

Componente formativo

**Redes, sistemas de desarrollo y lenguajes de programación.**

**Breve descripción:**

Este componente formativo está diseñado para comprender los conceptos de entornos de desarrollo de programación, redes de datos, manejo de archivos *JavaScript* y los diferentes repositorios de códigos tales como *GitHub* y *git*.

**Área ocupacional:**

Ciencias naturales, aplicadas y relacionadas

**Noviembre 2022**

Tabla de contenido

[Introducción 3](#_Toc129275344)

[1. Redes 4](#_Toc129275345)

[2. Sistemas operativos 10](#_Toc129275346)

[3. Tecnologías de versionamiento 14](#_Toc129275347)

[4. Instalación y configuración de servidores 17](#_Toc129275348)

[5. Manejo de herramientas de desarrollo de *software*: *IDEs* 23](#_Toc129275349)

[6. Estrategia *DevOps* 25](#_Toc129275350)

[7. Técnicas de codificación y manejo de los servicios en la nube 29](#_Toc129275351)

[Síntesis 33](#_Toc129275352)

[Material complementario 34](#_Toc129275353)

[Glosario 35](#_Toc129275354)

[Referencias bibliográficas 37](#_Toc129275355)

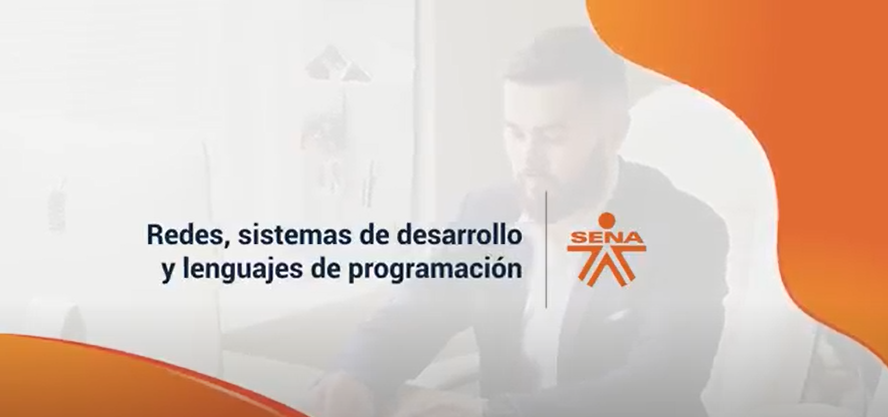
[Créditos 38](#_Toc129275356)

Introducción

Apreciado aprendiz, bienvenido a este componente formativo, donde abordaremos las temáticas referentes a la configuración de redes y los diferentes servicios que se utilizan a nivel de despliegue para las diferentes aplicaciones que se desarrollan en una empresa. Se estudiará el uso de repositorios de código a nivel mundial, la configuración de servidores para el manejo y despliegue de aplicaciones y bases de datos como servicios y componentes funcionales dentro de un proyecto de *software*. Además, como aprendizaje final, la configuración y despliegue de servicios en la nube.

En el siguiente video conocerá, de forma general, la temática que se estudiará a lo largo del componente formativo.

1. Redes, sistemas de desarrollo y lenguajes de programación



[**Enlace de reproducción del video**](https://www.youtube.com/watch?v=5M8llQQUEtw)

|  |
| --- |
| **Síntesis del video: Redes, sistemas de desarrollo y lenguajes de programación** |
| Hoy en día las herramientas y técnicas de desarrollo de programas informáticos juegan un papel fundamental en la calidad y funcionamiento de las aplicaciones, para garantizar su proceso de adaptabilidad y escalabilidad en los servicios que requiere la empresa. Es importante tener en cuenta que, de acuerdo con el tipo de aplicativo a desarrollar y las necesidades de la empresa, se requiere el uso e implementación de instrumentos para el despliegue y puesta en marcha de proyectos de *software*.  Muchas de estas asistencias se ofrecen a través de computación en la nube, que consiste en ofrecer a las empresas servicios de alojamiento y despliegue de su infraestructura tecnológica, permitiendo que esta pueda ser controlada desde cualquier parte del mundo, por un cómodo costo mensual o anual con una suscripción.  En la actualidad, no solo se requiere que la construcción de sistemas informáticos se realice de manera rápida, eficiente y con calidad, sino que se construya de la mejor manera. Es importante que el proceso sea controlado.  Las redes de computadoras han hecho posible que el uso del software y aplicaciones en general, puedan ser utilizadas de manera masiva por millones de usuarios y computadora nivel mundial.  Bienvenido a este maravilloso mundo. Lo invitamos a estudiar a profundidad cada uno de estos temas en el presente componente formativo. |

# Redes

Comencemos hablando de las redes de datos, las cuales surgen, inicialmente, como una necesidad para procesos de estrategia militar, donde era necesario transportar información de manera rápida y eficiente, aunque en sus inicios, el proceso de la seguridad de la información no se consideraba como una necesidad prioritaria, pero a lo largo de la historia fue tomando mucha fuerza debido a la confidencialidad de la información que estas manejan. Hoy en día, las redes se han convertido en uno de los servicios que hacen posible el transporte de información y datos a nivel mundial, y es utilizado por millones de usuarios todos los días; cada vez que abrimos nuestro correo electrónico o publicamos en nuestras redes sociales, estamos utilizando los servicios de la red más grande del mundo llamada internet, la cual hace posible la comunicación a nivel mundial.

Ahora, la configuración de una red depende de su tipología y las necesidades que se requieran para la comunicación de la información. Existen redes inalámbricas y redes cableadas y, para estos casos particulares, su configuración puede variar, de acuerdo con la implementación.

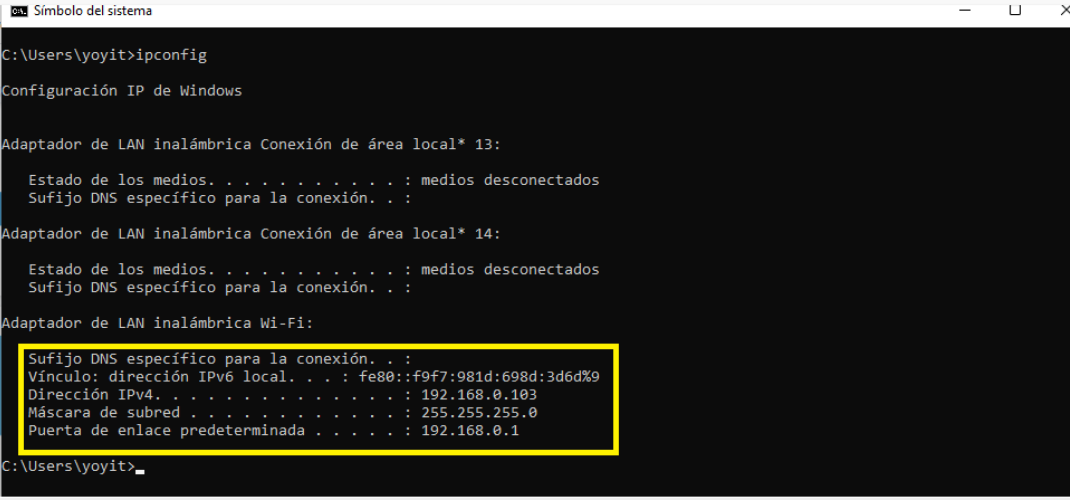
1. **Redes cableadas:** Utilizan diferentes tipos de filamentos en los cables, para el transporte de la información.

Como podemos observar en la imagen, la conexión que se da entre cada dispositivo que pertenece a la red, se lleva a cabo a través de un cable conductor, el cual transporta los datos y la información que puede ser compartida con otros dispositivos. Es una de las redes más utilizadas, porque ofrece como ventaja, la seguridad, estabilidad y velocidad en el transporte de la información; estas requieren, para su configuración, dispositivos tales como routers, los cuales se encargan de darle transporte a la información y las solicitudes realizadas por el cliente.

1. **Redes inalámbricas:** Las redes inalámbricas o más conocidas como WIFI, no utilizan un medio físico como las redes cableadas, sino que hacen uso del aire para la transmisión de información. Estas redes se han vuelto populares en muchos lugares del mundo puesto que su mantenimiento, uso e implementación son mucho más sencillos que el de la red cableada teniendo una ventaja, no requieren que el dispositivo esté conectado físicamente a un cable o medio para interactuar con los servicios requeridos, pero debemos tener en cuenta que este tipo de redes pueden llegar a ser vulnerables sino se realiza una correcta instalación o implementación, donde se utiliza una clave compartida en el dispositivo que despliega la señal de la red y luego esa misma clave se debe colocar en los dispositivos que requieren conectarse al mismo para realizar un proceso de autenticación y así poder utilizar los servicios de la red de manera adecuada y segura.

A continuación, veremos las diferentes configuraciones que se hacen en cada una de estas redes; esto implica utilizar una nomenclatura llamada dirección IP, formada por un número que se le asigna a cada equipo, para poder ser identificado dentro de la red de datos; así como en una ciudad existen nomenclaturas para las casas y de esa manera identificar a cada una de ellas, en las redes de datos, se utiliza la dirección IP para poder encontrar cada dispositivo en la misma. La siguiente imagen presenta un ejemplo de dirección IP.

1. Ejemplo dirección IP



Enmarcada con color amarillo, se muestra la configuración de la dirección IP de un equipo en la red; cada número que se refleja en la misma, indica el tipo de red, número del dispositivo y segmento donde se encuentra; adicional a esto, la máscara de red y puerta de enlace de comunicación.

