de lenguaje de programación: compilados, interpretados e híbridos (compilados e interpretados). Algunos autores solo consideran los compilados e interpretados. Imagen que contiene señal, firmar, reloj  Descripción generada automáticamenteImagen de la pantalla de un video juego  Descripción generada automáticamente con confianza baja  Los lenguajes de programación compilados son aquellos que necesitan de un traductor o compilador para ejecutarse, o sea, necesitan un compilador que convierte el código escrito en lenguaje, que es legible por el ser humano, a lenguaje máquina; ejemplo: C, C++.  Un lenguaje interpretado es aquel que se ejecuta sin que se genere código - objeto; por ejemplo: PHP, Python, JavaScript.  Por otro lado, los híbridos, son los que se encuentran en un estado entre compilado e interpretado; Java es uno de ellos, aunque algunos autores categorizan a Java como un lenguaje compilado.  Según su finalidad y nivel de abstracción los lenguajes de programación pueden ser de tipo máquina, *assemble*r, bajo nivel y alto nivel. |

| **Tipo de recurso** | Rutas / Pasos. Verticales 1 |
| --- | --- |
| **Introducción** | ¿Cómo funciona el proceso de compilación? |
| Capítulo 3: Componentes software. El sistema operativo - Página 2 de 7 -  SomeBooks.es  **Imagen:** 228125\_i25 | |
| **Botón 1** | Abre un programa IDE (Entorno integrado de desarrollo) y escribe su código en un lenguaje de programación de a**lto nivel,** el cual es legible y entendible por usted. |
| **Botón 2** | Compila su programa. Aquí ese código legible se convierte a código máquina, el código entendible por la CPU de su computadora: unos y ceros. |
| **Botón 3** | La máquina entiende ahora ese código, se puede ejecutar en ella y le muestra los resultados de su programa. |

| Cuadro de texto |
| --- |
| Existe un sinnúmero de lenguajes de programación, pero mencionaremos los más populares y demandados por la industria hoy día, aunque esto es relativo porque su uso varía con respecto a regiones geográficas. Por otra parte, hablando en términos de remuneración económica, unos son mejor pagos que otros. |

| Cuadro de texto |
| --- |
| Uno de los más populares es Python, y una de las razones principales de su popularidad es el crecimiento de la ciencia de datos en los últimos años; otras razones: la simplicidad, conjunto de librerías para distintos propósitos y facilidad de programación.  En la industria a nivel mundial se trabaja mucho con Java y Javascript, pero se trabaja fuertemente usando librerías y marcos de trabajo (*frameworks*), temas que se tomarán más adelante. PHP sigue siendo popular porque la mayoría de las páginas y aplicaciones web se crearon en este lenguaje y aún se siguen creando páginas con gestores de contenido como Wordpress, Joomla, Drupal, entre otros.  Otro lenguaje de programación que ha tomado fuerza en los últimos años es C y C++. Se puede afirmar que ha sido por su eficiencia con el *hardware* y el crecimiento del internet de las cosas: el desarrollo de videojuegos.  Existen otros lenguajes de programación tales como Ruby, Dart, Kotlin, Swift, C, TypeScript, Go, Rust, SQL (para bases de datos relacionales), Scala, R, Perl, Shell Script, etc.  En la siguiente tabla encontrará una lista de algunos lenguajes de programación y su uso, propósito o plataformas en las que puede construir *software* con ellos. |

**Tabla 2**

*Lenguajes de programación*

|  | **Tipo de aplicación o *software* (videojuegos, utilidades, empresarial)** | | | | |
| --- | --- | --- | --- | --- | --- |
| **Lenguaje de programación** | **Escritorio (videojuegos, utilidades, empresarial)** | **Web** | **Móvil** | **Embebido** | **Consola** |
| Java | X | X | X | X | X |
| PHP |  | X |  | X | X |
| Python | X | X | X |  | X |
| JavaScript | X | X | X |  | X |
| TypeScript | X | X | X |  | X |
| C | X |  |  | X | X |
| C++ |  |  |  | X | X |
| C# | X | X | X | X | X |

| Tipo de recurso | Cajón de texto de color |
| --- | --- |
| Hoy en día existen librerías y *frameworks* para hacer posible que, por ejemplo, lenguajes de programación como PHP construyan aplicaciones de escritorio y móvil, pero no es su propósito principal, y su *performance* puede no ser el mejor. | |

