



ESCUELA DE
INGENIERÍA EN CIENCIAS Y SISTEMAS
FACULTAD DE INGENIERÍA
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



Día, Fecha:	Jueves, 17/08/2023
Hora de inicio:	07:10

Introducción a la programación y computación 1 [F]

José Eduardo Morales García

Contenido



1. **Problemas con DTT**
2. Versionamiento
3. Tareas básicas
4. Herramientas de control de versiones
 - a. Git
 - b. Github, Gitlab
 - c. Git Kraken
5. Colaboración en repositorios remotos
 - a. Clonación de repositorio
 - b. Resolución de conflictos

Problemas con en DTT

Upss!!! Ocurrió un problema con el sistema de Registro y Estadística.
 "Aunque la vida en esta vida es muy corta, tenemos darnos cuenta de que hay mucho por hacer" Alan Turing
 intenta nuevamente. x

Inicio de sesión

Nombre de Usuario: [Verificar](#)

Universidad de San Carlos de Guatemala uses Simio [simulation software](#) under a grant from Simio LLC ([www.simio.com](#)).

Soporte y Asistencia
 soporte@dtl-ecys.org

Sistema de Seguimiento de la Escuela de Ciencias y Sistemas

Activar Windows

Contenido

1. Problemas con DTT
2. **Versionamiento**
3. Tareas básicas
4. Herramientas de control de versiones
 - a. Git
 - b. Github, Gitlab
 - c. Git Kraken
5. Colaboración en repositorios remotos
 - a. Clonación de repositorio
 - b. Resolución de conflictos

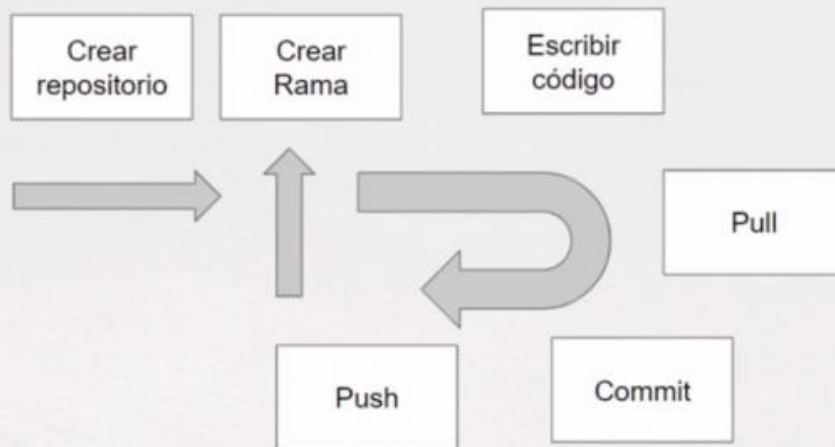
Versionamiento

- ▶ El versionamiento, también conocido como control de versiones, se refiere a la práctica de rastrear y gestionar los cambios realizados en el código fuente, documentos u otros activos digitales a lo largo del tiempo. Es una parte esencial del desarrollo de software y otras disciplinas en las que se crean y modifican archivos de manera colaborativa.
- ▶ El propósito principal del versionamiento es permitir a los equipos de desarrollo trabajar de manera conjunta en proyectos, mantener un historial completo de cambios y facilitar la colaboración mientras se desarrolla y evoluciona un producto.

Características

- Historial de cambios
- Ramificaciones y fusiones
- Revertir cambios
- Trabajo en equipo
- Documentación y seguimiento
- Gestión de lanzamientos

