



Programación de Sistemas

CCPG1008

Federico Domínguez, PhD.

Unidad 2 – Sesión 3: Versionamiento de código con Git

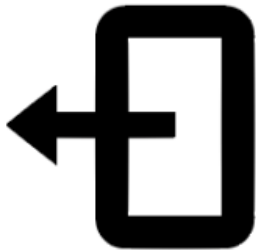
In case of fire



1. `git commit`



2. `git push`



3. `leave building`

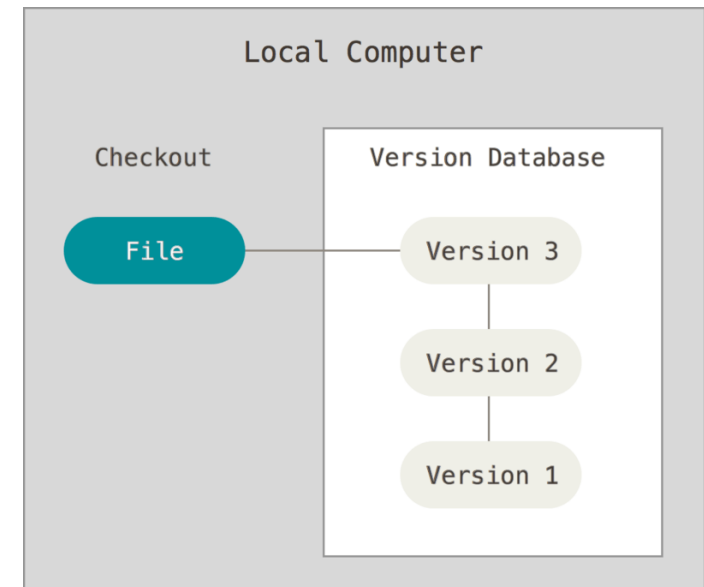
Contenidos

1. Versionamiento de código con Git

Versionamiento de código

Software de control de versión: Gestiona y mantiene las diferentes versiones de archivos de código fuente en un proyecto de desarrollo de software.

- Bitácora de los cambios en el proyecto
- Auditoría de cambios (quién hizo qué?)
- Manejo de diferentes versiones o funcionalidades en el proyecto
- Gestión de trabajo en equipo
- Respaldo del proyecto distribuido o centralizado



Versionamiento de código

Sistemas de versionamiento más populares:

CVS: Code Versioning System, código abierto, está entre los primeros

SVN: Subversion, código abierto, una mejora ante CVS y uno de los más populares

Git: Código abierto, cada vez más popular, completamente distribuido

Mercurial: Código abierto

Team Foundation Version Control: Microsoft

BitKeeper: Código abierto desde el 2016



Git

Creado por Linus Torvalds

Usado en el desarrollo del kernel de Linux

Extremadamente rápido y completamente distribuido

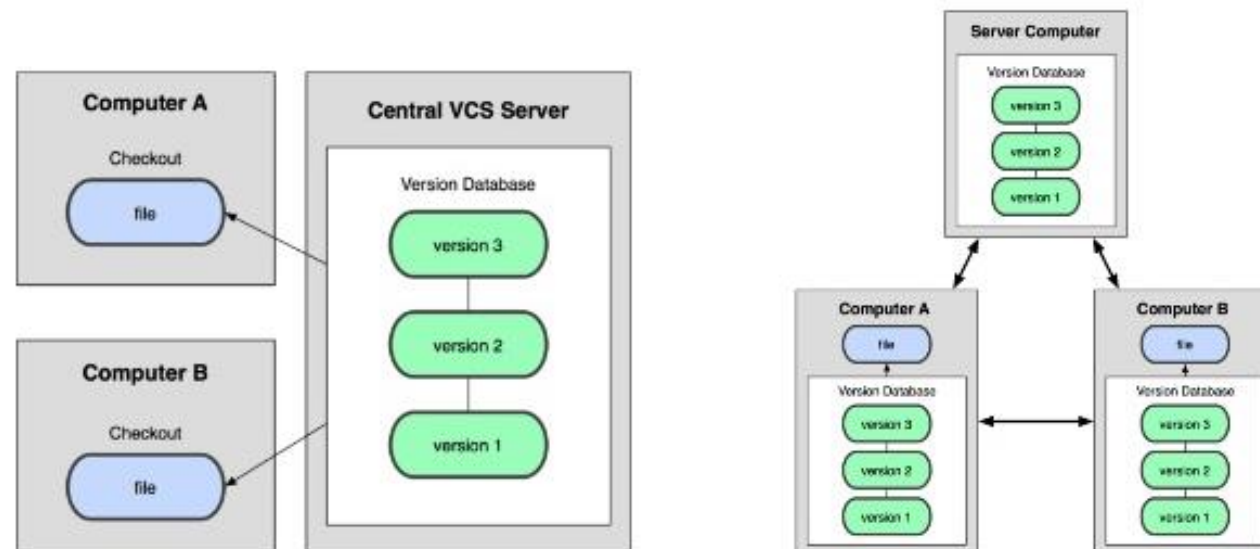
Portales como Github y Bitbucket han incrementado exponencialmente su popularidad

