**REPÚBLICA DOMINICANA**

**INSTITUTO TECNOLÓGICO DE LAS AMÉRICAS**

**ESCUELA DE SOFTWARE**

**Programación 3**



**Herramientas de administración de fuentes**

**Leandro E. Pujols Polanco**

**2021-1308**

**Ing. Kelyn Tejada Belliard**

Santo Domingo, República Dominicana, Santo Domingo Este

Julio de 2024

Contenido

[Introducción 3](#_Toc173179816)

[¿Qué es Git? 4](#_Toc173179817)

[¿Cuál es el propósito del comando git init en Git? 4](#_Toc173179818)

[¿Qué representa una rama en Git y cómo se utiliza? 4](#_Toc173179819)

[¿Cómo puedo determinar en qué rama estoy actualmente en Git? 4](#_Toc173179820)

[¿Quién es la persona responsable de la creación de Git y cuándo fue desarrollado? 5](#_Toc173179821)

[¿Cuáles son algunos de los comandos esenciales de Git y para qué se utilizan? 5](#_Toc173179822)

[¿Puedes mencionar algunos de los repositorios de Git más reconocidos y utilizados en la actualidad? 6](#_Toc173179823)

[Conclusión 7](#_Toc173179824)

# Introducción

Git es una herramienta de control de versiones diseñada por Linus Torvalds en 2005, inicialmente para gestionar el código fuente del núcleo de Linux. La necesidad de un sistema eficiente, confiable y compatible para manejar proyectos grandes con múltiples colaboradores llevó a la creación de Git. Esta herramienta permite llevar un registro detallado de los cambios en archivos de código fuente, facilitando la coordinación del trabajo entre varias personas en un repositorio compartido. A través de comandos esenciales como git init, git clone, git add, y git commit, entre otros, Git ofrece un robusto conjunto de funcionalidades para la gestión del desarrollo de software. Además, las ramas en Git permiten desarrollar nuevas características y corregir errores en áreas aisladas del repositorio principal, asegurando un flujo de trabajo ordenado y eficiente.

# ¿Qué es Git?

Git es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia, la confiabilidad y compatibilidad del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos de código fuente. Su propósito es llevar registro de los cambios en archivos de computadora incluyendo coordinar el trabajo que varias personas realizan sobre archivos compartidos en un repositorio de código.

# ¿Cuál es el propósito del comando git init en Git?

Este comando crea un repositorio Git vacío. Básicamente un directorio .git con subdirectorios para objects, reas/heads, reos/tags, y ficheros de plantillas. Se creará una rama inicial sin confirmaciones (ver la opción --initial-branch abajo por su nombre).

Si la variable de ambiente $GIT\_DIR se configura, entonces especifica una ruta a usar en lugar de ./.git para la base del repositorio.

Si el directorio de almacenamiento de objetos se especifica con la variable de ambiente $GIT\_OBJECT\_DIRECTORY, entonces los directorios sha1 se crean bajo él, de lo contrario se usa el directorio predeterminado $GIT\_DIR/objects.

Ejecutar git init en un repositorio existente es seguro. No sobre-escribirá cosas que ya existan ahí. La razón primaria para volver a ejecutar git init es para recoger plantillas agregadas recientemente (o para mover el repositorio a otro lugar si se proporciona --separate-git-dir).

# ¿Qué representa una rama en Git y cómo se utiliza?

Las ramas te permiten desarrollar características, corregir errores, o experimentar con seguridad las ideas nuevas en un área contenida de tu repositorio.

Siempre puedes crear una rama a partir de otra rama existente. Habitualmente, puedes crear una rama nueva desde la rama predeterminada de tu repositorio. Podrás entonces trabajar en esta rama nueva aislado de los cambios que otras personas hacen al repositorio. A la rama que creas para construir una característica se le conoce como rama de característica o rama de tema.

# ¿Cómo puedo determinar en qué rama estoy actualmente en Git?

Cuando creas un repositorio con contenido en GitHub.com, GitHub crea el repositorio con una sola rama. Esta primera rama en el repositorio es la rama predeterminada. La rama predeterminada es la rama que GitHub muestra cuando alguien visita tu repositorio. La rama predeterminada también es la rama inicial que Git verifica localmente cuando alguien clona el repositorio. A menos de que especifiques una rama diferente, la rama predeterminada en un repositorio será la rama base para las sgnmholicitudes de cambio nuevas y para las confirmaciones de código.

