



El futuro digital  
es de todos

MinTIC



Git



Universidad de Caldas

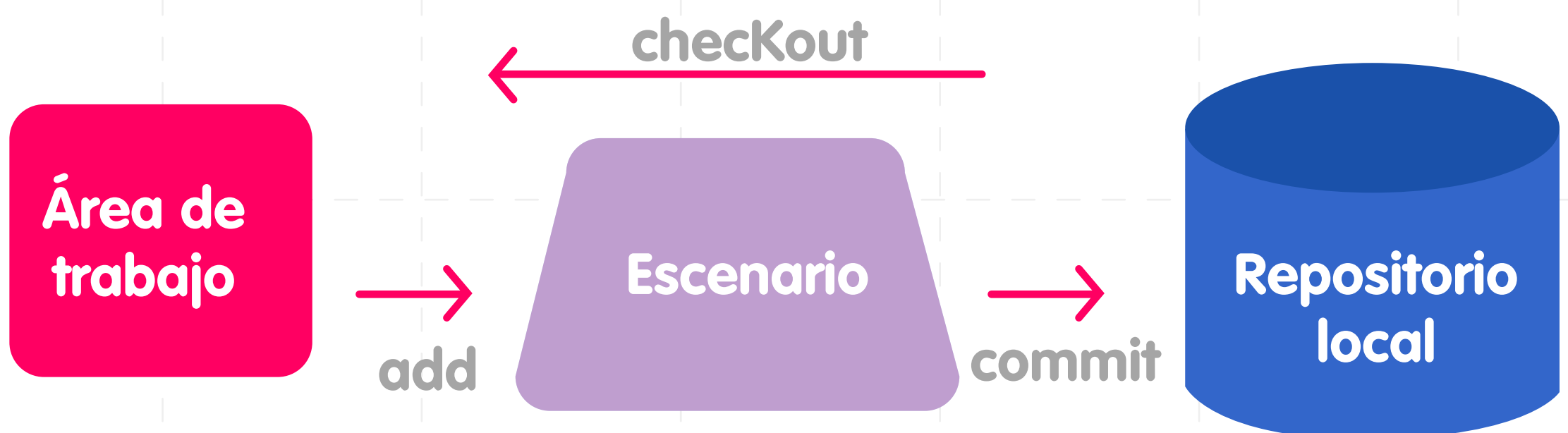
# Hola:

Git es un sistema de control de versiones, que tiene un esquema distribuido, es decir, el mismo repositorio se puede compartir y sincronizar periódicamente con diferentes integrantes del equipo, cada uno trabajando en sus diferentes elementos. Esto permite independencia, coordinación y mayor seguridad para recuperar las versiones.

En el esquema de trabajo en Git habrá un área con los archivos correspondientes al entorno de desarrollo en el que se está trabajando. En este lugar se realizan los cambios y pruebas necesarias. Además, Git dispone de un repositorio local donde va a estar el historial de versiones. Entre estos dos elementos hay un escenario llamado *'Staging area'* o *'index'*, donde se colocan temporalmente algunos archivos del área de trabajo, que posteriormente pasarán a la versión definitiva del repositorio local.

Algunas de las instrucciones que más se usan en Git son:

- *'add'*: para pasar archivos del área de trabajo al *index*.
- *'commit'*: para pasar de ese *index* al repositorio, creando así una nueva versión.
- *'checkout'*: Permite volver a una versión anterior.

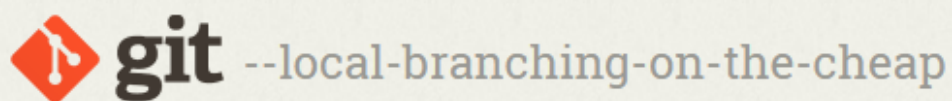


Los pasos para trabajar con Git son los siguientes:

- Crear el repositorio con la instrucción *'init'* y, posteriormente trabajar con los archivos del entorno de desarrollo.
- *'add'* permite adicionar los archivos al entorno.
- *'commit'* es la instrucción para guardar.

A medida que se trabaja con Git, internamente en el repositorio se crea una estructura con las diferentes versiones. La estructura más simple consta de una rama, que por lo general se llama *"master"* o *"main"*, y allí, a medida que se ejecuta *'commit'*, se guardan las versiones, tanto la actual como las anteriores.

En conclusión, Git permite determinar el esquema de trabajo a través del uso de un repositorio que se crea mediante instrucciones correspondientes que actúan de manera interna.



🔍 Search entire site...

Git is a **free and open source** distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is **easy to learn** and has a **tiny footprint with lightning fast performance**. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like **cheap local branching**, convenient **staging areas**, and **multiple workflows**.





Universidad de Caldas