

Práctica 2.6 Repositorios de Software

(CE2.g)

Qué entregar

Realiza en un PDF las capturas de pantalla de todo el proceso.

1. Repositorios Los repositorios de software son fundamentales para los programadores por varias razones que afectan directamente su eficiencia, colaboración y la calidad de su trabajo.

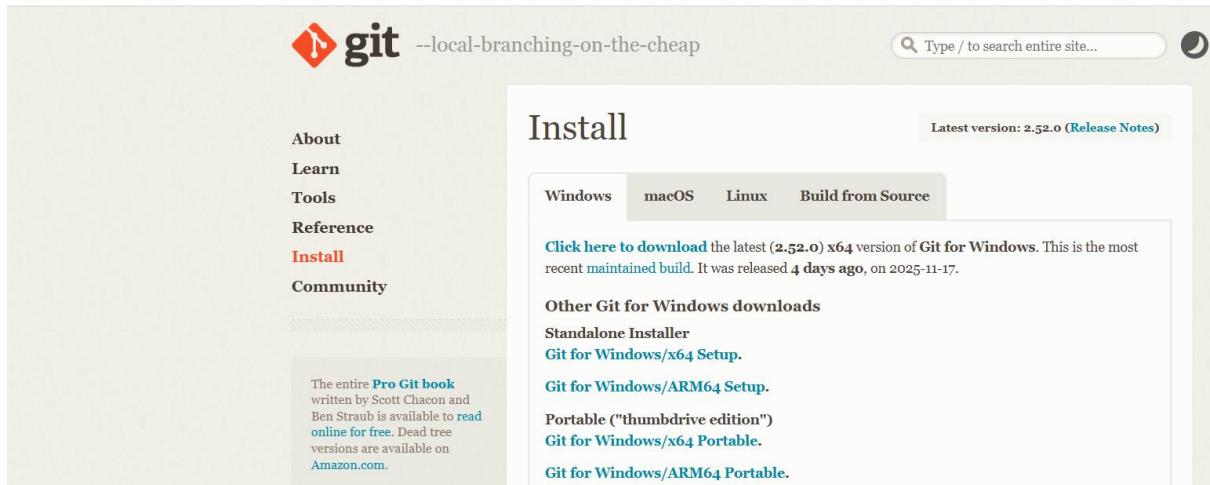
Se encuentran integrados en numerosos IDE, de forma que se facilite su uso durante el desarrollo de aplicaciones.

2. Git Se trata de un sistema de control de versiones distribuido y de código abierto que se utiliza de forma bastante generalizada en el desarrollo de software. Fue creado por Linus Torvalds en 2005 y se ha convertido en una

herramienta fundamental para los desarrolladores. Con Git, cada desarrollador tiene una copia completa del historial de desarrollo, permitiendo el trabajo sin conexión y facilitando la colaboración en equipos distribuidos. Además, facilita la gestión de múltiples líneas de desarrollo de manera concurrente, algo esencial para proyectos grandes y complejos.

2.1. Instalación La mayoría de los sistemas operativos basados en Unix ya incluyen una versión de Git. Sin embargo, en los Sistemas Operativos Windows, deberás proceder a su descarga e instalación desde la página oficial <https://git-scm.com/downloads>

En mi caso yo tengo windos de 64 bits por lo que elijo esta version para instalar