**Roles y usuarios**

Es importante identificar qué es un rol y qué es un usuario.

Conozcamos cada uno de ellos.

1. **Usuario**: Usuario es el nombre que recibe el individuo que utiliza, de manera habitual, un producto o un servicio y, también se identifica como usuario, al que un sistema reconoce, para que pueda utilizarlo; es decir, existe una separación entre la persona que utiliza un sistema y el nombre o usuario que se utiliza, para identificarse en un sistema y poder acceder a él.

Un ejemplo de esto se presenta en los servidores de videojuegos; en ellos, la persona tiene un nombre de pila para poder ser identificado por los otros usuarios del sistema; por ejemplo: Dulfran Montaño es el usuario: yoyito1984. Este es un claro ejemplo del uso de esta característica para la identificación de una persona, dentro de un conjunto de usuarios.

Los usuarios son los que reciben los roles y también las restricciones para el ingreso al sistema; es decir, no se bloquea a quien está sentado en la computadora, sino al usuario o credenciales que este tiene para ingresar al sistema, como tal; esto en teoría, se considera algo muy sencillo, pero en la práctica suele ser un poco más complejo de lo que parece y, por esta razón, es importante que antes de la creación de un sistema de información, determinar cuáles son los mecanismos que se utilizarán para que los diferentes usuarios puedan utilizar los servicios ofrecidos por la aplicación misma.

1. **Rol**: Los usuarios, para conectarse a un sistema de información, representan un estatus y las funciones que pueden realizar dentro de ese sistema; el rol representa la distinción de cada persona dentro del sistema. Dentro de los roles, podemos destacar los siguientes:

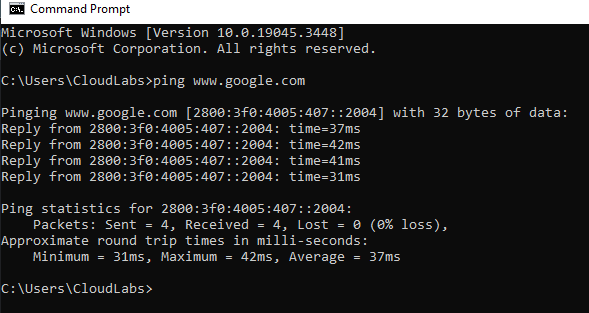
* Usuario registrado: se refiere a las personas que ingresan su información a un sistema para utilizar sus funciones, almacenar información y procesar la misma, de alguna manera; en algunos casos, como en las redes sociales o servidores de correo electrónico, son usados para diferentes funciones.
* Administrador o gestor de información: se encarga del mantenimiento y gestión de la información; sus funciones van desde bloquear usuarios, a realizar copias de seguridad de los sistemas de almacenamiento, entre otros; adicional a esto, se puede decir que los roles permiten también determinar a qué lugares del sistema tienen acceso los usuarios; por ejemplo, si un usuario tiene acceso solo a los módulos del sistema, se le conceden los permisos, pero el usuario administrador, regularmente, tiene acceso a todas las funciones del sistema, incluyendo las que son de configuración.

**Comandos de red**

Son instrucciones que permiten controlar y verificar la información que se tiene, como configuraciones o estados de la red que se ha implementado. Existen muchos comandos de red, pero, en este componente formativo, abordaremos aquellos que permiten determinar el estado de la misma, si existe comunicación entre los diferentes dispositivos que se encuentran instalados en la red, y se realizará una verificación de la conectividad y qué tan estable se encuentra la misma.

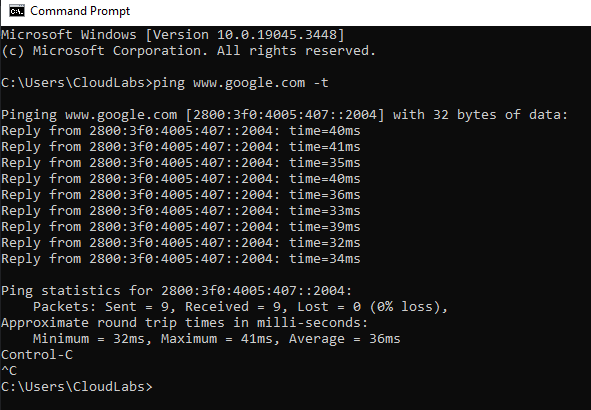
Por ejemplo, el comando *ping* se utiliza para verificar si existe conectividad de extremo a extremo entre dos computadoras en una red de datos; el comando realiza 4 verificaciones, las cuales deben ser comprobadas de manera correcta por el sistema, retornando como respuesta, paquetes enviados = 4, recibidos = 4 y perdidos = 0; esto indica que no se registró ninguna pérdida de información y que la conexión de los dos dispositivos es estable y no tiene interrupciones; pero si en algún momento se detecta que los paquetes recibidos disminuyen, se puede llegar a la conclusión que existen problemas de conexión en los dos extremos de la conexión. En algunas ocasiones esto se debe a una mala ponchada de los cables que conforman el conector *RJ45* que se conecta al dispositivo o computadora y también se presenta cuando se invierten algunos cables en el orden normal, existen otras posibilidades como deficiencia en la señal en caso de las redes inalámbricas o interrupción en la comunicación de los mismos. La siguiente imagen nos muestra un comando *ping*.

1. Comando ping



Y un comando ping – t es un derivado del comando anterior, con la particularidad que este realiza un conjunto de más de 4 peticiones para realizar una estadística de las respuestas y los tiempos que le toma al dispositivo entregar las diferentes respuestas a las solicitudes que hace a los diferentes nodos o equipos que se encuentran en la red; esto se realiza para obtener información detallada de la interacción de este equipo con los otros en la red. La siguiente imagen nos presenta un ejemplo de este comando.

1. Comando ping – t



En la siguiente tabla veremos los principales comandos utilizados y su especificación.

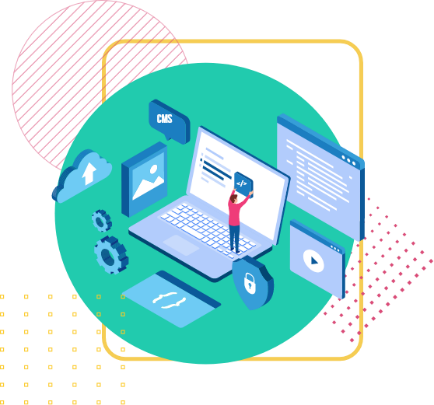
1. Comandos de red

| Comando | Especificación |
| --- | --- |
| -a | Resolver direcciones en nombres de host. |
| -n count | Número de solicitudes de eco para enviar. |
| -l size | Enviar tamaño de búfer. |
| -f | Establecer marca, no fragmentar en paquetes (solo IPv4). |
| -v TOS | Tipo de servicio (solo IPv4. Esta opción está desusada y no tiene ningún efecto sobre el campo de tipo de servicio del encabezado IP). |
| -r count | Registrar la ruta de saltos de cuenta (solo IPv4). |
| -j host-list | Ruta de origen no estricta para lista-host (solo IPv4). |
| -4 | Forzar el uso de IPv4. |
| -6 | Forzar el uso de IPv6. |

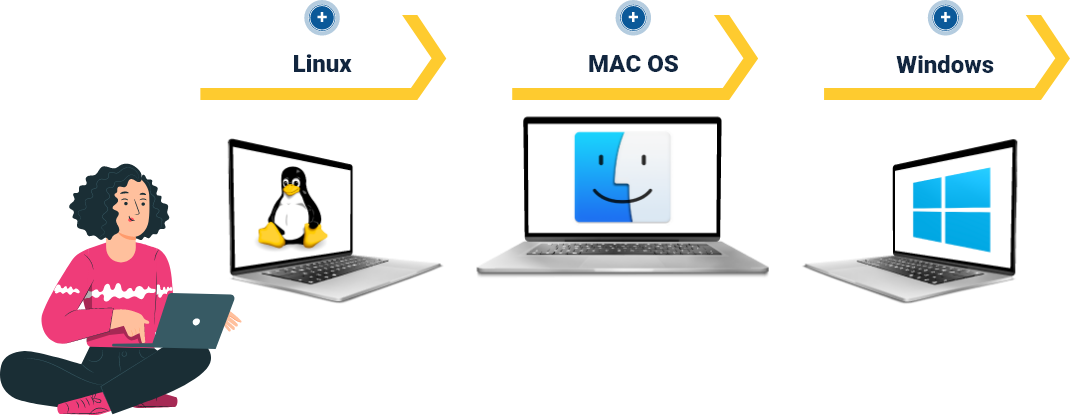
# 

# Sistemas operativos

Con respecto a los sistemas operativos, son un tipo de *software* que permite que los periféricos y el *hardware* de la computadora pueda recibir las órdenes que el usuario le entrega al computador, es un *software* que se encarga de controlar todas las funciones que realiza el usuario a través de la computadora y que estas se cumplan de acuerdo a lo solicitado, otra de las funciones que tiene el sistema operativo es realizar la comunicación entre los diferentes equipos, esto es a lo que llamamos protocolos de comunicación, los cuales permiten crear las reglas de comunicación entre los computadores que se encuentran conectados a una red informática, sea para compartir información o archivos en general. La gestión del sistema operativo es fundamental para el correcto funcionamiento de todos los otros programas instalados en una computadora, porque cuando hablamos de la ejecución de un programa nos estamos refiriendo a la asignación de memoria RAM para que estos funcionen y es el sistema operativo el encargado de asignar los espacios de memoria RAM requeridos para la operación de dicho *software*, por esta razón toda computadora debe contar con un sistema operativo para su operación y puesta en marcha.