Flujo de trabajo

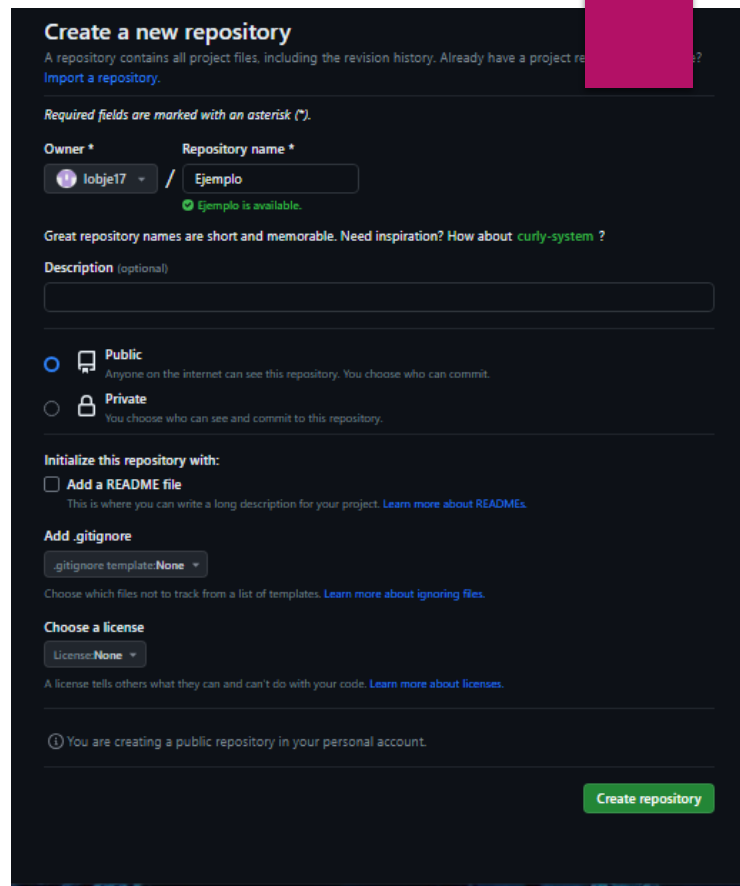


Contenido

1. Problemas con DTT
2. Versionamiento
3. **Tareas básicas**
4. Herramientas de control de versiones
 - a. Git
 - b. Github, Gitlab
 - c. Git Kraken
5. Colaboración en repositorios remotos
 - a. Clonación de repositorio
 - b. Resolución de conflictos

Tareas básicas

- ▶ Creación de repositorio
- ▶ Commit
- ▶ Creación de ramas





Create a new repository

A repository contains all project files, including the revision history. Already have a project repository? [Import a repository.](#)

Required fields are marked with an asterisk ().*


Owner * **Repository name ***


 lobjel7 /

 **Ejemplo is available.**

Great repository names are short and memorable. Need inspiration? How about [curly-system](#) ?

Description (optional)

☒  **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**
You choose who can see and commit to this repository.

Initialize this repository with:


☐ **Add a README file**
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)


 You are creating a public repository in your personal account.

[Create repository](#)

Tareas básicas

- ▶ Creación de repositorio
- ▶ Commit
- ▶ Creación de ramas

Your work / Projects / New project **Create blank project**



Create blank project

Create a blank project to store your files, plan your work, and collaborate on code, among other things.

Project name

Must start with a lowercase or uppercase letter, digit, emoji, or underscore. Can also contain dots, pluses, dashes, or spaces.

Project URL

<https://gitlab.com/> / **Project slug**

Want to organize several dependent projects under the same namespace? [Create a group](#).

Project deployment target (optional)

Visibility Level ⓘ

☒ **Private**
Project access must be granted explicitly to each user. If this project is part of a group, access is granted to members of the group.

☐ **Public**
The project can be accessed without any authentication.

Project Configuration

☒ **Initialize repository with a README**
Allows you to immediately clone this project's repository. Skip this if you plan to push up an existing repository.

☐ **Enable Static Application Security Testing (SAST)**
Analyze your source code for known security vulnerabilities. [Learn more](#).

