

Git

**Universidad De Las Fuerzas Armadas Espe  
bderazo**[**@**](mailto:parodi_caj@up.edu.pe)**espe.edu.ec**

# 

**Resumen Ejecutivo**

Git es un sistema de control de versiones distribuido que ha revolucionado la forma en que los equipos de desarrollo gestionan el código fuente a lo largo del tiempo. Esta herramienta de código abierto cuenta con una arquitectura distribuida que permite a cada usuario tener una copia local completa del repositorio, facilitando el trabajo sin conexión y la colaboración remota. Sus principales funciones incluyen la creación de ramas para trabajar en paralelo, la fusión de ramas para integrar cambios y la creación de etiquetas para marcar versiones.

Entre las ventajas de Git destacan la colaboración en equipo, la recuperación sencilla de versiones anteriores y la capacidad de detectar y resolver problemas en el código. No obstante, algunas desventajas se relacionan con su curva de aprendizaje pronunciada y su complejidad para usuarios nuevos.

La instalación de Git se logra fácilmente a través del sitio web oficial, y su utilización puede aprenderse mediante tutoriales en línea. Atlassian ofrece un ejemplo práctico en este sentido.

En conclusión, Git es una herramienta fundamental para el control de versiones en el desarrollo de software. Su arquitectura distribuida, funciones poderosas y ventajas en trabajo en equipo lo hacen ideal para proyectos de todos los tamaños, a pesar de requerir cierto esfuerzo inicial de aprendizaje.

# GIT

1. **Definición y características**

Git es un sistema de control de versiones distribuido que permite gestionar los cambios en el código fuente a lo largo del tiempo. Los repositorios locales plenamente funcionales permiten trabajar sin conexión o de forma remota con facilidad. (*¿Qué Es Git? - Azure DevOps*, 2023) Git es un controlador de versiones con arquitectura distribuida. Mantiene un espacio de trabajo compartido entre todos los integrantes del proyecto. (*En Resumen: Control De Versiones*, 2021)Además de ser de código abierto, cuenta con mantenimiento activo por una comunidad bastante grande y es muy seguro. Puede albergar el historial completo de todos los cambios. (*Git - Acerca Del Control De Versiones*, n.d.)

Git es un control de versiones distribuido que permite rastrear y gestionar los cambios realizados en el código fuente. A diferencia de los sistemas centralizados, Git opera con una arquitectura distribuida, lo que significa que cada usuario tiene una copia local completa del repositorio, lo que facilita el trabajo sin conexión y la colaboración remota. Es una herramienta de código abierto respaldada por una comunidad activa y comprometida. Su robusta seguridad y capacidad para almacenar el historial completo de cambios lo convierten en una opción confiable para proyectos de todos los tamaños.

# Funciones principales

Las funciones principales de Git son la creación de ramas para trabajar en paralelo, la fusión de ramas para integrar cambios y la creación de etiquetas para marcar versiones. (*En Resumen: Control De Versiones*, 2021)

Entre las funciones clave de Git, destacan la creación de ramas, que permite trabajar en paralelo y desarrollar nuevas características sin afectar al código principal. La fusión de ramas es otra función fundamental, que permite integrar los cambios de diferentes ramas en el repositorio principal. Además, Git permite crear etiquetas para marcar versiones específicas del código, lo que facilita la identificación y el seguimiento de versiones estables.

# Ventajas y desventajas

Entre las ventajas del control de versiones se encuentran la posibilidad de trabajar en equipo, la facilidad para recuperar versiones anteriores del código fuente y la capacidad para detectar y solucionar problemas en el código. (Castellanos, 2021)

El uso de Git ofrece numerosas ventajas para los equipos de desarrollo. La capacidad de trabajar en equipo de manera colaborativa y coordinada es uno de los mayores beneficios. También proporciona la flexibilidad para revertir a versiones anteriores del código en caso de errores o problemas. La detección temprana y la solución rápida de problemas en el código son posibles gracias a las características de control de versiones de Git.

Entre las desventajas se encuentran la complejidad en su uso y la necesidad de aprender a utilizarlo. (*En Resumen: Control De Versiones*, 2021)

Sin embargo, Git también presenta desventajas, especialmente para aquellos que se están familiarizando con la herramienta. Su curva de aprendizaje puede ser pronunciada, y algunos usuarios pueden encontrarlo complejo en comparación con sistemas centralizados más tradicionales.

# Instalación y configuración

Git se puede instalar fácilmente descargando el paquete correspondiente desde su sitio web oficial. La configuración inicial suele ser sencilla, pero es importante seguir las pautas adecuadas para garantizar un uso eficiente y seguro de la herramienta.

1. Descarga el instalador de GIT para Windows.
2. Una vez que hayas descargado el instalador, haz doble clic sobre el ejecutable para que comience el proceso de instalación y sigue las instrucciones que te aparecerán en pantalla. Al igual que cualquier otro programa, tendrás que dar “Next” (siguiente) en varias ocasiones hasta que aparezca la opción “Finish” (terminar) para completar la instalación.
3. Ahora tienes que abrir el símbolo de sistema y escribir los siguientes comandos en la terminal:

git config --global user.name "Tu nombre"

git config --global user.email "[ejemplo@email.com](mailto:ejemplo@email.com)"

Recuerda que debes cambiar tu Nombre y ejemplo@email.com por tu información.