Existen varios sistemas operativos, pero mencionaremos los más utilizados actualmente.



1. **Linux**: Es un sistema operativo desarrollado por la empresa Microsoft, el cual se ha convertido a lo largo de los años en el principal sistema operativo utilizado en el mundo, por su sencillez y facilidad. Adicional a ello, tiene una gran compatibilidad con los diferentes dispositivos y fabricantes actuales, para el tema de videojuegos e integración de multimedia; es la mejor alternativa para diseñadores y programadores de computadoras, porque es un sistema operativo muy versátil y utilizado.
2. **MAC OS**: Es un sistema operativo desarrollado por la empresa Apple, el cual permite realizar el control y administración de los equipos de dicha marca. Este sistema operativo es muy versátil y utilizado por muchos usuarios a nivel mundial, por su gran capacidad de adaptación al usuario final y su seguridad; es importante mencionar que este sistema operativo se distribuye también para dispositivos móviles tales como tabletas y teléfonos celulares.
3. **Windows**: Es un sistema desarrollado bajo la filosofía de software libre, el cual tiene un concepto de manejo de libertades en el uso del *software*; este sistema operativo está más relacionado con los equipos que prestan servicios llamados servidores, los cuales permiten instalar programas que pueden ser utilizados por millones de usuarios en Internet.

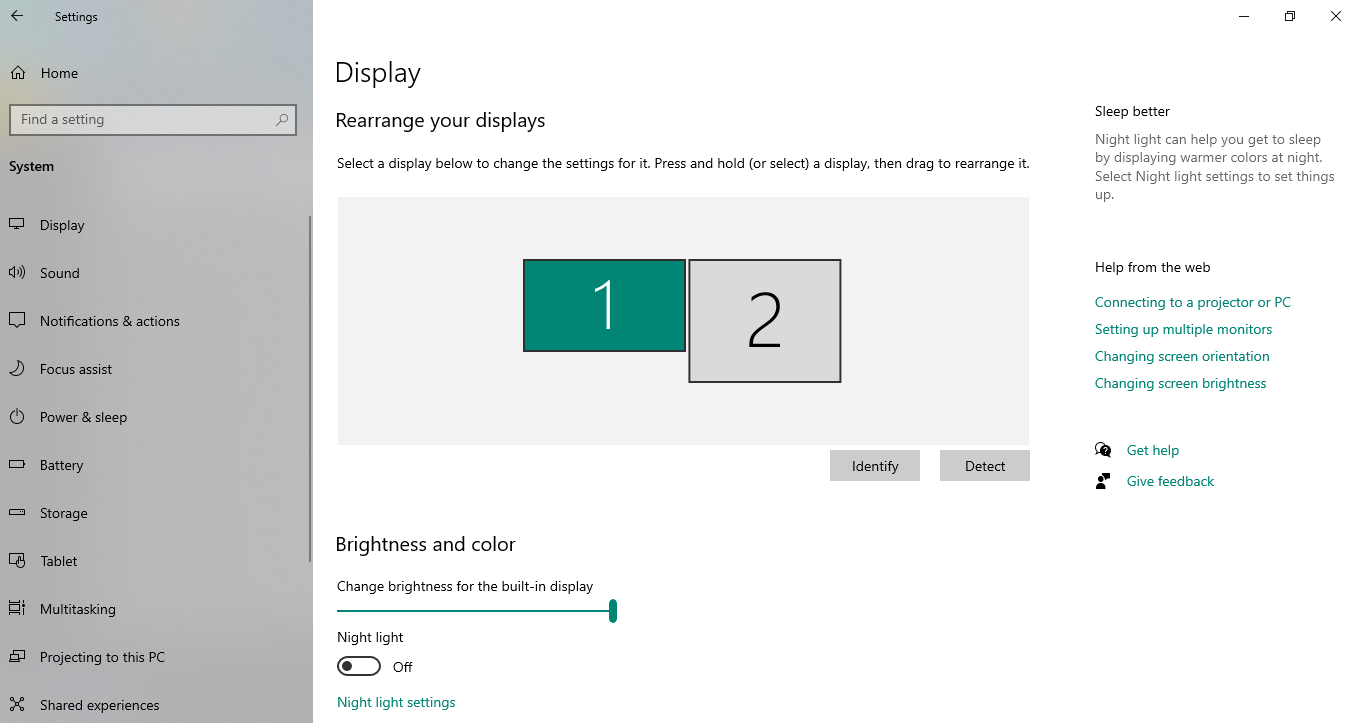
Ahora, la configuración del sistema operativo inició desde el proceso de instalación del mismo, lo cual integra diferentes fases, las cuales inician desde el proceso de preparación del dispositivo de instalación hasta la instalación de los archivos de ejecución del mismo, ahora bien, parte de la configuración también consiste en verificar ciertos puntos fundamentales del sistema operativo para poder instalar o colocar en marcha ciertos programas que son fundamentales para el correcto funcionamiento de la computadora.

Uno de los puntos importantes de la configuración del sistema operativo, se fundamenta en los periféricos principales; por ejemplo, tarjeta de red, teclado y dimensiones de la pantalla, debido a que estos dispositivos son fundamentales para que el usuario del sistema pueda acceder a los recursos de la red tales como internet. La configuración de la tarjeta de red, debe tener como finalidad la conectividad correcta del equipo, como vimos anteriormente.

A continuación, veremos algunas configuraciones de red, pantalla y teclado para el ejemplo nos enfocaremos en el sistema operativo Windows, porque es el más utilizado, como mencionamos anteriormente.

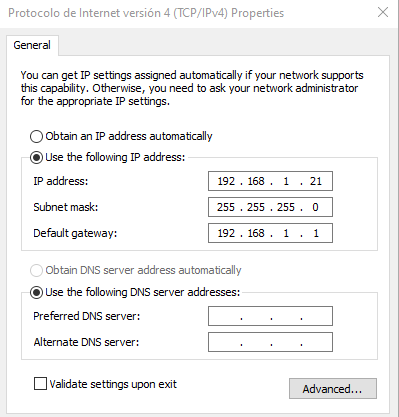
En la imagen podemos observar la configuración de la pantalla; contiene varias configuraciones, las cuales permiten al usuario, hacer un uso adecuado de los elementos que se tienen en la pantalla; adicional a esto, se puede controlar el brillo de la pantalla, la resolución y el tamaño de los objetos.

1. Configuración de la pantalla



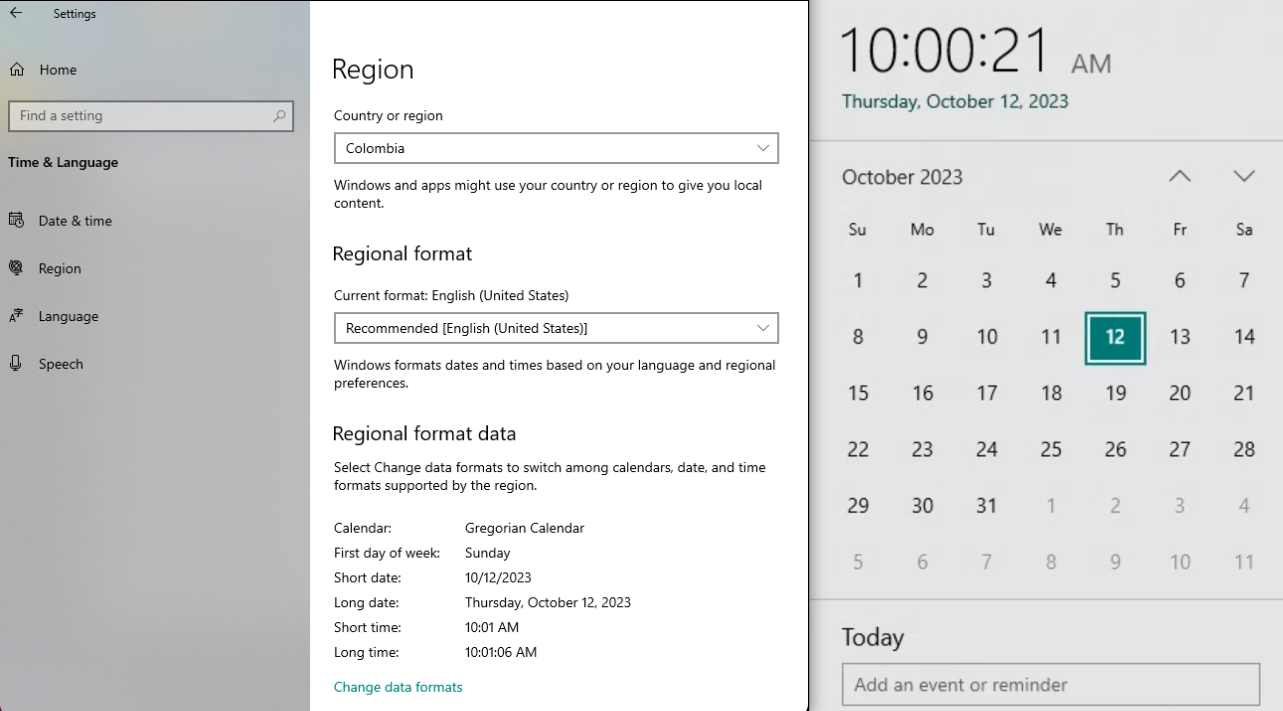
Continuemos con la configuración de la tarjeta de red, la cual es fundamental y hace parte de la configuración del sistema operativo. Permite que un dispositivo pueda ser configurado para acceder a ciertos recursos de red, en el caso de los servidores estos requieren utilizar muchas veces VPN que son redes virtuales privadas para el manejo de la seguridad y conexión de estos, todo esto se realiza a través de la configuración de red, por ello es importante tener en cuenta este proceso. La siguiente imagen nos presenta la configuración de la tarjeta de red.