Tareas básicas

- ▶ Creación de repositorio
- ▶ **Commit**
- ▶ Creación de ramas

Working directory

- ▶ Revisar el estado
- ▶ Agregar cambios al área de preparación
- ▶ Hacer el commit
- ▶ Enviar al repositorio

Tareas básicas

- ▶ Creación de repositorio
- ▶ **Commit**
- ▶ Creación de ramas

```
ricma@DESKTOP-SE76K4T MINGW64 ~/Desktop/PruebaGitHub2
$ git init
Initialized empty Git repository in C:/Users/ricma/Desktop/PruebaGitHub2/.git/

ricma@DESKTOP-SE76K4T MINGW64 ~/Desktop/PruebaGitHub2 (master)
$ git add .

ricma@DESKTOP-SE76K4T MINGW64 ~/Desktop/PruebaGitHub2 (master)
$ git commit -m "Pribando este repositorio"
[master (root-commit) f2e4113] Pribando este repositorio
4 files changed, 32 insertions(+)
create mode 100644 .classpath
create mode 100644 .project
create mode 100644 bin/Operaciones.class
create mode 100644 src/Operaciones.java

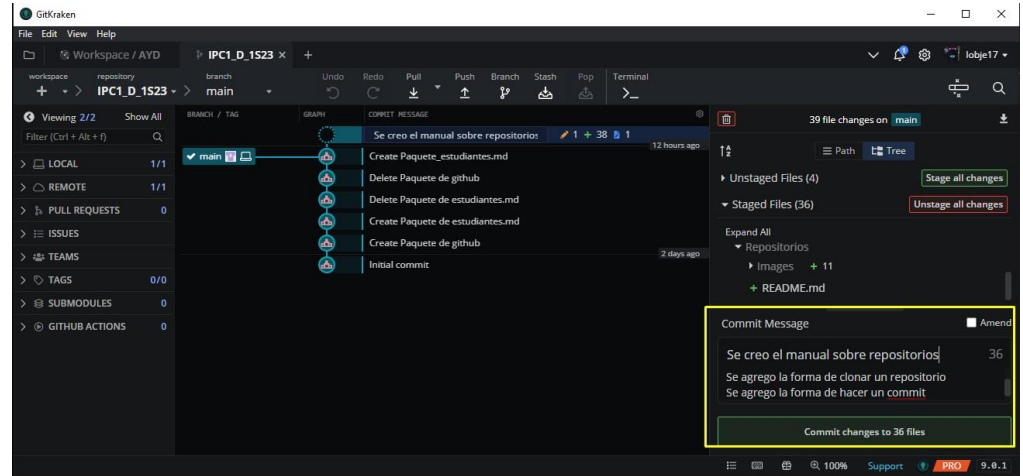
ricma@DESKTOP-SE76K4T MINGW64 ~/Desktop/PruebaGitHub2 (master)
$ git remote add origin https://github.com/rickmanzam/Aprendiendo-Git.git

ricma@DESKTOP-SE76K4T MINGW64 ~/Desktop/PruebaGitHub2 (master)
$ git log --oneline
f2e4113 (HEAD -> master) Pribando este repositorio

ricma@DESKTOP-SE76K4T MINGW64 ~/Desktop/PruebaGitHub2 (master)
$ |
```