***Stack* tecnológico**

| Cuadro de texto |
| --- |
| Cómo diseñar un stack tecnológico para tu negocio  *Stack*, en español, traduce pila. Entonces, se puede mirar un *stack* tecnológico como una pila de tecnologías y cómo ellas se mezclan en conjunto como una sola para construir aplicaciones, según las necesidades que se tengan.  Antes de mencionar algunas miraremos qué son un *framework* (marco de trabajo) y una librería, para que pueda establecer la diferencia con un lenguaje de programación; estos tres términos suelen confundirse muy comúnmente cuando se está iniciando en este mundo. Existe un término más, “entorno de ejecución”, el cual es relativo y no tiene una definición exacta. |

| Cuadro de texto |
| --- |
| Con un lenguaje de programación se pueden crear “librerías”, esto normalmente surge cuando se están haciendo una o varias aplicaciones *software* distintas y se necesita resolver una tarea en común y evitar el trabajo de reescribir código una y otra vez para lo mismo; entonces se crea una librería y se “importa” o se usa cada vez que se necesite hacer dicha tarea.  Un *framework* es un conjunto de librerías ―patrones de diseño, patrones arquitectónicos― con el que se pueden trabajar, en muchas ocasiones, “buenas prácticas” de código y organización. En consecuencia, un *framework* puede contener varias librerías, y una librería no contiene *frameworks*.  A continuación se mencionan algunos *frameworks* y los lenguajes de programación que usan: Spring (Java), Laravel (PHP), Express (NodeJS/Javascript), Flask y Django (Python), Ionic (TypeScript, Javascript), Flutter (Dart). |

| **Tipo de recurso** | Acordeón tipo 1 |
| --- | --- |
| **Introducción** | Entre los *stack* más comunes en la actualidad se pueden encontrar: |
| Stacks de Desarrollo: Qué son y cómo elegirlos  **Imagen:** 228125\_i27 | |
| **MERN**  Estas son las siglas de MongoDB, Express.js, React JS, Node JS. MongoDB es un motor de base de datos no relacional. Express es un *framework* que usa el entorno de ejecución de Node JS que se usa para trabajar BackEnd. React JS es una librería que originalmente empezó usando el lenguaje de programación Javascript para construcción de interfaces gráficas de usuario (*FrontEnd*). | |
| **MEAN**  Son siglas de un *stack* tecnológico, al igual que MERN, pero a diferencia de este, la A se refiere al *framework* de Angular, que principalmente hace el uso del lenguaje de programación TypeScript. | |
| **LAMP**  Son las siglas de Linux, Apache, MySQL, PHP. Linux es un sistema operativo que serviría de plataforma para la instalación del servidor de aplicaciones Apache, y MySQL es un motor de base de datos. | |
| **.NET: e**s un ecosistema o *framework*, propiedad de Microsoft para la construcción de aplicaciones; usa lenguajes de programación como Visual basic o C#, entre otros. | |
| **Ruby On Rails**  Es un *framework* para construir aplicaciones *full stack* con el lenguaje de programación Ruby. | |

**5. Gestor de base de datos y servidores**

| Cuadro de texto |
| --- |
| Un ejemplo claro de base de datos es una biblioteca, allí se encuentra la información clasificada y organizada. Otro ejemplo son los libros contables en papel ―no sistematizados― de los pequeños negocios, incluso hojas de Excel donde alguien lleve sus cuentas.  Hoy en día, los sistemas de información de cualquier tamaño poseen una base de datos, así sea almacenada en archivos planos de texto. Anteriormente, en la mayoría de las aplicaciones persistían los datos en archivos o ficheros. Luego, se optó por usar motores de bases de datos por todas las ventajas que poseen, tales como: la velocidad de acceso a la información, el fácil mantenimiento, la consistencia de los datos, etc. En sus aplicaciones web será indispensable que cree bases de datos y las integre con estas. |