1. Configuración de tarjeta de red.



En esta imagen, observamos cómo se establece la configuración del teclado, idioma, formato de hora y fecha del sistema. Este proceso es muy importante, debido a que muchas aplicaciones informáticas requieren que el formato de hora y fecha tenga unas características especiales y, en este espacio, se puede realizar la configuración del mismo; adicional a esto, se pueden configurar las preferencias del idioma que el usuario quiera utilizar, lo cual puede ser muy importante en un momento indicado.

1. Configuración de hora y fecha.



Para finalizar, lo invitamos a ver el documento Sistemas operativos y sus comandos, donde encontrará los diferentes comandos que se utilizan en los diferentes sistemas operativos.

Anexo: [Sistemas operativos y sus comandos](ANEXO_Sistemas_operativos_y_sus_comandos.pdf)

# Tecnologías de versionamiento

Ahora veamos las tecnologías de versionamiento, estas son una serie de software que permite el control del código fuente de aplicaciones, las cuales mantienen un control sobre los cambios que se realizan en las mismas. Estas tecnologías son muy utilizadas para el control de proyectos de software, en muchas ocasiones las versiones del software y la codificación se realizan en la computadora del desarrollador, lo cual puede ponerel código elaborado en riesgo de seguridad y también en pérdida del mismo. Cuando se utilizan herramientas de versionamiento y repositorio de código, estos quedan almacenados en la nube evitando que se pueda perder información, permitiendo que todos los miembros del equipo puedan realizar modificaciones y trabajar de manera colaborativa, siendo esto una ventaja competitiva para todos los equipos de desarrollo que utilizan estos servicios a nivel mundial. Otro servicio que ofrecen es el versionamiento, este permite la verificación de las versiones del software que se está desarrollando y así poder tener un control mayor sobre las modificaciones que sufre el software en cada versión.

Conozcamos los sistemas de control de versiones más utilizados:

1. **GIT**: Git es, con diferencia, el sistema de control de versiones moderno más utilizado del mundo. Es un proyecto de código abierto maduro y con un mantenimiento activo que desarrolló originalmente Linus Torvalds, el famoso creador del kernel del sistema operativo Linux en el 2005. Este sistema funciona a la perfección en una amplia variedad de sistemas operativos y en IDE (Entorno de Desarrollo Integrado).

L as características básicas del rendimiento de Git, son muy sólidas en comparación con muchas otras alternativas. Los algoritmos implementados en Git aprovechan el profundo conocimiento sobre los atributos comunes de los auténticos árboles de archivos de código fuente, cómo suelen modificarse con el paso del tiempo y cuáles son los patrones de acceso. Su arquitectura distribuida, también permite disfrutar de importantes ventajas en términos de rendimiento.

1. **GITHUB**: GitHub es un sitio “social coding”. Permite subir repositorios de código para almacenarlo en el sistema de control de versiones Git. El sistema es código abierto por defecto, lo que significa que cualquiera en el mundo puede encontrar su código en GitHub, usarlo, aprender de él, y mejorarlo.

Es considerado hasta el momento como uno de los mejores gestores de código fuente del mundo, debido a que permite realizar actualizaciones en tiempo real del mismo y también controlar mediante permisos qué personas pueden realizar cambios en el código fuente.

GitHub es una forja para alojar proyectos, utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas de ordenador. El software que opera GitHub fue escrito en *Ruby on Rails.*

1. **GITLAB**: GitLab es una *suite* completa que permite gestionar, administrar, crear y conectar los repositorios con diferentes aplicaciones y hacer todo tipo de integraciones con ellas, ofreciendo un ambiente y una plataforma en la cual se pueden realizar varias etapas de SDLC/ADLC y DevOps.

Esta *suite*, gestiona cualquier tipo de repositorio de códigos y establece las integraciones de los diferentes códigos que esta pueda contener, en otras palabras, podríamos decir que GITLAB es una plataforma que contiene un ambiente para integrar varios códigos de proyectos al mismo tiempo, esto con los desafíos que tienen las infraestructuras actuales en desarrollo de *software*, ofrece una gran ventaja porque en los equipos de desarrollo no es necesario que implementen varias herramientas sino que tienen todo lo que requieren en un mismo ambiente de trabajo, adicional a esto, se pueden manejar todas las etapas del desarrollo de un proyecto de *software* desde la creación del proyecto, codificación y despliegue de la misma.

A diferencia de algunos programas de software para control de versiones, Git no se deja engañar por los nombres de los archivos a la hora de determinar cuál debería ser el almacenamiento y el historial de versiones del árbol de archivos; en lugar de esto, se centra en el contenido del propio archivo. En definitiva, los archivos de código fuente cambian de nombre, se dividen y se reorganizan con frecuencia. El formato de objeto de los archivos del repositorio de Git, emplea una combinación de codificación delta (que almacena las diferencias de contenido) y compresión, guardando explícitamente el contenido de los directorios y los objetos de metadatos de las versiones.

Siguiendo con el tema de las tecnologías, el siguiente video nos brinda una explicación sobre las herramientas de control de código.

1. Herramientas de control de código.



[**Enlace de reproducción del video**](https://www.youtube.com/watch?v=J73jR7mQfoE)

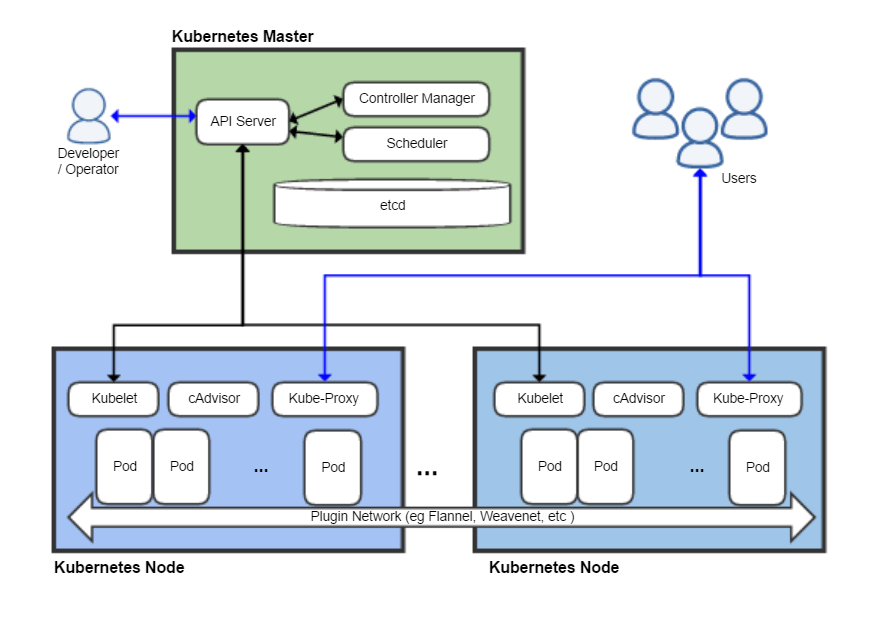
|  |
| --- |
| **Síntesis del video: Herramientas de control de código.** |
| Las herramientas de control de código permiten llevar, en tiempo real, los cambios que se han realizado a un sistema en particular y quién ha sido el responsable de realizar este proceso. Se puede compartir el proyecto con los involucrados en un documento en línea, así puede ser editado al mismo tiempo por varias personas, estas herramientas permiten editar el código fuente, diseño y demás componentes del aplicativo, de manera remota; esto es una ventaja, porque se puede visualizar cómo va quedando el proyecto.  Dentro de este orden de ideas, también se cuenta con herramienta de interpretación, las cuales permiten llevar un registro de las diferentes versiones que el software ha tenido y, de esta manera, controlar dicho aspecto, garantizando que los cambios realizados se llevan a cabo en la versión correcta.  Las nuevas tecnologías son el futuro de los sistemas de información, donde herramientas como TabNine permite, a través de una inteligencia artificial, ayudar al programador a mejorar su codificación y la manera como se emplea la lógica en el mismo; nos hacen pensar que en un futuro muy cercano podremos desarrollar aplicaciones de una manera mucho más eficiente, evitando cometer errores en la codificación; esto como aspecto de mejora, tiene toda una visión futurista. |

# Instalación y configuración de servidores

Para iniciar este proceso de exploración en instalación de servidores, debemos entender qué es un servidor, encontrando como definición que, es una computadora con unas características especiales en cuanto a rendimiento y almacenamiento, tiene como función principal ofrecer servicios a las diferentes computadoras que se encuentran conectadas a una red, estos servicios pueden ser almacenamiento de información, subida de archivos, bajada de archivos, procesamiento de información, correo electrónico, entre otras. Gracias a estas máquinas, es posible realizar el intercambio de información y utilizar los servicios que ahora usamos, tales como redes sociales, correo electrónico, entre otros. Los servidores son instalados en lugares donde el acceso a ellos es restringido y en algunas ocasiones con medidas de seguridad extrema porque la información que almacenan suele ser personal y no puede ser divulgada a terceros, debido a que se generarían problemas con dicha información.

