

Práctica GIT

Nombre: Esnor Noel Enrique Vaca Moreno

Creación del Proyecto en GitHub

The screenshot shows the GitHub 'Create a new repository' page. The page title is 'Create a new repository'. Below the title, it says 'A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)'. There is a 'Repository template' section with a 'No template' button. The 'Owner' is 'enevaca' and the 'Repository name' is 'practica-git', which is marked as available. The 'Description (optional)' field is empty. The 'Public' checkbox is selected, indicating the repository will be public. The 'Initialize this repository with:' section has an 'Add a README file' checkbox. The 'Add .gitignore' section has a '.gitignore template: None' dropdown. The 'Choose a license' section has a 'License: None' dropdown. At the bottom, there is a 'Create repository' button.

The screenshot shows the GitHub repository page for 'enevaca/practica-git'. The page header includes the repository name and a 'Public' label. The main content area shows a list of files: 'PracticaGitHub.docx', 'README.md', '~\$acticaGitHub.docx', and '~WRL0005.tmp'. The 'README.md' file is selected, showing its content: 'practica-git' and 'Para rebase Pare revert Para cherry-pick Para cherry-pick2'. The right sidebar contains sections for 'About', 'Releases', and 'Packages'. A user profile dropdown menu is open on the right, showing options like 'Your profile', 'Your repositories', 'Your projects', 'Your stars', 'Your gists', 'Upgrade', 'Try Enterprise', 'Feature preview', 'Help', 'Settings', and 'Sign out'.

Nombre: Esnor Noel Enrique Vaca Moreno

Repaso de GIT

Integración, Entrega y Despliegue continuo - MDEISV2E1

Configuración inicial de GIT:

```
MINGW64/d/mdeis/11. Integracion, Entrega y Despliegue Continuo/practica-git
Noel@DESKTOP-FJ0CUOU MINGW64 /d/mdeis/11. Integracion, Entrega y Despliegue Continuo/practica-git
$ echo "# practica-git" >> README.md

Noel@DESKTOP-FJ0CUOU MINGW64 /d/mdeis/11. Integracion, Entrega y Despliegue Continuo/practica-git
$ git init
Initialized empty Git repository in D:/mdeis/11. Integracion, Entrega y Despliegue Continuo/practica-git/.git/

Noel@DESKTOP-FJ0CUOU MINGW64 /d/mdeis/11. Integracion, Entrega y Despliegue Continuo/practica-git (main)
$ git add README.md

Noel@DESKTOP-FJ0CUOU MINGW64 /d/mdeis/11. Integracion, Entrega y Despliegue Continuo/practica-git (main)
$ git commit -m "first commit"
[main (root-commit) 35e13fd] first commit
1 file changed, 1 insertion(+)
create mode 100644 README.md

Noel@DESKTOP-FJ0CUOU MINGW64 /d/mdeis/11. Integracion, Entrega y Despliegue Continuo/practica-git (main)
$ git branch -M main

Noel@DESKTOP-FJ0CUOU MINGW64 /d/mdeis/11. Integracion, Entrega y Despliegue Continuo/practica-git (main)
$ git remote add origin https://github.com/enevaca/practica-git.git

Noel@DESKTOP-FJ0CUOU MINGW64 /d/mdeis/11. Integracion, Entrega y Despliegue Continuo/practica-git (main)
$ git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 229 bytes | 229.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/enevaca/practica-git.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

Noel@DESKTOP-FJ0CUOU MINGW64 /d/mdeis/11. Integracion, Entrega y Despliegue Continuo/practica-git (main)
$ |
```

Tareas Completadas



1. Introducción a los commits

```
MINGW64; d/mdeis/11. Integracion, Entrega y Despliegue Continuo/practica-git (main)
Noel@DESKTOP-FJ0CUOU MINGW64 /d/mdeis/11. Integracion, Entrega y Despliegue Continuo/practica-git (main)
$ git add .

Noel@DESKTOP-FJ0CUOU MINGW64 /d/mdeis/11. Integracion, Entrega y Despliegue Continuo/practica-git (main)
$ git commit -m 'Introducción a los commits'
[main cde9cb6] Introducción a los commits
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 PracticaGitHub.docx

Noel@DESKTOP-FJ0CUOU MINGW64 /d/mdeis/11. Integracion, Entrega y Despliegue Continuo/practica-git (main)
$ git add .

Noel@DESKTOP-FJ0CUOU MINGW64 /d/mdeis/11. Integracion, Entrega y Despliegue Continuo/practica-git (main)
$ git commit -m 'Introducción a los commits'
[main 588e750] Introducción a los commits
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 ~$acticaGitHub.docx

Noel@DESKTOP-FJ0CUOU MINGW64 /d/mdeis/11. Integracion, Entrega y Despliegue Continuo/practica-git (main)
$ |
```