| **Tipo de recurso** | | Pestañas o tabs Verticales |
| --- | --- | --- |
| **Introducción** | | Es importante aclarar y saber diferenciar entre: |
| Gestión y desarrollo del CMS, concepto de administración de contenidos de sitios web. Administrador trabajando con base de datos, sistemas de tecnología de la información. Ilustración vectorial plana aislada en fondo blanco | | |
| **Gestor de base de datos** | Es aquel *software* especializado para realizar conexión con un motor de base de datos y todo tipo de operaciones: gestión de usuarios, bases de datos y sentencias con el motor como consultas, creación, borrados y actualizaciones que suelen ser de forma “grafica” o “consola”. Ejemplo de estos: DBEaver, MySQL Workbench, PgMyAdmin, PHPMyAdmin, SQL Server. | |
| **Sistema gestor de base de datos** | Se refiere a todo el conjunto que entra en juego con la base de datos: los usuarios que interactúan y las herramientas necesarias o *software* para trabajar con las bases de datos, motores, incluso el gestor de base de datos y la infraestructura asociada. | |
| **Motor de base de datos** | Es un *software* para administración de bases de datos. | |

| Tipo de recurso | Cajón de texto de color |
| --- | --- |
| En la actualidad, a nivel empresarial se trabaja con motores de bases de datos tanto relacionales como no relacionales. Los gestores de bases de datos (SGBD) más usados   * Relacionales: se pueden mencionar algunos como MySQL (es uno de los más populares), PostgreSQL, Oracle, Microsoft SQL Server. * No relacionales: entre ellos MongoDB, Cassandra, Redis.   El lenguaje de programación para los motores de bases de datos relacionales mencionados es SQL. Los no relacionales almacenan formato JSON o BSon. | |

| **Tipo de recurso** | Tarjetas Avatar | |
| --- | --- | --- |
| **Introducción** | SQL tiene 2 tipos de verbos-instrucciones: | |
| **DDL** (Lenguaje de definición de datos): CREATE: definición. | |  |
| **DML** (Lenguaje de manipulación de datos): SELECT, realizar consultas; INSERT, insertar registros; UPDATE, actualizar registros; DELETE, borrar registros. | |  |

| Cuadro de texto |
| --- |
| Entre las bases de datos relacionales y no relacionales (No-SQL) hay ciertas diferencias respecto a sus elementos, incluso sus propósitos. Pero se mencionan en la siguiente tabla algunas relacionadas con estructuras y usos. |

**Tabla 3**

*Diferencias bases de datos relacionales y no relacionales*

| **Tipo** | **SQL** | **NoSQL** |
| --- | --- | --- |
|  | Base datos | Base de datos |
| Tabla | Colección |
| Fila | Documento |
| Columna | Campo |
| Llave primaria | Llave primaria |
| **Usos** | Web, móvil, empresarial, *datamart.* | Videojuegos, redes sociales, IoT, web, móvil, empresarial. |

| Cuadro de texto |
| --- |
| Para ampliar la información, lo invitamos a leer en el material complementario, los siguientes artículos:   * Utilidad y funcionamiento de las bases de datos NoSQL * Un estudio comparativo de Bases de Datos Relaciónales y Bases de Datos NoSQL |

**Servidores**

| Cuadro de texto |
| --- |
| Los servidores son los que prestan el servicio requerido por un cliente. |

| **Tipo de recurso** | *Slider* presentación | |
| --- | --- | --- |
| **Introducción** | Algunos tipos de servidores son: | |
| **Servidores web**  Es un *software* especial para el procesamiento de peticiones en doble dirección cuando un cliente le solicita. También se le llama servidores HTTP, servidores que deben contar con un *hardware*. Hoy en día se habla de servidores web en la nube, pero al final estos también necesitan un *hardware* para funcionar.  En este tipo de servidores normalmente se alojan las páginas web que consulta día a día, y aplicaciones de compra, plataformas *streaming*, de música, etc., a las cuales accede mediante su navegador. | | Diagrama  Descripción generada automáticamente  **Imagen:** 228125\_i29 |
| **Servidores de FTP**  Las siglas FTP significan “Protocolo de transferencia de archivos”. Son servidores para el propósito de transferir archivos por internet. Funcionan cuando un cliente realiza la petición para enviar un archivo a ese servidor o se pide transferir dicho archivo desde ese servidor al cliente. Para realizar dicha transferencia debe contar con los respectivos permisos, sean: escritura, lectura o ambos. Un ejemplo de servidor FTP es Filezilla Server, que también tiene una versión cliente: Filezilla Client. | | Qué es el Protocolo de transferencia de archivos (FTP)?**Imagen:** 228125\_i30 |
| **Servidores de bases de datos**  Son aquellos en los que se alojan motores de bases de datos, relacionales o no relacionales. Incluso, pueden ser simples archivos que sean la base de datos de una aplicación. | | Network,database on green background,clean vector  **Imagen:** 228125\_i31 |