**Contenedor web**

Ahora bien, los contenedores son una forma de virtualización del sistema operativo. Un solo contenedor se puede usar para ejecutar cualquier cosa, desde un microservicio a un proceso de software en una aplicación de mayor tamaño. Dentro de un contenedor se encuentran todos los ejecutables, el código binario, las bibliotecas y los archivos de configuración necesarios. Sin embargo, en comparación con los métodos de virtualización de máquinas o servidores, los contenedores no llevan imágenes del sistema operativo, esto los hace más ligeros y portátiles, con una sobrecarga significativamente menor. En implementaciones de aplicaciones de mayor tamaño, se pueden poner en marcha varios contenedores como uno o varios clústeres de contenedores. Estos clústeres se pueden gestionar mediante un orquestador de contenedores, como Kubernetes.



Este tipo de tecnologías son muy utilizadas por la mayoría de programadores y empresas que se dedican a la programación de aplicaciones, porque permiten la implementación de muchos servicios mediante la virtualización de los mismos. Si lo vemos desde un punto de vista de seguridad y practicidad es mucho mejor, ahora bien, si lo vemos desde el punto de vista de tiempos de respuesta son mucho más rápidos y permiten realizar un despliegue en máquinas que no tienen grandes recursos tecnológicos, lo cual se vuelve un punto a favor para empresas que no cuentan con equipos o servidores con gran potencia, es mucho más accesible y económico.

Los beneficios de estos contenedores son:

1. **Menos sobrecarga**: Los contenedores requieren menos recursos del sistema que los entornos de máquinas virtuales tradicionales o de hardware, porque no incluyen imágenes del sistema operativo.
2. **Mayor portabilidad**: Las aplicaciones que se ejecutan en contenedores se pueden poner en marcha fácilmente en sistemas operativos y plataformas de hardware diferentes.
3. **Funcionamiento más constante**: Los equipos de DevOps saben que las aplicaciones en contenedores van a ejecutarse igual, independientemente de donde se pongan en marcha.
4. **Mayor eficiencia**: Los contenedores permiten poner en marcha, aplicar parches o escalar las aplicaciones con mayor rapidez.
5. **Mejor desarrollo de aplicaciones**: Los contenedores respaldan los esfuerzos ágiles y de DevOps para acelerar los ciclos de desarrollo, prueba y producción.

Para tener mayor compresión sobre la instalación de este servicio, lo invitamos a ver el video Docker y kubernetes sobre Windows 10 WSL, el cual se encuentra en el material complementario.

**Base de datos**

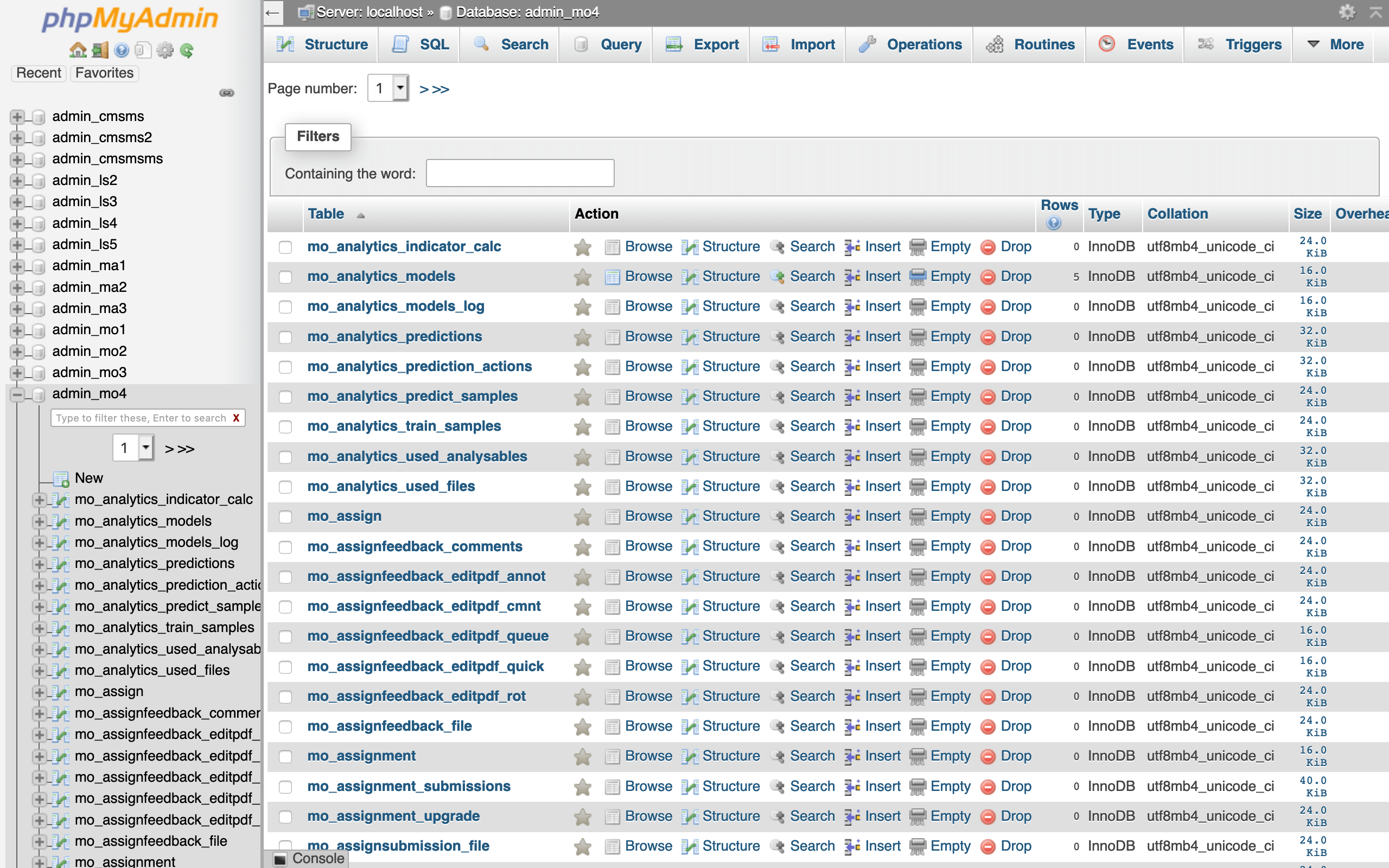
Las bases de datos son sistemas de almacenamiento que permiten resguardar la información de una manera organizada y segura; una característica es la velocidad de respuesta al momento de solicitar la búsqueda a realizar, pero en este momento nos concentramos en los servidores de bases de datos, los cuales son utilizados para el despliegue de información que presta servicios a millones de usuarios en la red. Uno de los aspectos a tener en cuenta al momento de realizar la instalación de un servidor de bases de datos, se refiere a los requerimientos de infraestructura que se necesitan para poder atender a la cantidad de usuarios que requieren de dicho servicio; esto se puede ver en la cantidad de estos, o de conexiones simultáneas que pueda soportar el servicio. Como bien se sabe, existen muchos motores de bases de datos que actualmente son utilizados para el almacenamiento de información, pero nos concentraremos en MySQL**.**



El servidor de bases de datos MySQL es un software que se encarga del almacenamiento y procesamiento de la información. Este servidor puede ser ejecutado de manera local o de manera remota, instalado en unos servicios de alojamiento llamados hosting, los cuales son servicios que se ofrecen a través de Internet y se tiene que adquirir un plan para utilizarlos. Una de las cosas a tener presente en un servidor de bases de datos es la configuración de acceso, teniendo en cuenta la cantidad de conexiones y usuarios autorizados para acceder a la información almacenada; otro aspecto se refiere a los permisos que se les da a esos usuarios para modificar la información; un ejemplo sería un usuario administrador y un usuario de consulta, el usuario administrador tendrá permisos de actualizar y modificar ciertas funciones e información del sistema, mientras que el usuario de consulta solo podrá acceder a información almacenada pero no podrá modificarla. A continuación, veremos estas dos opciones.

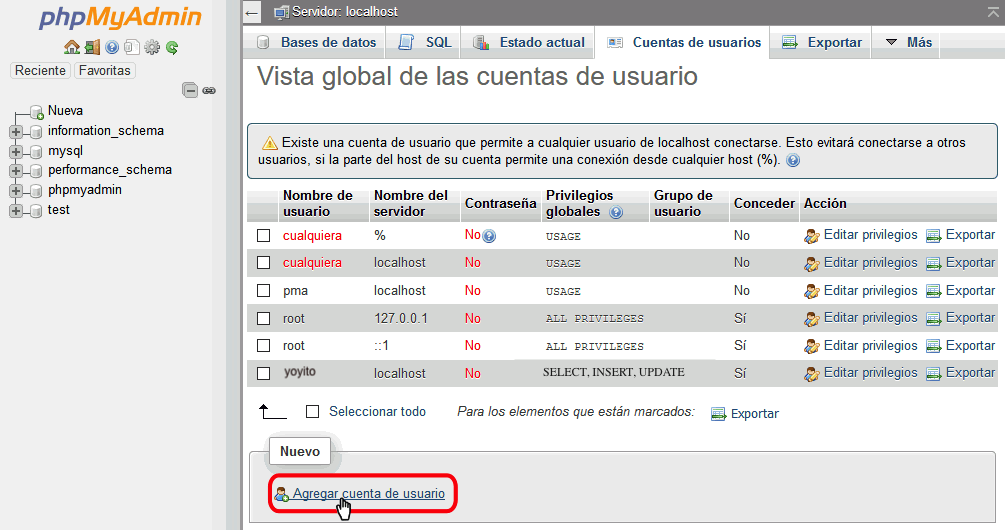
La siguiente imagen nos presenta un servidor de bases de datos MySQL.