Tareas básicas

- ▶ Creación de repositorio
- ▶ **Commit**
- ▶ Creación de ramas



Tareas básicas

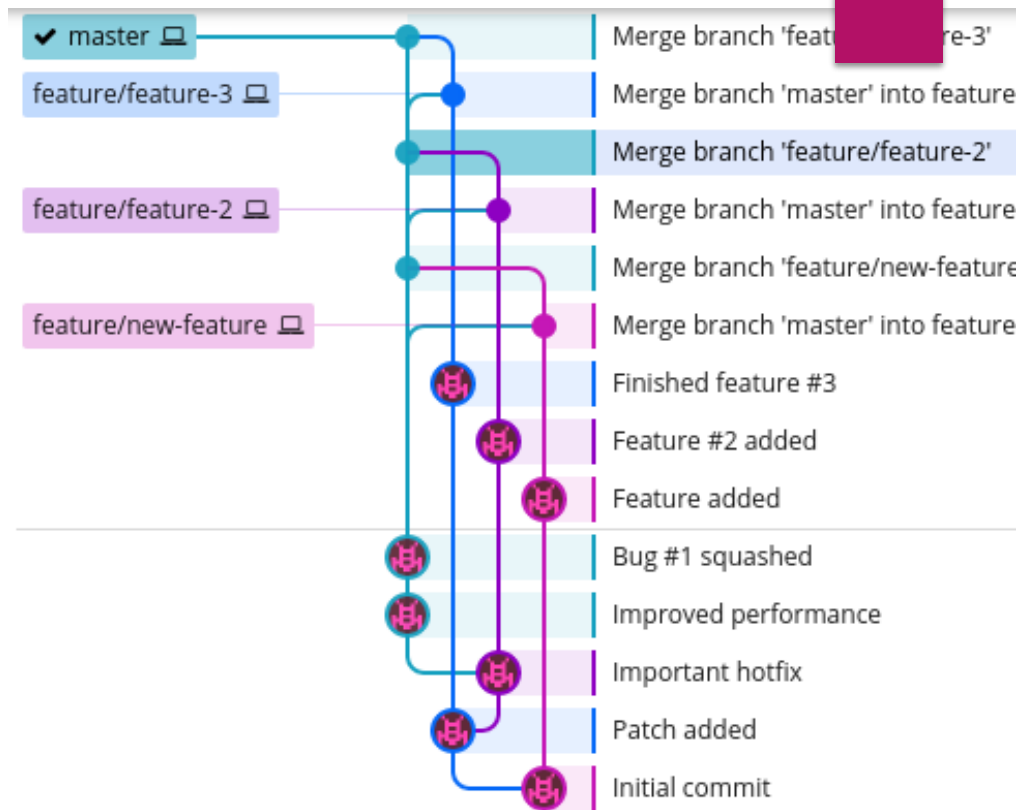
- Creación de repositorio
- **Commit**
- Creación de ramas

The screenshot shows the VS Code interface with the following components and annotations:

- Changes View:** A box highlights the 'Changes' tab and the 'Impresorajs' file, with an arrow pointing to the 'Eliminado' (Deleted) label.
- Diff View:** Shows the code changes for 'Impresorajs'.
 - A red line indicates a deletion: `-const URL_PLUGIN = "http://localhost:8000";` (labeled 'Eliminado').
 - A green line indicates an addition: `+const URL_PLUGIN = "http://localhost:5000";` (labeled 'Agregado').
 - Other changes are labeled 'Archivos modificados' (Modified files).
- Commit Message:** A box highlights the commit message input area, with arrows pointing to 'Descripción corta' (Short description) and 'Descripción larga' (Long description).
- Commit Button:** A box highlights the 'Commit to master' button, with an arrow pointing to 'Guardar cambios' (Save changes).

Tareas básicas

- ▶ Creación de repositorio
- ▶ Commit
- ▶ Creación de ramas

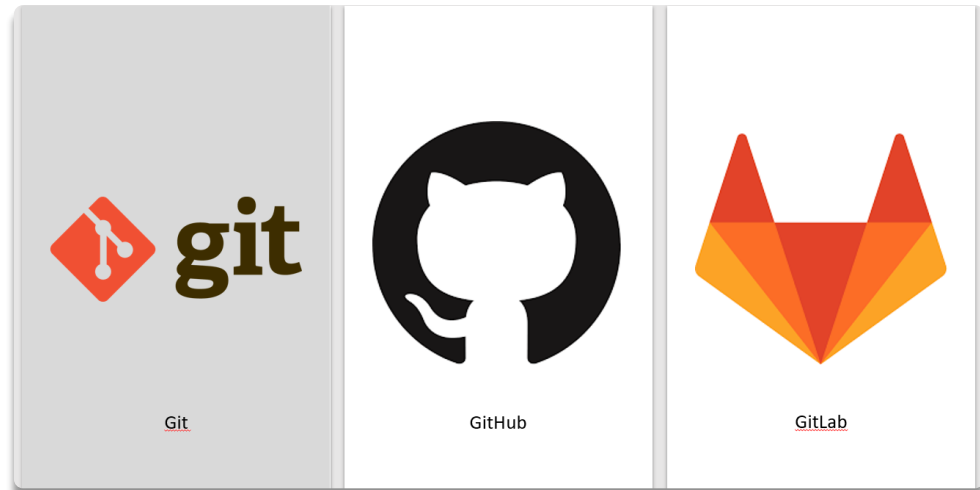


Contenido

1. Problemas con DTT
2. Versionamiento
3. Tareas básicas
4. **Herramientas de control de versiones**
 - a. Git
 - b. Github, Gitlab
 - c. Git Kraken
5. Colaboración en repositorios remotos
 - a. Clonación de repositorio
 - b. Resolución de conflictos

¿Qué es Git?