| Cuadro de texto |
| --- |
| Ampliemos la información sobre los servidores web.  Anteriormente se hablaba de “sistemas operativos para red”, los cuales se instalan en servidores físicos utilizados para distintos propósitos, entre ellos el de “servir” para la comunicación web. Se podría afirmar que al referirnos a un servidor web se está hablando de un servidor físico con un sistema operativo especial para red o un *software* como servidor, el cual se menciona a continuación:  Pero antes de hacer mención veamos qué diferencia hay entre un servidor web y un contenedor de aplicaciones. Un contenedor de aplicaciones es un *software* que puede alojarse dentro de un servidor web.  Apache Tomcat es un contenedor de aplicaciones de código abierto bastante usado para desplegar aplicaciones con Java en entorno web (https://tomcat.apache.org/). |

| **Tipo de recurso** | Tarjetas Avatar | |
| --- | --- | --- |
| **Introducción** | Algunos servidores web bastante demandados en el mercado son: | |
| **Apache**  Servidor HTTP de código abierto; se puede instalar sobre distintas plataformas como Linux, Window, MacOS. El puerto HTTP por defecto de Apache Server es el 80 y el puerto seguro es el 443 | | Descarga gratuita de Logotipo, La Fundación De Software Apache, Los Servidores De Un Ordenador Imágen de Png  **Imagen** 228125\_i32 |
| **Microsoft IIS**  Propiedad de Microsoft. Se pueden hacer despliegues de varios tipos de aplicaciones realizadas en PHP, JS, etc., pero es especial para aplicaciones construidas con el Framework .NET. | | Servidores de microsoft png imágenes | PNGWing  **Imagen:** 228125\_i33 |
| **Nginx**  Es un servidor web/proxy. Muchos han migrado a él porque han encontrado mejoramiento en su rendimiento. También es de código abierto. | | Descarga gratuita de  Imágen de Png  **Imagen:** 228125\_i34 |
| **GWS**  El gigante de Google también tiene su servidor, usado para su infraestructura concerniente a la web. | | Google logo google plataforma en la nube gboard google pay, 4, empresa,  Servicio, logo png | PNGWing  **Imagen:** 228125\_i35 |

**6. Protocolos de comunicación**

| Cuadro de texto |
| --- |
| Los protocolos son reglas que se deben seguir para tener un estándar de comunicaciones en las redes de datos y entre los clientes y los servidores.  Una hipotética situación en que cada uno cree su propia forma de comunicar a los clientes con los servidores se volvería un caos total, es cuando una autoridad llega y estandariza normas para que se cumplan en dichas comunicaciones. De esta manera habrá un número considerable y definido para cada tarea. |

| Tipo de recurso | Cajón de texto de color |
| --- | --- |
| “El servicio TCP se obtiene al hacer que tanto el servidor como el cliente creen puntos terminales, llamados sockets. Cada socket tiene un número (dirección), que consiste en la dirección IP del host, y un número de 16 bits, que es local a ese host, llamado **puerto** [sic]”. (Hernández, G., s.f.) | |