1. Servidor de bases de datos MySQL



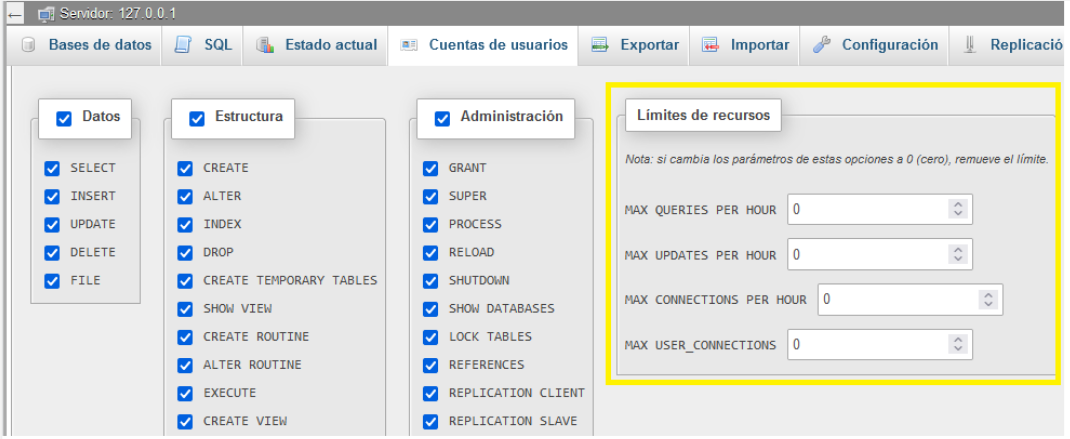
Como se puede apreciar en la siguiente imagen, tenemos varias cuentas de usuarios creadas en nuestro servidor de bases de datos, pero no todas tienen la misma configuración en cuanto a los privilegios y permisos que poseen. Por ejemplo, la cuenta llamada localhost tiene configurados todos los privilegios, es decir, puede realizar la modificación de todas las configuraciones del sistema, de hecho, tiene permisos para crear otros usuarios, pero en el caso de la cuenta llamada yoyito, solo tiene configurados algunos permisos como consultar, inserción, entre otros; esto quiere decir que en una base de datos, se deben configurar los usuarios para permitirles ciertos accesos para la manipulación y administración de los datos allí almacenados.

1. Configuración de cuentas de usuario

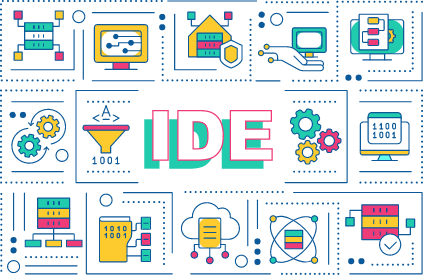


Y en la siguiente imagen se encuentran, seleccionados con color amarillo, los límites de recursos para asignar a los usuarios para su utilización. En este caso, se debe colocar el máximo de consultas, máximo de actualizaciones, máximo de conexiones por hora y cantidad de usuarios conectados. Estos límites se deben establecer de acuerdo con las necesidades de la organización que la implementa, pero existe algo particular en esta imagen y es que todos los números se encuentran en 0, esto quiere decir que no existe límite de conexiones, sino que el sistema de bases de datos, de acuerdo con los recursos que tenga disponibles la computadora, se usarán para cumplir la entrega y procesamiento de la información que requieren. Ahora bien, esto es una mala práctica desde el punto de vista real, es decir, esta configuración se puede utilizar en una computadora local para aprovechar todo el potencial del motor de la base de datos, pero al momento de instalar y configurar en un ambiente real, se deben tener en cuenta los límites establecidos y mencionados anteriormente.

1. Configuración de límite de recursos



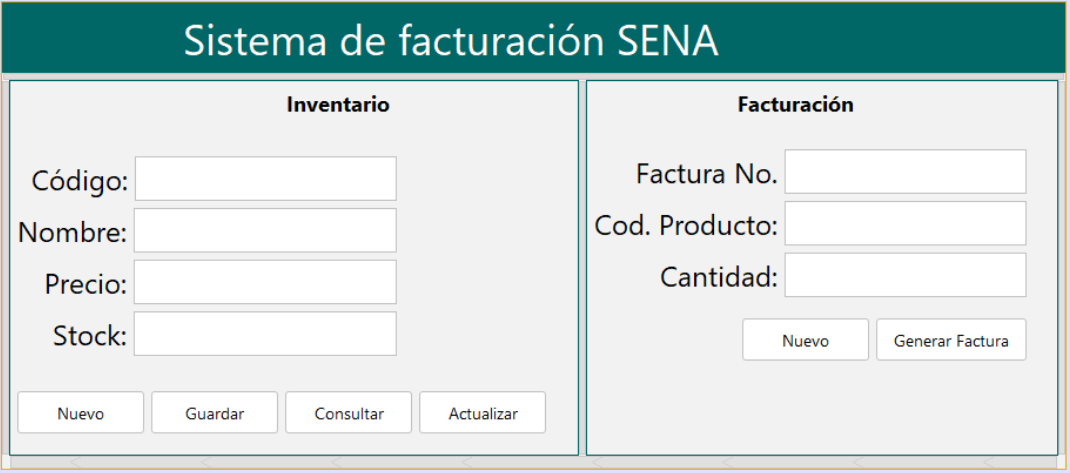
# Manejo de herramientas de desarrollo de *software*: IDEs



Las herramientas de desarrollo de software son las que hacen posible la construcción de las aplicaciones informáticas que manejamos a diario y son un tipo de software que permite la codificación y ensamble de dichos aplicativos. Ahora nos vamos a concentrar en los IDEs, lo que significa Entorno de desarrollo integrado; estos software permiten integrar tanto el Front-end como Back-end, es decir, permiten realizar las vistas o diseño del aplicativo, como también la codificación del mismo y, a su vez, la integración de estos; un ejemplo podría ser Visual Studio Code, que solo permite la codificación de lo que denominamos Back-end, pero no tiene un diseñador para la construcción de interfaz y es ahí donde encontramos las primeras diferencias significativas. Un entorno de desarrollo integrado es una herramienta que tiene todos los componentes para desarrollar una aplicación desde su creación, diseño, codificación y pruebas.

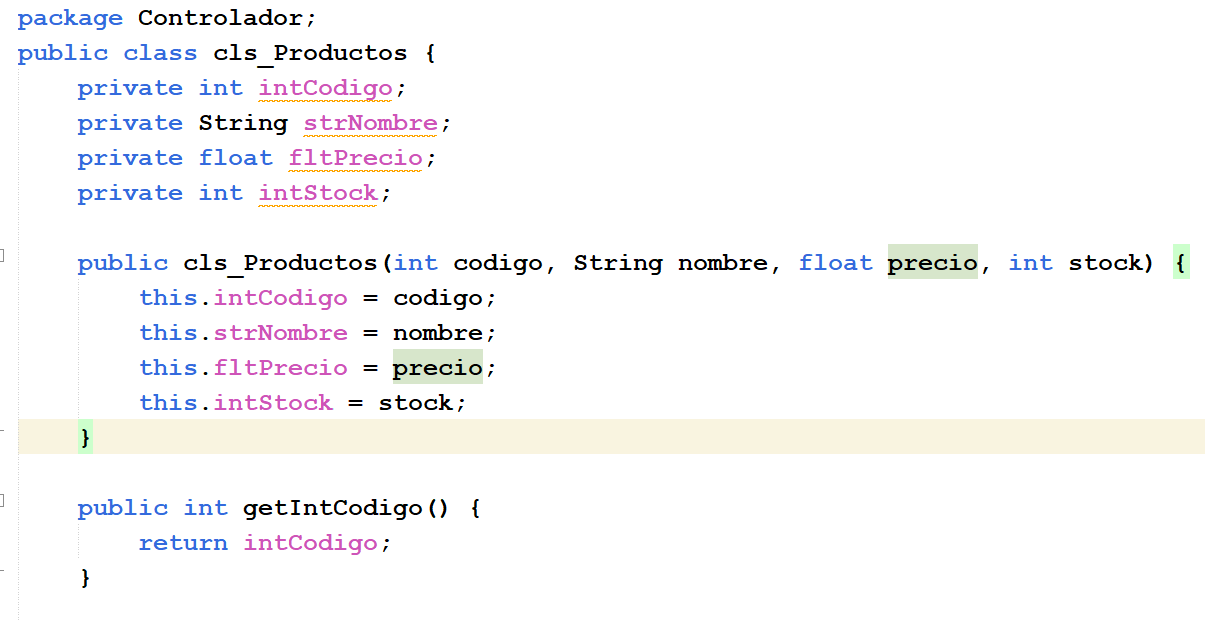
En la imagen a continuación, se observa la creación de un front-end de un software desarrollado en un aplicativo llamado Netbeans, el cual es un IDE que trabaja con el lenguaje de programación JAVA y mediante un diseñador, se pueden agregar los componentes requeridos para que el usuario pueda interactuar con el sistema.