2. Ramas en GIT

```
Noel@DESKTOP-FJ0CUOU MINGW64 /d/mdeis/11. Integracion, Entrega y Despliegue Continuo/practica-git (main)
$ git branch bugFix

Noel@DESKTOP-FJ0CUOU MINGW64 /d/mdeis/11. Integracion, Entrega y Despliegue Continuo/practica-git (main)
$ git checkout bugFix
Switched to branch 'bugFix'

Noel@DESKTOP-FJ0CUOU MINGW64 /d/mdeis/11. Integracion, Entrega y Despliegue Continuo/practica-git (bugFix)
$ |
```

3. Haciendo merge en ramas

```
Noel@DESKTOP-FJ0CUOU MINGW64 /d/mdeis/11. Integracion, Entrega y Despliegue Continuo/practica-git (bugFix)
$ git add .

Noel@DESKTOP-FJ0CUOU MINGW64 /d/mdeis/11. Integracion, Entrega y Despliegue Continuo/practica-git (bugFix)
$ git commit -m 'Commit en bugFix para merge'
[bugFix f216101] Commit en bugFix para merge
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 ~$acticaGitHub.docx

Noel@DESKTOP-FJ0CUOU MINGW64 /d/mdeis/11. Integracion, Entrega y Despliegue Continuo/practica-git (bugFix)
$ git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 3 commits.
(use "git push" to publish your local commits)

Noel@DESKTOP-FJ0CUOU MINGW64 /d/mdeis/11. Integracion, Entrega y Despliegue Continuo/practica-git (main)
$ git merge bugFix
Updating 6d28bba..f216101
Fast-forward
 PracticaGitHub.docx | Bin 303976 -> 339960 bytes
 ~$acticaGitHub.docx | Bin 0 -> 162 bytes
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 ~$acticaGitHub.docx

Noel@DESKTOP-FJ0CUOU MINGW64 /d/mdeis/11. Integracion, Entrega y Despliegue Continuo/practica-git (main)
$ |
```

4. Introducción a rebase

```
MINGW64:/d/mdeis/11. Integracion, Entrega y Despliegue Continuo/practica-git (main)
$ git checkout bugFix
Switched to branch 'bugFix'

MINGW64:/d/mdeis/11. Integracion, Entrega y Despliegue Continuo/practica-git (bugFix)
$ git add .

MINGW64:/d/mdeis/11. Integracion, Entrega y Despliegue Continuo/practica-git (bugFix)
$ git commit -m 'Modificación en bugFix para rebase'
[bugFix 1757964] Modificación en bugFix para rebase
1 file changed, 1 insertion(+)

MINGW64:/d/mdeis/11. Integracion, Entrega y Despliegue Continuo/practica-git (bugFix)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

MINGW64:/d/mdeis/11. Integracion, Entrega y Despliegue Continuo/practica-git (main)
$ git add .

MINGW64:/d/mdeis/11. Integracion, Entrega y Despliegue Continuo/practica-git (main)
$ git commit -m 'Modificación en main para rebase'
[main 210de8b] Modificación en main para rebase
2 files changed, 0 insertions(+), 0 deletions(-)
delete mode 100644 ~$acticaGitHub.docx

MINGW64:/d/mdeis/11. Integracion, Entrega y Despliegue Continuo/practica-git (main)
$ git checkout bugFix
Switched to branch 'bugFix'

MINGW64:/d/mdeis/11. Integracion, Entrega y Despliegue Continuo/practica-git (bugFix)
$ git rebase main
Successfully rebased and updated refs/heads/bugFix.

MINGW64:/d/mdeis/11. Integracion, Entrega y Despliegue Continuo/practica-git (bugFix)
$ |
```

5. Desatachar HEAD

```
MINGW64:/d/mdeis/11. Integracion, Entrega y Despliegue Continuo/practica-git (main)
$ git log -1 --stat
commit 2afd9e127a43481408f54054a980615db672d47d (HEAD -> main, origin/main, bugFix)
Author: Noel Vaca <noel.v.m.2501@gmail.com>
Date: Sun Nov 13 23:48:59 2022 -0400

    Finalizando paso rebase

    PracticaGitHub.docx | Bin 446802 -> 587355 bytes
    1 file changed, 0 insertions(+), 0 deletions(-)

MINGW64:/d/mdeis/11. Integracion, Entrega y Despliegue Continuo/practica-git (main)
$ git checkout 2afd9e127a43481408f54054a980615db672d47d
Note: switching to '2afd9e127a43481408f54054a980615db672d47d'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

    git switch -c <new-branch-name>

Or undo this operation with:

    git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at 2afd9e1 Finalizando paso rebase

MINGW64:/d/mdeis/11. Integracion, Entrega y Despliegue Continuo/practica-git ((2afd9e1...))
$ |
```