- ▶ Git es un sistema de control de versiones de software utilizado para llevar un registro de las modificaciones realizadas en los archivos de un proyecto y coordinar el trabajo de múltiples personas en ese proyecto. Git permite trabajar de manera descentralizada, lo que significa que cada usuario tiene una copia completa del repositorio en su equipo, lo que facilita la colaboración y el control de versiones.



ejemplo básico de cómo usar Git:

- a) Instalar Git en su sistema operativo
- b) Inicializar un repositorio local en un directorio utilizando el comando **git init**
- c) Agregar archivos al área de seguimiento usando **git add**
- d) Realizar una "commit" de los cambios con un mensaje de descripción usando **git commit -m "commit message"**
- e) Conectar el repositorio local con un repositorio remoto, como GitHub, usando **git remote add origin <url>**
- f) Subir los cambios al repositorio remoto usando **git push origin master**

GitHub

- GitHub es una plataforma web que ofrece alojamiento para proyectos de software y control de versiones utilizando Git. También ofrece herramientas colaborativas para desarrolladores, como seguimiento de problemas, gestión de tareas y wikis.



GITHUB

github / codespaces-demo

Watch 1 Fork 1 Starred 7.4k

<> Code Issues Pull requests Actions Wiki Security Insights

main Branches Tags

Go to file Add file Code

mona Update README.md 546f1a8 2 days ago 13 commits

.devcontainer	commit message	2 days ago
.github	commit message	2 days ago
README.md	commit message	2 days ago
package.json	commit message	2 days ago
devcontainer-local-run.sh	commit message	2 days ago

README.md

Hello world.

About

A package from Mona, for GitHub.

Readme MIT license Code of conduct 1 stars 1 watching 1 forks

Packages 1

octopackage/octopackage

Contributors 1

mona Octocat

GitLab

- GitLab es una plataforma integral de desarrollo de software que proporciona un conjunto de herramientas para la gestión del ciclo de vida de las aplicaciones y el control de versiones basado en Git. GitLab permite a los equipos de desarrollo colaborar, gestionar proyectos, rastrear problemas, automatizar procesos y desplegar aplicaciones de manera eficiente. Es tanto una herramienta de control de versiones como una plataforma de desarrollo y colaboración.



GitLab

GITLAB

The screenshot displays the GitLab web interface for the 'gitlab' project. The left sidebar contains navigation links: Project information, Repository, Issues (39,264), Merge requests (1,208), CI/CD, Deployments, Monitor, Packages & Registries, Analytics, and Snippets. The main content area shows the project details for 'gitlab' (Project ID: 278964), including statistics like 249,102 Commits, 8,152 Branches, 1,976 Tags, 8 GB Files, 66.4 TB Storage, and 115 Releases. A description states: 'GitLab is an open source end-to-end software development platform with built-in version control, issue tracking, code review, CI/CD, and more. Self-host GitLab on your own servers, in a container, or on a cloud provider.' Below this is a progress bar for CI/CD coverage. A recent commit is highlighted: 'Merge branch "jv-gitally-pack-objects-feature-flag" into "master"' by Sean McGivern, authored 20 minutes ago. At the bottom, a table lists project files and their last commit details.

Name	Last commit	Last update
github	Rename GitLab CE to FOSS in GitHub issue ...	1 year ago
gitlab	Merge branch 'always-run-full-migration-sp...	2 hours ago
zheia	Revert "Move host authorization to gitp...	10 months ago
app	Merge branch 'ph/324166/hewReadyToMe...	1 hour ago
bin	Skip hooks when pushing security branche...	3 weeks ago
builds	Add missing build/ folder to fix backup tests	5 years ago
changelogs	Remove wrapper methods from Gitlab:Dat...	4 weeks ago
config	Merge branch 'jv-gitally-pack-objects-featur...	20 minutes ago
danger	Clarify the milestone requirements for sec...	4 days ago
data/whats_new	Create 14.2 in-app top 7 product highlights	1 day ago
db	Merge branch 'mc-remove-column-from-pr...	3 hours ago
doc	Merge branch 'docs-fix-dag-detail-2' into '...	32 minutes ago