1. Front-end



En la siguiente imagen, vemos el proceso de codificación del aplicativo, el cual se conectará con el front-end para recibir todas las instrucciones que el usuario desea que realice el programa, recordemos que el back-end depende de las funciones que el programador puso en el mismo y que de acuerdo a lo que el usuario realice así será la respuesta a entregar.

1. Back-end



Los entornos de desarrollo integrados son muy utilizados, debido a que permiten de una manera sencilla y eficiente, realizar todo el proceso de la construcción del software sin instalar una gran cantidad de elementos, que es lo que sucede en el caso de los editores de código, los cuales solo permiten trabajar con la codificación y luego toca realizar el diseño con otra herramienta y en algunas ocasiones se tienen problemas de compatibilidad entre las mismas, mientras que con el entorno de desarrollo integrado se garantiza la compatibilidad tanto de la codificación como del diseño e implementación del sistema.

# Estrategia DevOps

DevOps (software development “Dev” e IT operations “Ops”) es una de las últimas tendencias en el mundo del software. Pero, ¿qué son las DevOps? Los equipos DevOps son los intermediarios en la fase de implementación de un proyecto, conectando a los desarrolladores con las necesidades de operaciones y asegurándose que todo funcione de la manera más rápida y estable posible.

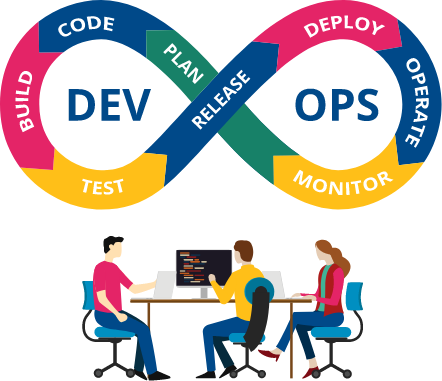
Así, los equipos DevOps se están convirtiendo en una parte fundamental de casi todas las empresas de software de la actualidad.

Para comprender el origen de DevOps, hay que hacer referencia a dos expertos en este campo: John Arundel y Justin Domingus, coautores de “Cloud Native DevOps with Kubernetes”. Si bien el enfoque principal del libro es Kubernetes, el sistema de software libre que también es conocido como K8s, habla sobre algunos de los principios claves de DevOps y sobre su creación. Antes de DevOps, el desarrollo y la operación de software eran esencialmente dos trabajos separados, realizados por dos grupos diferentes de personas.

Los desarrolladores escriben el software y se lo pasan al personal de operaciones, quienes ejecutan y mantienen el software. Entre estas dos áreas existía muy poca alineación, de hecho, los dos departamentos tenían objetivos e incentivos bastante diferentes, que a menudo entraban en conflicto entre sí. Los desarrolladores tienden a concentrarse y recompensar el envío rápido de nuevas funciones o versiones, mientras que los equipos de operaciones se preocupan por hacer que los servicios sean estables y confiables a largo plazo.

Con estos objetivos en conflicto, tiende a existir mucha discordia entre los roles de desarrollador y de operaciones, desafortunadamente, esta situación puede ser un obstáculo importante para las organizaciones que buscan innovar y superar a la competencia. Las personas que desarrollan el software deben comprender cómo este se relaciona con el resto del sistema y las personas que operan el sistema deben comprender cómo funciona o falla el software.

L a implementación exitosa de las prácticas de DevOps en una organización puede tener un impacto significativo a través de una mayor eficiencia, mayor seguridad y colaboración organizacional.



Las organizaciones que adoptan DevOps, pueden evolucionar y mejorar los productos de software mucho más rápido, en comparación con aquellas que utilizan procesos de desarrollo de software tradicionales.

DevOps es un puente entre el desarrollo y las operaciones en una organización y tiene como objetivo, mejorar la productividad en su conjunto.

Es así como las empresas que han empleado prácticas de DevOps, dedican un 21 % menos de tiempo al trabajo no planificado y al retrabajo, y un 44 % más de tiempo al trabajo nuevo, que conduce a una mayor eficiencia. Los tipos de metodología que implementan este tipo de estrategias, son metodologías ágiles tales como Scrum y Kanban, porque requieren de un control constante de cada uno de los procesos que se realizan en los proyectos.

En DevOps, debemos hablar de dos temas importantes:

1. **CI/CD Integracionistas Continua**: La integración continua CI/CD (continuous integration) es una práctica donde los desarrolladores combinan el código en un repositorio común, facilitando la realización de *test* o pruebas para detectar y resolver posibles errores. Con la CI se impide que se desarrollen distintas divisiones de una aplicación que luego puedan tener conflictos entre sí.

Los desarrolladores integran periódicamente su código en el repositorio central (como GitHub) en lugar de realizarlo de forma aislada al final del ciclo de producción (como se realizaba de forma tradicional). De esta manera, se descubren antes los conflictos entre los nuevos códigos y los existentes, haciendo que la resolución de los mismos sea más sencilla y acarree un menor costo.

En la integración continua, los *test* son fundamentales para poder garantizar que el código es confiable y será necesario contar con un sistema automatizado para poder realizar todo el proceso de integración de código y test (pruebas unitarias o testing de resolución, por ejemplo).

En el desarrollo moderno de aplicaciones, los equipos están compuestos por distintos desarrolladores que trabajan de forma simultánea en distintas funciones del programa. Con la integración continua estos desarrolladores podrán ir integrando o probando sus cambios y comprobando que funcionen de forma correcta y no entren en conflicto con los cambios de otros desarrolladores.

En el desarrollo de software, la integración continua es uno de los pilares en los que se basan metodologías como DevOps, para conseguir reducir costos, a la vez que se incrementan la calidad y la agilidad en el desarrollo.

1. **Entrega continua**: La integración continua, está relacionada con la entrega continua, la cual consiste en la automatización del proceso de entrega del software, permitiendo que pueda ser implementado en producción de forma confiable y sencilla. Una manera práctica, para entender la CD, es la entrega de actualizaciones de software a los usuarios o clientes, de forma sólida y continua.

La entrega continua en DevOps automatiza todo este proceso, desde que se integra el código en el repositorio único, se realizan las pruebas y se comprueban los cambios hasta la entrega al usuario.

La entrega continua se conoce como el siguiente paso o extensión de la integración continua. Tiene como objetivo principal conseguir que la entrega de los nuevos cambios o actualizaciones del software a los clientes se realicen de forma ágil y fiable.

Aunque la automatización es una de las características de la entrega continua, esta fase del desarrollo del *software*, tiene un componente de intervención humana; en la entrega continua, los desarrolladores controlan cuándo y cuántas veces se realiza la entrega del programa; la entrega continua acelera su entrega a los usuarios y puede ser realizada varias veces al día, a la semana, o según las necesidades o características de cada proyecto de desarrollo.

**Automatización**

Finalizamos el tema con la automatización de DevOps, que consiste en incorporar las tecnologías que ejecutan tareas con poca intervención humana, en los procesos que facilitan los ciclos de retroalimentación, entre los equipos de operaciones y los de desarrollo, para implementar las actualizaciones constantes de las aplicaciones en la producción con mayor rapidez; dicho de otra manera, es la forma de implementar herramientas que permitan alejar en algunos momentos, la intervención humana en los procesos de construcción de aplicaciones informáticas; por ejemplo, el proceso de elaboración de pruebas y casos de prueba para verificar el correcto funcionamiento de un sistema de información, se puede automatizar mediante el uso de algoritmos e inteligencia artificial, para que pueda encontrar huecos de seguridad en la codificación y que estos puedan ocasionar, de alguna manera, problemas en el futuro.

En la actualidad, por ejemplo, los entornos de desarrollo integrados incorporan herramientas de análisis de código y, de hecho, hasta entregan sugerencias para hacer que el código sea más eficiente, eliminando líneas de código innecesarias o que utilizan librerías viejas y entregan sugerencias para mejorarlo; todo esto va acompañado por inteligencia artificial y machine *learning*, que permiten conectar nuestra aplicación a estos servicios realizando, en tiempo real, este análisis y ayudando a mejorar la codificación de nuestras aplicaciones; otro proceso de automatización es el que implementa las bases de datos para realizar los procesos de copias de seguridad o *backups* de la información, con el fin de evitar, que esta se pueda perder o dañar; en este caso, también podemos ver un proceso de automatización.

# Técnicas de codificación y manejo de los servicios en la nube

Y como último aspecto en el abordaje de la temática de este componente formativo, estudiamos las técnicas de codificación de *software* que se refiere a la forma como un programador coloca las líneas que hacen funcionar al sistema; este proceso debe ser elaborado de manera organizada y tratando siempre que cada línea sea consecutiva del problema a resolver; otra de las apreciaciones que se debe tener en cuenta en este proceso, es la de realizar un lenguaje técnico comprensible, en el cual se deje la puerta abierta para el caso que otro programador lea el código y pueda comprender cuáles fueron las funciones y procesos que se realizaron en el *software* y, además, en muchas ocasiones se utilizan ciertas nomenclaturas para dar a entender cuáles controles o componentes, se están utilizando en el sistema.