En este curso usaremos Git en conjunto con Bitbucket: bitbucket.org

Git

Excelente referencia: ProGit 2da Edición, gratuito en línea: <https://git-scm.com/book/en/v2>

Central Vs. Distributed



Git

Concepto principal de Git: el **commit**

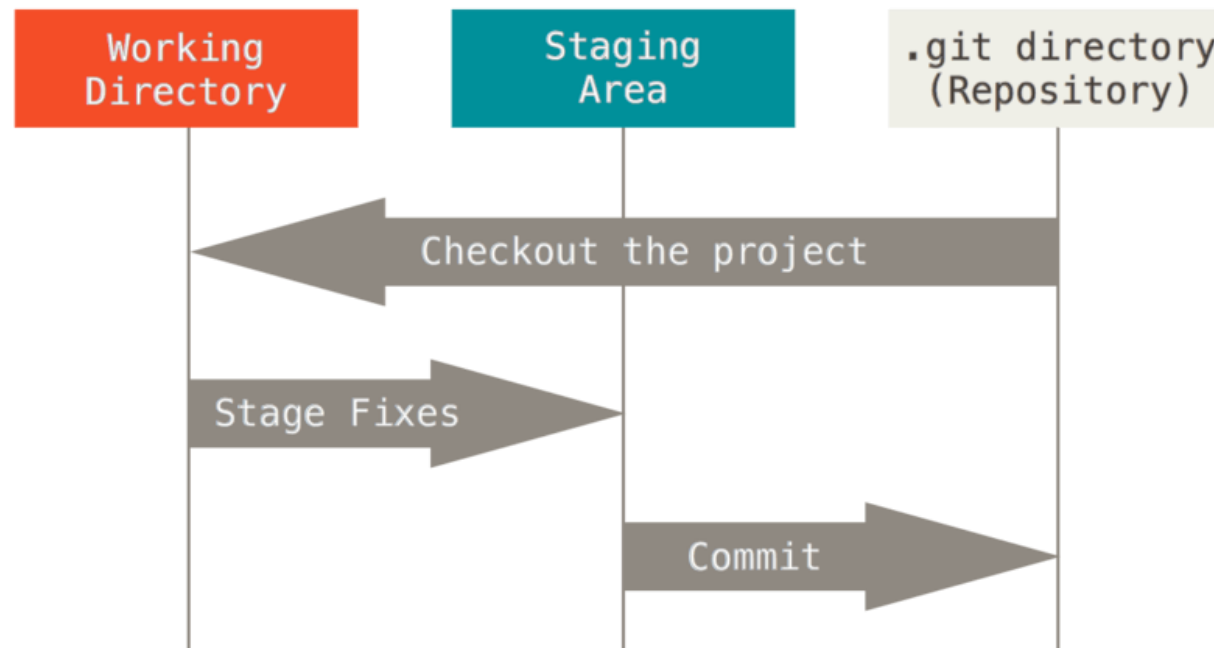
- Guarda los cambios hecho en un repositorio
- Puede ser considerado como una “unidad de trabajo”

Los archivos en un directorio gestionado por Git manejan **tres** estados:

- *Committed*: Guardado en el repositorio
- *Modified*: Archivo ha sido modificado, pero no ha sido guardado en el repositorio
- *Staged*: Archivo modificado para ser guardado en el siguiente *commit* al repositorio
- *Untracked*: Han sido creados recientemente y Git no los gestiona (esto no es un estado valido, para Git este archivo no existe)

Git

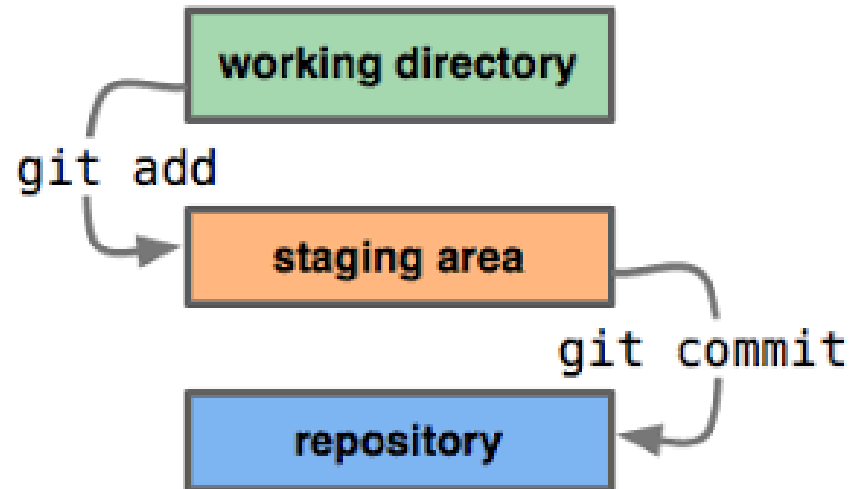
Un proyecto en Git tiene tres secciones: working directory (directorio de trabajo), staging area, git repository.