2.2. La terminal de Git

A continuación probamos los diferentes comandos de git: cuando iniciamos git ponemos el comando git y nos salen sus comandos.

```
MINGW64:/c/Users/kaexd
kaexd@Mateooo MINGW64 ~ (master)
$ git
usage: git [-v | --version] [-h | --help] [-C <path>] [-c <name>=<value>]
           [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
           [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--no-lazy
-fetch]
           [--no-optimal-locks] [--no-advice] [--bare] [--git-dir=<path>]
           [--work-tree=<path>] [--namespace=<name>] [--config-env=<name>=<envva
r>]
           <command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
  clone      Clone a repository into a new directory
  init       Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
  add        Add file contents to the index
  mv         Move or rename a file, a directory, or a symlink
  restore    Restore working tree files
  rm         Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)
  bisect    Use binary search to find the commit that introduced a bug
  diff      Show changes between commits, commit and working tree, etc
  grep      Print lines matching a pattern
  log       Show commit logs
  show      Show various types of objects
  status    Show the working tree status

grow, mark and tweak your common history
  backfill  Download missing objects in a partial clone
  branch    List, create, or delete branches
  commit    Record changes to the repository
  merge     Join two or more development histories together
  rebase    Reapply commits on top of another base tip
  reset    Reset current HEAD to the specified state
  switch   Switch branches
  tag      Create, list, delete or verify tags

collaborate (see also: git help workflows)
  fetch    Download objects and refs from another repository
  pull     Fetch from and integrate with another repository or a local branch
  push     Update remote refs along with associated objects

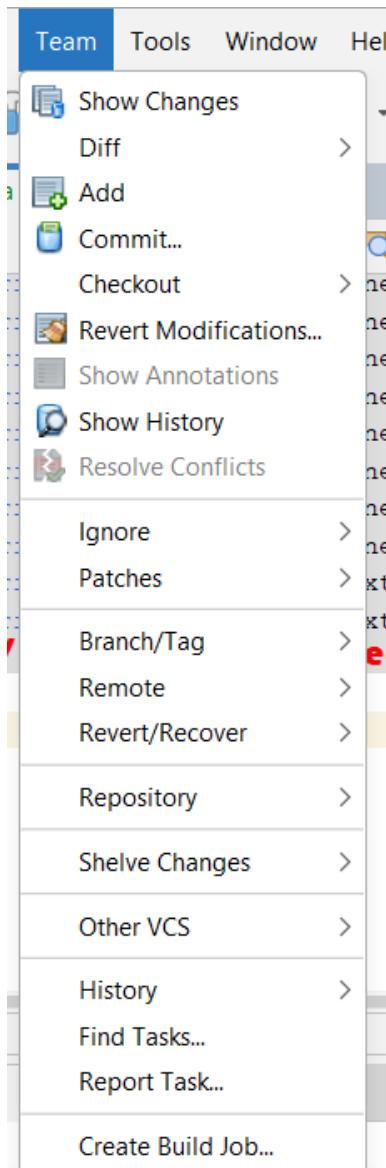
'git help -a' and 'git help -g' list available subcommands and some
concept guides. See 'git help <command>' or 'git help <concept>'
to read about a specific subcommand or concept.
See 'git help git' for an overview of the system.

kaexd@Mateooo MINGW64 ~ (master)
$ Ss
```

3.1. Inicializar un Repositorio: Para crear nuestro repositorio en NetBeans:

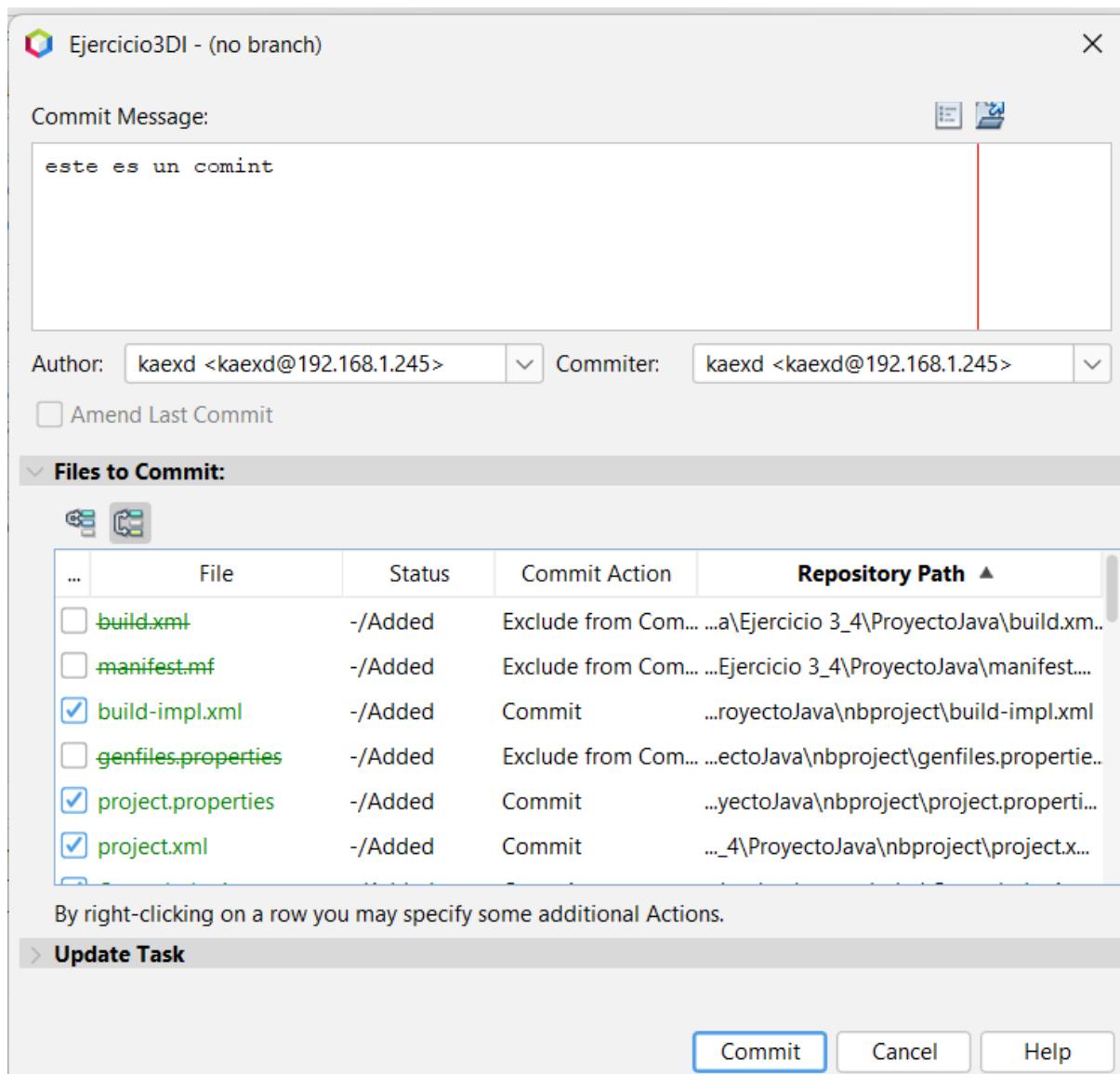
1. Abre o Crea un Proyecto.
2. Habilita Git en el Proyecto: Si estás creando un nuevo proyecto, puedes habilitar la opción "Git" en el asistente de creación de proyectos, o esperar a que ya esté creado .
3. Inicia el Repositorio: Si estás trabajando en un proyecto existente, dirígete a "Team" en la barra de menú y selecciona "Initialize Git"