En las técnicas de codificación encontramos:

1. **TDD (Test Development Drive**): DD es una técnica utilizada para el desarrollo de aplicaciones informáticas, que consiste en escribir los escenarios de pruebas y las pruebas en sí, para obtener los resultados favorables que debe arrojar la aplicación una vez esta sea construida; de esta manera se garantiza que los resultados obtenidos al momento de realizar las pruebas del *software*, se den de manera satisfactoria, aunque se debe tener en cuenta que la mayoría de veces, los programadores prefieren realizar la codificación del sistema y después realizar las respectivas pruebas; esto no es que sea una mala práctica sino que los datos que se utilizan en dichas pruebas, no logran satisfacer todas las necesidades de validaciones requeridas y, en muchas ocasiones, quedan procesos sin evaluar de manera correcta, lo que se puede convertir en un defecto en el *software*.
2. **no TDD**: Es una técnica que involucra un concepto opuesto a lo reflejado por TDD, donde las pruebas juegan un papel importante para la correcta codificación del aplicativo; aunque en muchas ocasiones se opta por un desarrollo más tradicional, donde primero se construye el código y luego se realizan las respectivas pruebas para obtener los criterios de aceptación que requiere el sistema para operar de manera correcta; aunque se podría llegar a pensar que es una desventaja, también involucra muchos aspectos favorables y uno de ellos es la independencia del código, es decir, se le confiere la libertad al desarrollador en la construcción del código y no se está amarrado al resultado esperado; es como cuando ya conoces el resultado de algo y solo trabajas para ello, muchas veces se deja a un lado la creatividad y muchos aspectos que, en muchas ocasiones, se convierten en el alma del proyecto.

Ahora bien, si nos concentramos en las ventajas que el TDD puede ofrecer al momento de generar código robusto y estable, es la mejor manera de realizarlo; es por eso que, al tener unos datos de prueba ya establecidos y los casos o condiciones que debe ejecutar dicho código, se evidencia un mejor desempeño del mismo, porque sabemos qué estamos buscando y cuáles serían las excepciones a controlar dentro del aplicativo.

Aunque podemos decir que es una excelente práctica, reiteramos que no muchas personas suelen aplicarlas, muchas de estas técnicas provienen de los movimientos de desarrollo ágil tales como Scrum, XP entre otros, debido a que se pretende realizar una constante revisión de los resultados arrojados por el software y los arrojados por las pruebas realizadas de manera previa, esto garantiza ya la obtención de los resultados correctos de hecho antes de la construcción del código a esto se le denomina criterio de aceptación que es cuando decimos la prueba tuvo éxito o no, ahora bien, es importante tener en cuenta que el código escrito es un código básico que no contempla grandes particularidades luego que se verifica que pasa la prueba se refactoriza y se agregan todos los componentes necesarios.

**Manejo de los servicios en la nube**

Los servicios en la nube son una alternativa bastante favorable, no solo por la facilidad y seguridad al momento de realizar el despliegue de aplicaciones y servicios en Internet, sino que ofrecen un bajo costo para quienes utilizan estos servicios.

Para conocer cada uno de estos servicios, lo invitamos a leer el documento “Manejo de los servicios en la nube”.

Anexo: [Manejo de los servicios en la nube](ANEXO_Manejo_de_los_servicios_en_la_nube.pdf)

A continuación, presentamos un resumen de estos servicios:

1. **SaaS**: Con este modelo, su empresa no necesita instalar, mantener y actualizar *hardware* y *software*. El acceso es fácil y simple: solo es necesario contar con una conexión a internet.
2. **PaaS**: Al no tener que preocuparse por los recursos de hardware y software (sistema operativo), mejoran su eficacia, centrándose solo en la parte que les interesa.
3. **IaaS**: La contratación de este hardware permite elegir la capacidad de proceso (procesadores), la memoria a utilizar (memoria RAM) y el espacio de almacenamiento (disco duro).
4. **CaaS**: Son un servicio de nube que permite gestionar e implementar aplicaciones usando el aislamiento en contenedores, y que se puede implementar en las instalaciones o en la nube.

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

El siguiente mapa integra los criterios y especificidades de los conocimientos expuestos en el presente componente formativo.

La imagen nos muestra un mapa conceptual del contenido estudiado en este componente formativo. 

Se divide en ocho componentes 

1. Redes: Configuración de redes, seguridad (roles, usuarios, permisos), comando de red. 

2. Sistemas operativos: Configuración y manejo de comandos. 

3. Tecnologías  de versionamiento: GIT, GITHUB, GITLAB

4.Instalación y configuración de servidores: Contenedor web, Base de datos. 

5.Manejo de herramientas de desarrollo de software: IDEs 

6. Estrategia DevOps: CI/DI Integracionistas continua + entrega continua, automatización. 

7. Técnicas de codificación: TDD ( Test Development Drive) y no TDD 

8. Manejo de los servicios en la nube: SaaS, PaaS, laaS, CaaS 

[Anexo. Síntesis](sintesis.pdf)

Material complementario

|  |  |  |  |
| --- | --- | --- | --- |
| Tema | Referencia APA del Material | Tipo de material | Enlace del Recurso o  Archivo del documento material |
| Sistemas operativos | TecnoMáticas. (2020). *Sistemas Operativos ¿Qué es? Características, tipos, función, TODO sobre S.O.* (video). YouTube. | Video | <https://www.youtube.com/watch?v=1C8x3nn-u_w> |
| Instalación y configuración de servidores | Digitalthinking with sotobotero. (2021). *Docker y kubernetes sobre Windows 10* wls2 | Video | <https://www.youtube.com/watch?v=OGtLAMUyDrc> |
| Manejo de herramientas de desarrollo de software: IDEs | OS Infor. (2019). *Metodologías del desarrollo software – Análisis y diseño* (video). | Video | <https://www.youtube.com/watch?v=5tyU_h69F-s&t=310s> |
| Técnicas de codificación y manejo de servicios en la nube | Païola, P. (2021). *Microsoft Azure: Gestione su Sistema de Información en la Nube.* Ediciones ENI. | Libro | <https://sena-primo.hosted.exlibrisgroup.com/primo-explore/fulldisplay?docid=sena_biblioteca_eniEPT3AZWIN&vid=SENA&search_scope=sena_completo&tab=sena_completo&lang=es_ES&context=L> |

Glosario

**Amazon Web Service**: es una plataforma que ofrece servicios de computación en la nube para alojar aplicaciones informáticas desarrollada y gestionada por Amazon.

**Azure**: es una plataforma que pertenece a los servicios de computación en la nube para alojar aplicaciones informáticas y es desarrollada por la empresa Microsoft.

**Base de datos**: es un software que permite almacenar información de manera permanente y organizada es considerado uno de los mejores modelos de almacenamiento de información.

**CaaS**: son un servicio de nube que permite gestionar e implementar aplicaciones usando el aislamiento en contenedores, y que se puede implementar en las instalaciones o en la nube.

**GIT:** es una plataforma de versionamiento de código que permite controlar los cambios realizados en la misma.

**GITHUB**: es una plataforma de repositorio de código para trabajar de manera colaborativa entre equipos de desarrollo de software.

**IaaS**: en este tipo de servicios estamos hablando de infraestructuras como servicio. Las empresas contratan la infraestructura de hardware a un tercero a cambio de una cuota o alquiler. La contratación de este hardware permite elegir la capacidad de proceso (procesadores), la memoria a utilizar (memoria RAM) y el espacio de almacenamiento (disco duro).

**LINUX**: es un sistema operativo desarrollado por Linus Torvalds y es considerado uno de los más seguros del mundo utilizado para gestionar grandes volúmenes de información.

**MAC** **OS**: es un sistema operativo desarrollado por la empresa Apple y está presente en muchos dispositivos tales como: tablets, celulares, computadores personales, servidores entre otros.

**PaaS**: el servicio PaaS ofrece plataformas como servicios. En estas plataformas se pueden lanzar aplicaciones como bases de datos, middleware, herramientas de desarrollo, servicios de inteligencia empresarial, etc.

**Roles**: es un nivel que recibe un usuario en un sistema informático con el cual tendrá los permisos necesarios para hacer uso de una aplicación informática.

**SaaS**: es una forma de poner a disposición software y soluciones de tecnología por medio de la internet, como un servicio. Con este modelo, la empresa no necesita instalar, mantener y actualizar hardware y software.

**TDD**: es una técnica que se utiliza para realizar pruebas unitarias al código de una aplicación informática.

**Usuario**: es el nombre que recibe una persona para hacer uso de un sistema informático.

**Windows**: es un sistema operativo desarrollado por la empresa Microsoft y es considerado el más popular de todos.

Referencias bibliográficas

Dauzon, S. (2022). *Git : Controle la gestión de sus versiones (conceptos, utilización y casos prácticos)* (2ª edición). Ediciones ENI. <https://sena-primo.hosted.exlibrisgroup.com/permalink/f/1j5choe/sena_biblioteca_eniEPT3GIT>

Khtan66. (2016). Kubernetes <https://commons.wikimedia.org/wiki/File:Kubernetes.png>

Ramírez, R., Pellecchia, Á., Saa, A., Santos, A., González, H de J. & Bonilla-Morales, B. (2022). Prácticas orientadas por pruebas para el desarrollo de software, una revisión sistemática. *Revista De Iniciación Científica*, 8(2), 50-56.<https://sena-primo.hosted.exlibrisgroup.com/permalink/f/1i756fj/TN_cdi_doaj_primary_oai_doaj_org_article_ba44f4c2c32a4970b1573271c3f1b5b7>

Créditos

Elaborado por Innovative Education