GitHub Desktop

- GitHub Desktop es una aplicación de escritorio gratuita que permite a los desarrolladores interactuar con repositorios de Git de forma más fácil y amigable que a través de la línea de comandos. Ofrece una interfaz gráfica para realizar tareas comunes de Git, como hacer commits, fusionar ramas, resolver conflictos, y navegar por el historial del repositorio. GitHub Desktop es una opción popular para aquellos que quieren comenzar a utilizar Git de forma más accesible y para aquellos que buscan una herramienta más eficiente que la línea de comandos.

FileEditViewRepositoryBranchHelp

Current repository
Demo_Project_Repo_VN

Current branch
master

Fetch origin
Last fetched a minute ago

Changes

History

0 changed files

Summary (required)

Description

1+

Commit to master

No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.

Open the repository in your external editor
Select your editor in [Options](#)
Repository menu or **Ctrl Shift A**

Open in Atom

View the files of your repository in Explorer
Repository menu or **Ctrl Shift F**

Show in Explorer

Open the repository page on GitHub in your browser
Repository menu or **Ctrl Shift G**

View on GitHub

GitKraken

- GitKraken es una herramienta de software de escritorio y navegador que proporciona una interfaz gráfica para trabajar con repositorios de Git. Ofrece funciones como control de versiones, colaboración en tiempo real, visualización de ramas y historial de commits, y gestión de tareas y problemas. GitKraken es utilizado por desarrolladores individuales y equipos para mejorar la eficiencia y la productividad en el proceso de desarrollo de software.

GitKraken

File Edit View Help

Vester master

Viewing 2/2 Show All

LOCAL 1/1

master 5x

REMOTE 1/1

origin 1/1

master

TAGS 0/0

SUBMODULES 0

master

master

Merge pull request #7 from equelin/feature-disconnect-cdrom 1 hour ago

Added specific tests for CDROM connected to ISO or Host Device 6 days ago

Merge pull request #6 from equelin/typo-readme

Added missing ; and) in examples

Added test against connected CDROM

Update template and fix error in config a week ago

Update example media image

Merge pull request #5 from WahlNetwork/build-params

Update README.md

Update README.md

Update config file with helpful tips

Add additional params to all tests

Begin adding proper params

Merge pull request #4 from equelin/feature-delete-snapshot 2 weeks ago

Added \$global:config.vm parameter in Config.ps1

Added test Delete-Snapshot

Merge pull request #3 from WahlNetwork/refactor

Create Syslog test 3 weeks ago

Create SSH test

Add error actions to fail tests properly

Create NFS test and add option to enable remediation logic

Update README.md

Merge branch 'refactor' of https://github.com/WahlNetwork/lab-config into refactor