| **Tipo de recurso** | Acordeón tipo 1 |
| --- | --- |
| **Introducción** | Algunos tipos de puertos son: |
| Imagen general que ilustre el tema    **Imagen:** Construya aquí el código de la imagen | |
| **HTTP (*Hyper Text Transfer Protocol*)**  Protocolo de transferencia utilizada en el *World Wide Web*, especifica cuáles mensajes pueden enviar los clientes a los servidores y qué respuestas obtienen. HTTP soporta distintos métodos que se relacionan en la siguiente tabla:   | **Método** | **Uso / descripción** | | --- | --- | | *Get* | Petición de lectura. | | *Head* | Petición lectura de encabezado. | | *Put* | Petición almacenamiento para edición. | | *Patch* | Igual a *put*, pero para parte del recurso a editar. | | *Post* | Petición de guardado de recursos. | | *Delete* | Petición para eliminar recursos. | | *Trace* | Repite solicitud entrante. | | *Connect* | Reservado para uso futuro. | | *Options* | Consulta ciertas opciones. | | |
| **FTP (*File Transfer Protocol*)**  Los servidores FTP casualmente son los que trabajan con el protocolo de transferencia de archivos. El puerto de conexión por defecto de FTP es el 21. | |
| **XMPP (*Extensible Messaging and Presence Protocol*)**  Estándar abierto para mensajería y presencia. | |
| **SMTP (*Simple Mail Transfer Protocol*)**  En internet, el correo electrónico se entrega al hacer que la máquina de origen establezca una conexión TCP con el puerto 25 de la máquina de destino. Escuchando en este puerto está un demonio de correo electrónico que habla con el SMTP (Protocolo Simple de Transporte de Correo); este demonio acepta conexiones de entrada y copia mensajes de ellas a los buzones adecuados. Si no puede entregarse un mensaje, se devuelve al remitente un informe de error que contiene la primera parte del mensaje que no fue posible entregarse. | |

| Cuadro de texto |
| --- |
| Para ampliar la información consulte los siguientes enlaces:   * <https://developer.mozilla.org/es/docs/Web/HTTP/Methods> * <https://developer.mozilla.org/es/docs/Web/HTTP/Status> |

**Instalación herramientas de trabajo**

| Cuadro de texto |
| --- |
| Para empezar a desarrollar aplicaciones web, o de otro tipo, es necesario contar con la configuración de un ambiente de desarrollo a nivel de producción o local, y es importante contar con las distintas herramientas que se han mencionado en el presente componente formativo.  En el siguiente video se explicará cómo realizar la instalación de algunas de estas herramientas recomendadas de trabajo. |