Git

El flujo de trabajo de Git es como sigue:

- Modificas los archivos en el directorio.
- Los archivos modificados que deseas que estén en un “*commit*”, los envías a la *staging area* con el comando *add*.
- Ejecutar un *commit* usando el comando *commit*, los archivos que estaban en la *staging area* son enviados al repositorio.



Git

Demostración de comandos básicos de Git:

- *git clone* <URL REPOSITORIO>: Copia un repositorio existente en la red
- *git init*: Inicializa un repositorio nuevo
- *git status*: Muestra el estado del directorio de trabajo
- *git add*: Agrega archivos modificados a la *staging area*
 - *git add* puede ser omitido usando el parámetro *-a* en el commit: *git commit -a*
- *git commit*: Envía los cambios al repositorio
- *git log*: Historial de commits

Git – Repositorios remotos

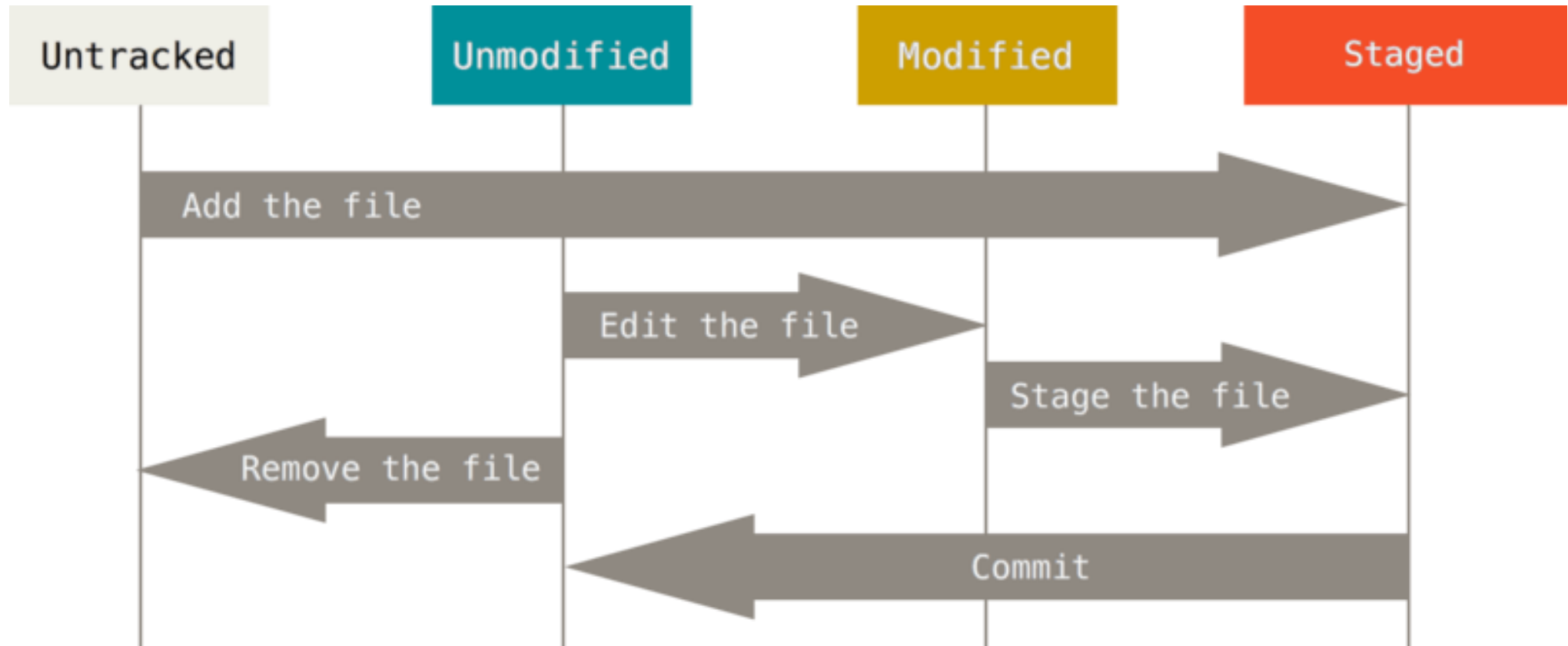
Repositorio remoto: Repositorio en la red, sus contenidos han sido copiados localmente usando *git clone*

Comandos *fetch*, *push* y *pull*

- *git fetch/pull*: obtiene los últimos *commits* del repositorio remoto y los agrega al local
- *git push*: sube los últimos *commits* locales al remoto

Demostración

Git – En resumen



Para la próxima clase

Lectura para la próxima semana:

- Capítulos 1 y 2 de **ProGit** (Scott Chacon, Ben Straub) 2da edición

Tarea

- Crear una cuenta en Bitbucket (bitbucket.org)

Práctica:

- Programación en C y uso de Git