Add example image

Update README.md

Add DNS tests

Embed remediation into tests; create NTP and DRS tests

search commits

Merge pull request #5 from WahlNetwork/build-params

Build params

Chris Wahl authored: 7/14/2016 @ 8:48 PM commit: 1231f7 parent: 7baca06b31b

0 added 0 deleted 9 modified

README.md

Tests/Config.ps1

Tests/Delete-Snapshot.Tests.ps1

Tests/Update-DNS.Tests.ps1

Tests/Update-DRS.Tests.ps1

Tests/Update-NFS.Tests.ps1

Tests/Update-NTP.Tests.ps1

Tests/Update-SSH.Tests.ps1

Tests/Update-Syslog.Tests.ps1

100% Provide Feedback

Contenido

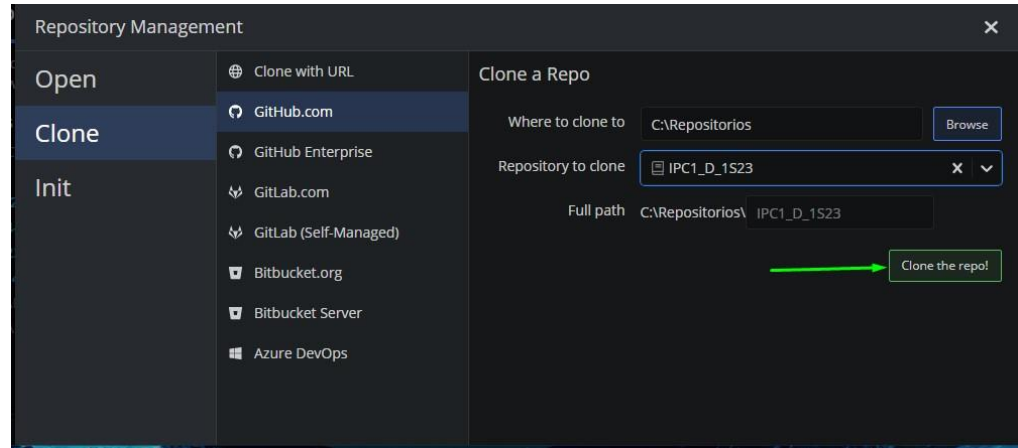
1. Problemas con DTT
2. Versionamiento
3. Tareas básicas
4. Herramientas de control de versiones
 - a. Git
 - b. Github, Gitlab
 - c. Git Kraken
5. **Colaboración en repositorios remotos**
 - a. Clonación de repositorio
 - b. Resolución de conflictos

Colaboración en repositorios remotos

La colaboración en repositorios remotos es una parte esencial del desarrollo de software moderno. Los repositorios remotos permiten a los equipos trabajar juntos en un proyecto compartido, facilitando la colaboración en tiempo real, la revisión de código y la gestión de cambios de manera distribuida.

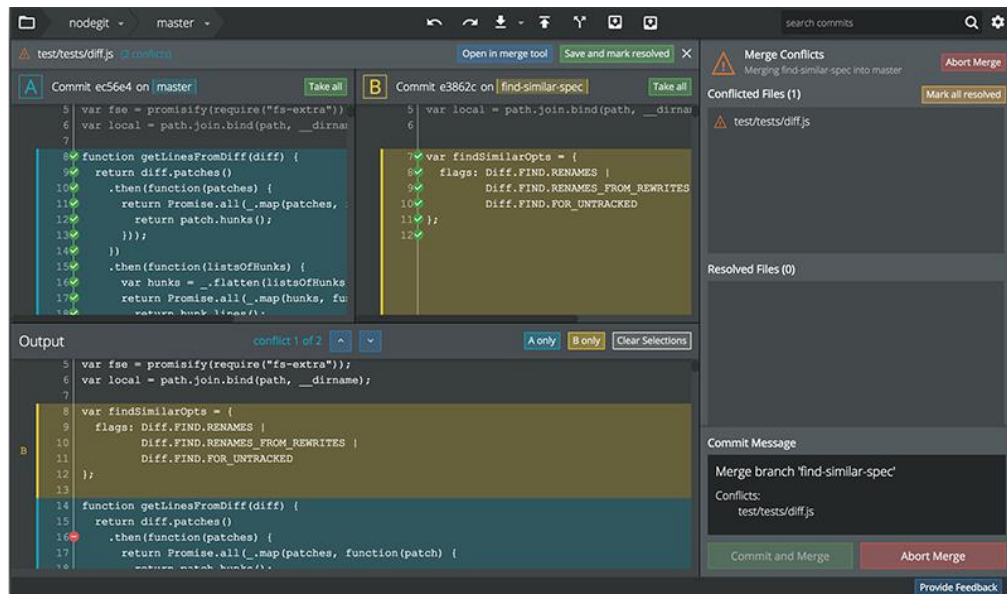
Clonación de Repositorios Remotos

Para colaborar en un proyecto, los miembros del equipo pueden clonar (copiar) el repositorio remoto en sus propias máquinas locales usando comandos u otras herramientas. Esto crea una copia completa del repositorio en sus sistemas.



Resolución de conflictos

En ocasiones, pueden surgir conflictos cuando se intenta fusionar cambios de diferentes ramas. Estos conflictos deben resolverse manualmente, generalmente mediante la edición del código para combinar las diferencias.



Dudas y Preguntas





Parte practica