| **Tipo de recurso** | Video clase / tutorial | | | |
| --- | --- | --- | --- | --- |
| **NOTA** |  | | | |
| **Título** | Instalación herramientas de trabajo | | | |
| **Escena** | **Imagen** | **Sonido** | **Narración** | **Texto** |
|  | Nota: son dos videos para unir, se encuentran en la carpeta Anexos con el nombre  Instalacion herramientas parte 1.mkv  Instalacion herramientas parte 2.mkv  Se debe grabar el audio y colocarlo en el video.  Si al final el video queda muy largo, se puede partir en 2.  El video tiene el audio del experto para que se guíe cuando se coloque el audio nuevo. |  | **Video parte 1**  En este video tutorial vamos a realizar la instalación de algunas herramientas recomendadas para el programa y que a futuro le servirán como profesional.  Son herramientas libres que se pueden descargar desde su web oficial. Tenemos **DBeaver**, herramienta que sirve para gestionar diferentes motores de bases de datos.  En la página oficial se encuentran dos versiones. La versión **DBeaver Community** y la versión **DBeaver PRO**.  Con la versión **Community** es suficiente para utilizarla y es una herramienta libre de uso.  Se descarga desde la página oficial, se selecciona el idioma, se instala, y al terminar la instalación, se busca en el computador.  Se puede iniciar conexión con distintos motores o gestores de bases de datos, todos los que se encuentran en el mercado, relacionales o no relacionales.  Para la configuración de uno de estos motores hay que tener en el equipo un motor instalado o también puede ser cualquier motor que se encuentre remoto, o sea, que se encuentre instalado en un servidor de bases de datos o, simplemente, un motor de bases de datos en la nube.  Otra herramienta es **Apache NetBeans.** Se busca en Google y en la página oficial aparecen varias versiones. La última es la **14**, pero se pueden descargar otras versiones anteriores. **NetBeans** es una herramienta de desarrollo y es posible programarla principalmente para Java. Pero no significa que no se pueda programar para otros lenguajes de programación.  **Visual Studio Code** es también una herramienta bastante útil**.** Se descarga según el sistema operativo que se tenga. Se puede trabajar con cualquier lenguaje de programación, incluso con JavaScript. Se recomienda trabajarla con lenguaje Front, pero con Java o con cualquier otro lenguaje también se trabaja esta herramienta. Visual Studio Code es interesante, recomendable y muy útil, además que su instalación es muy sencilla.  Vamos a realizarla. Tiene una serie de *plugins*. Se da un doble clic al instalador, se aceptan los términos y se instala. Se puede instalar en cualquier sistema operativo. Se instala y tenemos esta interfaz gráfica. Tiene una sección de *plugins* o extensiones y según el propósito o lenguaje de programación con el que se quiera trabajar, o cualquier *framework*, para formatear códigos, para muchas utilidades. Tiene un desarrollo web bastante ligero, rápido; en resumen, es muy bueno para cualquier propósito.  **Video parte 2**  Otra herramienta recomendada es **XAMPP**, pero solo trabaja en ambiente local.  No sirve para realizar aplicaciones en producción, aplicaciones reales; se pueden realizar, pero no es recomendable. Se recomienda para trabajar en el equipo local y realizar proyectos en ambiente local.  Es una herramienta que tiene integrado todo un ambiente de desarrollo con **Apache** y **MariaDB,** que es un motor alternativo a **MYSQL: PHP,** un lenguaje de programación, y **Perl**.  **XAMPP**, en resumen, instala un **STACK LAMP** que se vio en el curso.  Se descarga. Para configurar un ambiente LAMP, o sea Linux, Apache, MYSQL, PHP.  En este caso, nuestro sistema operativo es Windows, entonces se llamaría un WAMP.  XAMPP también está para Linux, sería un ambiente LAMP; y para Mac sería un ambiente MAMP.  Para realizar la configuración de un ambiente de estos hay que instalar herramienta por herramienta; o sea el motor de bases de datos MYSQL y el sistema operativo ya está. De manera independiente hay que instalar el intérprete PHP e instalar el servidor Apache.  Como esta herramienta nos instala todo, no tenemos que preocuparnos tanto por la configuración porque esta la hacemos nosotros. Simplemente ya tenemos el ambiente y podemos empezar a programar.  Esta herramienta tiene LAN, donde podemos poner el lenguaje; tiene MariaDB, PHP, pero no significa que solo sirva para realizar los proyectos con esas herramientas.  Es viable trabajar con JAVA y utilizar solamente el motor de base de datos.  Se pueden utilizar otros tipos de aplicaciones como JavaScript, una página que no tenga PHP y ser colocada en el servidor de Apache que se instaló con esta herramienta. Todas esas combinaciones son posibles.  No solo nos va a servir para trabajar Apache con PHP y MYSQL juntos, sino que se puede realizar otro tipo de proyectos con esta herramienta.  Por eso es muy importante y bastante útil instalarla, porque ahorra mucho trabajo en configuraciones para utilizar estos motores. Este lenguaje de programación, y ese servidor de Apache, un servidor bastante popular con PHP, es exclusivo para construcción de páginas web.  Este ambiente de desarrollo, ambiente de trabajo, sirve para trabajar más que todo aplicaciones web, pero el motor MYSQL , como lo instala también en todo este paquete, puede servir para realizar proyectos de bases de datos solamente.  Al momento de utilizar esta herramienta, y en el desarrollo de nuestros proyectos, se pueden realizar muchas tareas, en ambiente local, tareas con las que ya esta instalación se está realizando.  Aquí ya está terminando de instalar, pide permiso para el servidor de Apache. Le permitimos.  Normalmente el servidor de Apache, por defecto, se lee puerto HDTP, que es el puerto 80.  Aquí está instalada la herramienta, se puede iniciar si no arrancan los servicios, se clica permitir, inclusive hay un servidor FTP que es FileZilla; está también Tomcat.  Si no arrancan los servicios, no es posible entrar. Hay que iniciar *Local Host*, pero como no está encendido el servidor, no se va a encontrar nada, entonces, se enciende el servidor, que indica que está por el puerto 80 y por el puerto 443, el puerto seguro, que es el puerto por defecto.  Se encienden los puertos al dar Start. El puerto MYSQL es el 3306, por defecto.  Con este servidor se crea una conexión con Piper que sucederá más adelante. Por ahora, eso es todo. |  |
| **Nombre del archivo** | 228125\_v2 | | | |

| Cuadro de texto |
| --- |
| Recuerde explorar los demás recursos que se encuentran disponibles en este componente formativo; para ello, diríjase al menú principal, donde encontrará la síntesis, una actividad didáctica para reforzar los conceptos estudiados, material complementario, entre otros. |