Repository", o pincha con el botón derecho, y dirígete a "Versioning->Initialize Git Repository".

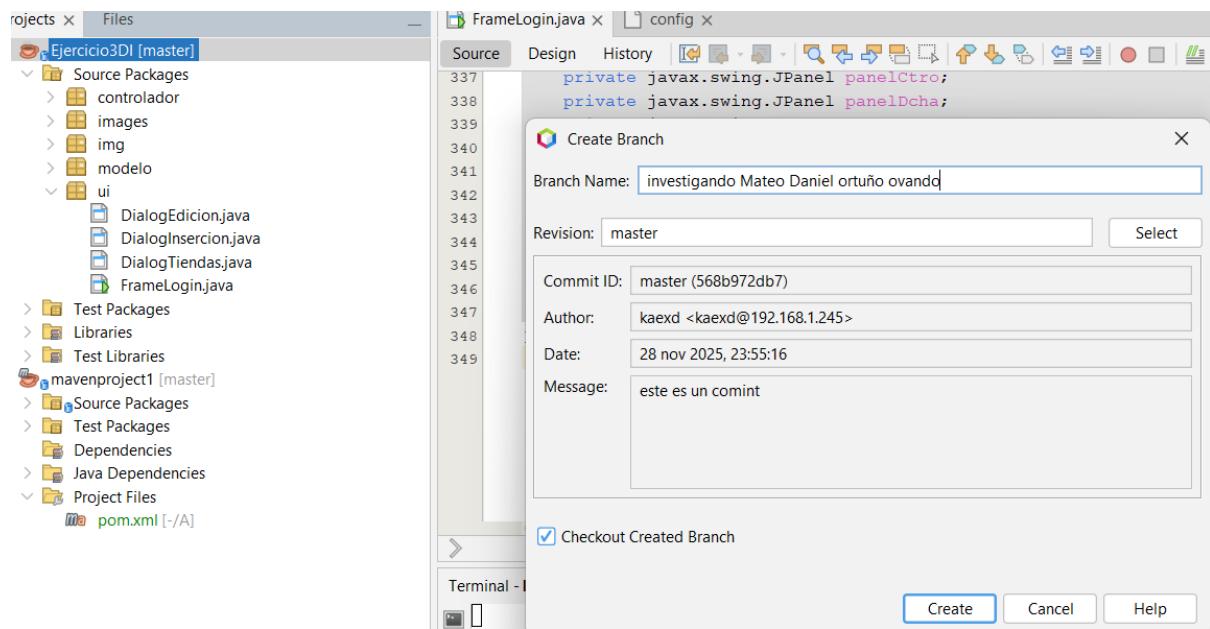


Creamos un repositorio en git,

Luego realizamos un comint



Luego creamos una rama con git branch



Le damos a crear y asi hemos creado una rama en nuestro proyecto.