# Utilización

cd //Desktop/Gestion/repositorio

git init

git add README.md

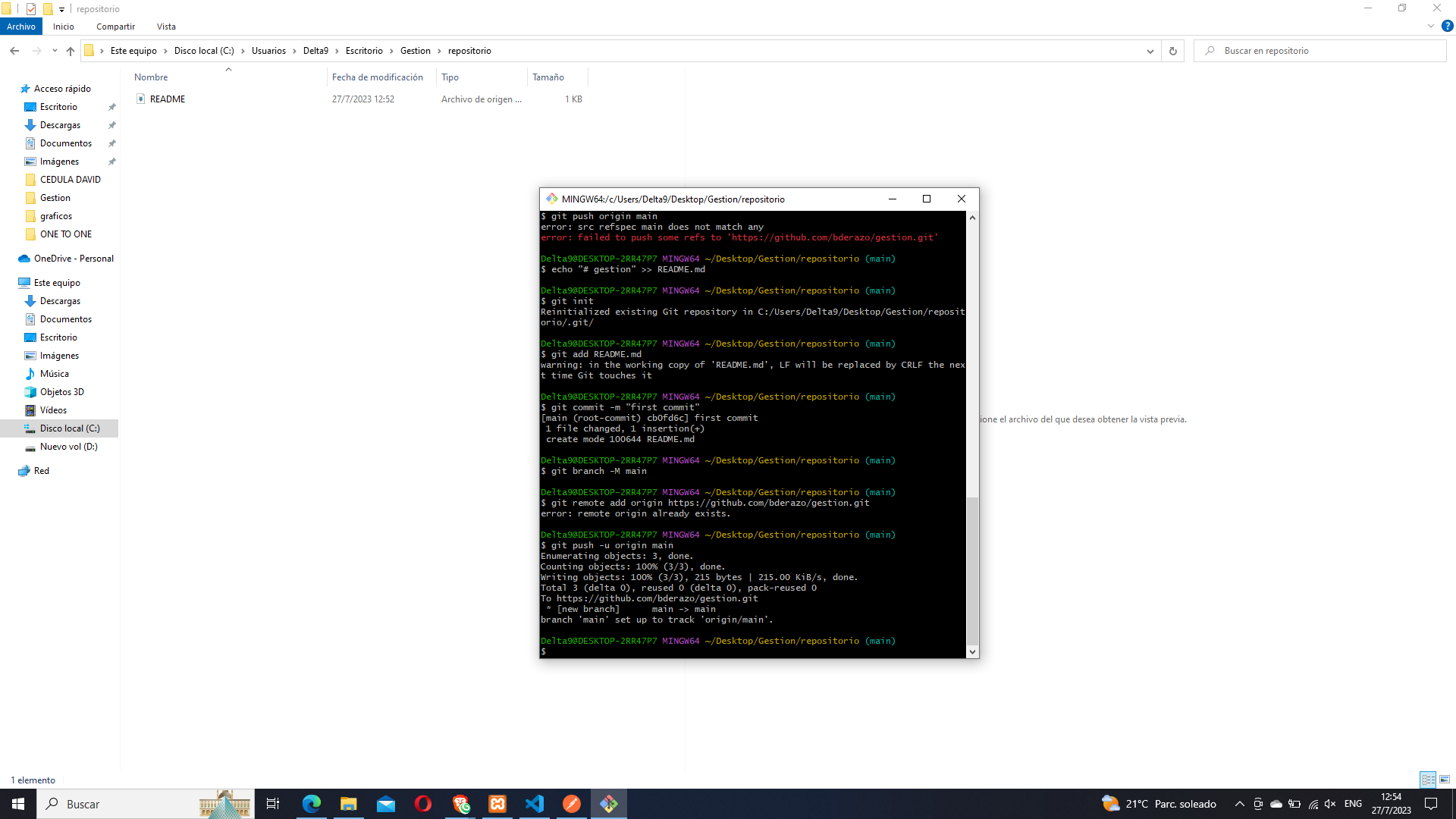
git commit -m "first commit"

git branch -M main

git remote add origin https://github.com/bderazo/gestion.git

git push -u origin main

# Conclusiones



# Bibliografía

AMIGHINI Alessia

2015 “China’s Economic Growth. Heading to a New Normal”, en, Amighini Alessia y Axel Berfosky (ed.), *Xi’s Policy Gambles: The Bumpy Road Ahead*, Italian Institute for International Political Studies Report, pp. 48-64.

ASLAM Aqib, BOZ Emme, Cerutti Eugenio, POPLAWSKI, Marcos y Petia TOPAROVA 2017 *The Slowdown in Global Trade: A sympton of a weak recovery*, IMF Working Paper WP/17/242, Washington.

ASLUND Anders

2013 *Why Growth in Emeging Economies is likely to fall?,* Petersen Institute for International Economics, Working Paper 13-10, Washington.

ATRADIUS

2016 *China’s rebalancing: still a long way ago*, Atradius Economic Research, November 2016.

BACKHOUSE Roger y Mauro BOIANOVSKY

2016 “Secular Stagnation: The history of a macroeconomic heresy”, *The European Journal of the History of Economic Thought* 23:6, pp. 946-970.

BANERJEE Ryan, KEARNS Jonhatan y Marco LOMBARDI

2015 *(Why) Is investment weak?* BIS Quartely Review, march, pp. 67-82.

BANK OF INTERNATIONAL SETTLEMENTS (BIS)

2014 BPI *84° Informe Annual 2013-2014*, Ginebra.

BORIO Claudio

2017 *Secular Stagnation or financial cycle drag?* Keynote speech at the National Association for Business Economics, 33rd Economic Policy Conference, Washington DC.

2012 The financial cycle and macroeconomics: what have we learnt? BIS Working Paper 395, Bank of International Settlements, Basilea.

BUSSIERE Mathieu, FERRARA Laurent y Juliana MILOVICH

2015 *Explaining the recent slump in investment: the role of expected demand and uncertainty*, Banco de Francia, Documento de Trabajo N° 571, Paris.