**Síntesis**

| **Tipo de recurso** | Síntesis |
| --- | --- |
| Desarrollo de aplicaciones web *full stack*  Selección de la arquitectura de *software* | |
| **Introducción** | A continuación encontrará una síntesis de las temáticas trabajadas en el componente formativo. |
|  | |

**Actividad didáctica**

| **Tipo de recurso** | Actividad didáctica. Opción múltiple | |
| --- | --- | --- |
| Apreciado aprendiz, a continuación encontrará una serie de preguntas que deberá resolver, con el objetivo de evaluar la aprehensión de los conocimientos expuestos en este componente formativo. | | Concepto de diseño de código de desarrollo web HTML  **Imagen 1:** 228125\_i37 |
| ¿Cuáles son los tres componentes básicos de una red de comunicaciones? | |  |
| Medios de comunicación, servicios y dispositivos (correcta) | | Internet, servicios, *router* |
| *Switch, router,* cables | | Ninguno de los anteriores |
| **Retroalimentación positiva:**  Muy bien, estos son los tres componentes básicos.  **Retroalimentación negativa:**  ¡Incorrecto! Es necesario revisar nuevamente el tema sobre *Networking.* | | |
| Las principales capas de un sistema operativo son: *kernel* o núcleo, servicios e intérprete de comandos. | |  |
| Verdadero (correcta) | | Falso |
| **Retroalimentación positiva:**  Muy bien, ha acertado.  **Retroalimentación negativa:**  ¡Incorrecto! Es necesario revisar nuevamente el tema sobre sistemas operativos. | | |
| Son proveedores de servicios c*loud* comerciales. | |  |
| AWS, Google Cloud, Azure. (correcta) | | Azure, Tomcat, Kernel. |
| AWS, CPU, Arpanet. | | AWS, Azure, FTP. |
| **Retroalimentación positiva:**  Muy bien, ha acertado al elegir los proveedores de servicios c*loud* comerciales.  **Retroalimentación negativa:**  ¡Incorrecto! Es necesario revisar nuevamente el tema sobre infraestructura. | | |
| La capa NFT es una capa de la arquitectura de *software* por capas. | |  |
| Verdadero | | Falso (correcta) |
| **Retroalimentación positiva:**  Muy bien, ha acertado.  **Retroalimentación negativa:**  ¡Incorrecto! Es necesario revisar nuevamente el tema sobre las capas básicas. | | |
| Con el lenguaje de programación Java puede crear: | |  |
| Aplicaciones web | | Aplicaciones de escritorio |
| Aplicaciones móviles | | Todas las anteriores (correcta) |
| **Retroalimentación positiva:**  Muy bien, ha acertado. Con Java se pueden crear aplicaciones de escritorio, web y para móviles.  **Retroalimentación negativa:**  ¡Incorrecto! Es necesario revisar nuevamente el tema sobre lenguajes de programación. | | |
| Ruby on Rails es un: | |  |
| Lenguaje de programación. | | Librería |
| Base de datos | | *Framework* (correcta) |
| **Retroalimentación positiva:**  Muy bien, ha acertado.  **Retroalimentación negativa:**  ¡Incorrecto! Es necesario revisar nuevamente el tema sobre *stack* tecnológico. | | |
| MySQL es considerado un motor de bases de datos no relacional: | |  |
| Verdadero (correcta) | | Falso |
| **Retroalimentación positiva:**  Muy bien, ha acertado.  **Retroalimentación negativa:**  ¡Incorrecto! Es necesario revisar nuevamente el tema sobre gestor de base de datos. | | |
| Son servidores para la transferencia de archivos. | |  |
| FTP (correcta) | | Web o HTTP |
| Bases de datos | | FTTP |
| **Retroalimentación positiva:**  Muy bien, ha acertado.  **Retroalimentación negativa:**  ¡Incorrecto! Es necesario revisar nuevamente el tema sobre servidores. | | |
| Apache Tomcat es catalogado como un servidor web*.* | |  |
| Verdadero | | Falso (correcta) |
| **Retroalimentación positiva:**  Muy bien, ha acertado.  **Retroalimentación negativa:**  ¡Incorrecto! Es necesario revisar nuevamente el tema sobre servidores web. | | |
| Es un verbo HTTP para realizar peticiones: | |  |
| GET (correcta) | | POSTTT |
| FTP | | SMTP |
| **Retroalimentación positiva:**  Muy bien, ha acertado.  **Retroalimentación negativa:**  ¡Incorrecto! Es necesario revisar nuevamente el tema sobre protocolos de comunicación. | | |

Retroalimentación general positiva:

¡Felicitaciones! Ha logrado una óptima aprehensión de los conocimientos relacionados con la selección de la arquitectura de *software*

Retroalimentación general negativa:

¡Inténtelo de nuevo! Lo invitamos a revisar nuevamente el material de estudio para afianzar los conocimientos presentados. ¡Ánimo!