De manera predeterminada, GitHub asigna el nombre main a la rama predeterminada en cualquier repositorio nuevo.Puedes cambiar la rama predeterminada de un repositorio existente.

Git Checkout

En esta página, se analiza el comando git checkout. Se abordarán ejemplos de uso y casos extremos. En términos de Git, "checkout" (extraer) es el acto de cambiar entre diferentes versiones de una entidad objetivo. El comando git checkout opera sobre tres entidades distintas: archivos, confirmaciones y ramas. Además de esta definición, se utiliza habitualmente la expresión "checking out" o "extraer" para referirse al acto de ejecutar el comando git checkout. En el tema sobre cómo deshacer cambios, explicamos cómo puede usarse git checkout para ver confirmaciones antiguas. El eje central de la mayor parte de este documento serán las operaciones de extracción en las ramas.

Los cambios de rama se asemejan al cambio de confirmaciones y archivos antiguos, ya que el directorio de trabajo se actualiza para reflejar la rama o revisión seleccionada. Sin embargo, los cambios nuevos se guardan en el historial del proyecto, es decir, no se trata de una operación de solo lectura.

# ¿Quién es la persona responsable de la creación de Git y cuándo fue desarrollado?

Git fue creado por Linus Torvalds en 2005. Fue desarrollado para gestionar el código fuente del núcleo de Linux, debido a la necesidad de un sistema de control de versiones rápido y eficiente que pudiera manejar proyectos grandes con muchos colaboradores. Linus Torvalds es también conocido por ser el creador del sistema operativo Linux.

# ¿Cuáles son algunos de los comandos esenciales de Git y para qué se utilizan?

algunos comandos esenciales de Git:

git init: Inicializa un nuevo repositorio de Git.

git clone [url]: Clona un repositorio remoto en tu máquina local.

git add [archivo]: Añade cambios en archivos específicos al área de preparación (staging).

git commit -m "mensaje": Guarda los cambios preparados en el historial del repositorio con un mensaje descriptivo.

git status: Muestra el estado de los archivos en el directorio de trabajo y el área de preparación.

git push: Sube los cambios del repositorio local a un repositorio remoto.

git pull: Descarga cambios desde un repositorio remoto y fusiona en el repositorio local.

git branch: Muestra las ramas locales o crea nuevas ramas.

git checkout [nombre-rama]: Cambia a otra rama.

git merge [nombre-rama]: Fusiona cambios de otra rama en la rama actual.

# ¿Puedes mencionar algunos de los repositorios de Git más reconocidos y utilizados en la actualidad?

Algunos de los repositorios de Git más reconocidos y utilizados son:

**Linux Kernel:** El código fuente del núcleo de Linux es gestionado utilizando Git.

**TensorFlow:** Un popular marco de aprendhfkfhkfizaje automático de Google.

**React:** Una biblioteca de JavaScript para construir interfaces de usuario, mantenida por Facebook.

**Django:** Un framework de desarrollo web en Python.

**Node.js:** Un entorno de ejecución para JavaScript fuera del navegador.

**Bootstrap:** Un popular framework de front-end para el diseño de sitios web.

**Kubernetes:** Un sistema de orquestación de contenedores de código abierto desarrollado por Google.

Estos repositorios están alojados en plataformas como **GitHub, GitLab, y Bitbucket**, que son servicios que facilitan el alojamiento, gestión y colaboración en proyectos Git.

# Conclusión

Para concluir, Git es una herramienta fundamental en el desarrollo de software moderno, proporcionando una estructura sólida para el control de versiones y la colaboración en proyectos. Su capacidad para manejar grandes cantidades de archivos y coordinar el trabajo de múltiples desarrolladores lo convierte en la elección preferida para proyectos de cualquier escala. El uso de comandos básicos y avanzados, junto con la gestión efectiva de ramas, permite a los equipos de desarrollo mantener un historial claro y detallado de los cambios realizados, facilitando la integración y el despliegue continuo. Con repositorios reconocidos y utilizados globalmente, como el del núcleo de Linux, TensorFlow, y React, Git se ha establecido como una herramienta indispensable en la industria del software.