**Material complementario**

| Tipo de recurso | Material complementario | | |
| --- | --- | --- | --- |
| Tema | Referencia APA del material | tipo | Enlace |
| Modelos de arquitectura | hdeleon.net. (2020). *Modelo de Arquitectura de Software C4 – Ejemplo práctico sencillo* (video). YouTube. <https://www.youtube.com/watch?v=2IJ29e0xTDc> | Video | <https://www.youtube.com/watch?v=2IJ29e0xTDc> |
| Lenguajes de programación y *stack* tecnológico | nicobytes. (2020). *¿Cuál es el mejor stack tecnológico para una aplicación?* (video). YouTube. <https://www.youtube.com/watch?v=16CauMlYygs> | Video | <https://www.youtube.com/watch?v=16CauMlYygs> |
| Gestor de base de datos y servidores | Castro Romero, A., González Sanabria, J. S. & Callejas Cuervo, M. (2012). Utilidad y funcionamiento de las bases de datos NoSQL. *Revista Facultad de Ingeniería*, 21(33). p. 21-32  <https://www.redalyc.org/articulo.oa?id=413940772003> | PDF | <https://www.redalyc.org/articulo.oa?id=413940772003> |
| Gestor de base de datos y servidores | Moreno, L., Olsowy, V., Thomas, P., Delia, L., Tesone, F., Fernández, J. & Pesado, P. (2019). *Un estudio comparativo de bases de datos relaciónales y bases de datos NoSQL.*  <http://sedici.unlp.edu.ar/handle/10915/91403> | PDF | <http://sedici.unlp.edu.ar/handle/10915/91403> |

**Glosario**

| **Tipo de recurso** | Glosario |
| --- | --- |
| Ciclo de vida del desarrollo de *software*: | estructura que contiene los procesos, actividades y tareas relacionadas con el desarrollo y mantenimiento de un producto de *software*, que abarca la vida completa del sistema, desde la definición de los requisitos, hasta la finalización de su uso. (https://ungoti.com/es/soluciones/desarrollo-de-software/sdlc/#:~:text=El%20ciclo%20de%20vida%20del,definici%C3%B3n%20de%20los%20requisitos%20hasta) |
| Lenguaje de programación: | es aquel lenguaje computacional que sirve para crear *software* en general, sean programas de consola, de escritorio, orientados a la *web*, móviles, dispositivos embebidos. |
| Medios de comunicación: | son los canales por donde viaja la información como cables UTP, de fibra óptica o wifi. |
| *Networking:* | redes de datos para comunicación e intercambio de información entre dispositivos o terminales. |
| Patrón de diseño: | soluciones habituales a problemas comunes en el diseño de *software*. Cada patrón es como un plano que se puede personalizar para resolver un problema de diseño particular de un código. (https://refactoring.guru/es/design-patterns) |
| Servicios: | programas y aplicaciones que están a lo largo de la red de comunicaciones. |
| Topología de redes: | forma como se interconectan los dispositivos de una red. |

**Referencias bibliográficas:**

| **Tipo de recurso** | Bibliografía |
| --- | --- |
| Developer Mozilla. (s.f.). *Códigos de estado de respuesta HTTP.* <https://developer.mozilla.org/es/docs/Web/HTTP/Status> | |
| Garlan, D. (2001). *Sofware Architecture.* <http://www.cs.cmu.edu/afs/cs.cmu.edu/project/able/ftp/encycSE2001/encyclopedia-dist.pdf> | |
| GNU. 2022. Recuperado de: <https://www.gnu.org/home.es.html> | |
| Hernández, G. (s.f.). *Protocolo de control de transferencia (TCP).* Universidad Autónoma del Estado de Hidalgo. <https://www.uaeh.edu.mx/scige/boletin/huejutla/n3/r1.html#refe1> | |