6. Referencias alternativas ^

```
Noel@DESKTOP-FJ0CUOU MINGW64 /d/mdeis/11. Integracion, Entrega y Despliegue Continuo/practica-git (main)
$ git checkout bugFix^
Note: switching to 'bugFix^'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

    git switch -c <new-branch-name>

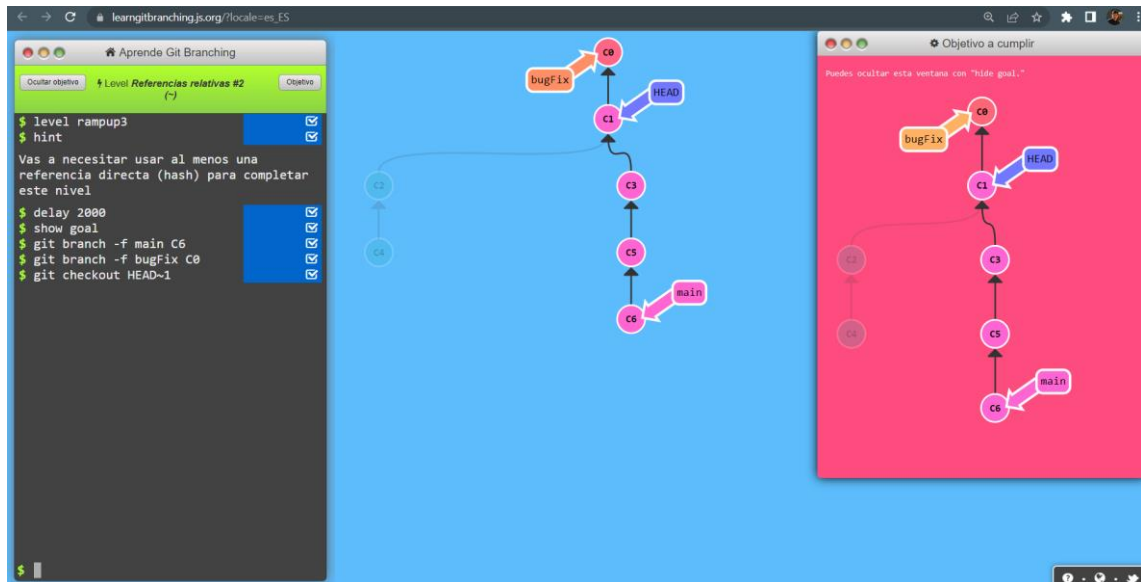
Or undo this operation with:

    git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at a74db3b Modificación en bugFix para rebase
Noel@DESKTOP-FJ0CUOU MINGW64 /d/mdeis/11. Integracion, Entrega y Despliegue Continuo/practica-git ((a74db3b...))
$ |
```

7. Referencias relativas #2 (~)



```
MINGW64:/d/mdeis/11. Integri x + v
Noel@DESKTOP-FJ0CUOU MINGW64 /d/mdeis/11. Integracion, Entrega y Despliegue Continuo/practica-git (main)
$ git checkout HEAD~1
Note: switching to 'HEAD~1'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

    git switch -c <new-branch-name>

Or undo this operation with:

    git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at 885d23f Finalizando desatachar HEAD
Noel@DESKTOP-FJ0CUOU MINGW64 /d/mdeis/11. Integracion, Entrega y Despliegue Continuo/practica-git ((885d23f...))
$ |
```

8. Revirtiendo cambios en GIT

```

Noel@DESKTOP-FJ0CUOU MINGW64 /d/mdeis/11. Integracion, Entrega y Despliegue Continuo/practica-git (main)
$ git add .

Noel@DESKTOP-FJ0CUOU MINGW64 /d/mdeis/11. Integracion, Entrega y Despliegue Continuo/practica-git (main)
$ git commit -m 'Cambio para revert'
[main 13d2373] Cambio para revert
1 file changed, 2 insertions(+), 1 deletion(-)

Noel@DESKTOP-FJ0CUOU MINGW64 /d/mdeis/11. Integracion, Entrega y Despliegue Continuo/practica-git (main)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 327 bytes | 327.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/enevaca/practica-git.git
c2a98a7..13d2373 main -> main

Noel@DESKTOP-FJ0CUOU MINGW64 /d/mdeis/11. Integracion, Entrega y Despliegue Continuo/practica-git (main)
$ git add .

Noel@DESKTOP-FJ0CUOU MINGW64 /d/mdeis/11. Integracion, Entrega y Despliegue Continuo/practica-git (main)
$ git commit -m 'Cambio para reset'
[main 940f8c0] Cambio para reset
1 file changed, 2 insertions(+), 1 deletion(-)

Noel@DESKTOP-FJ0CUOU MINGW64 /d/mdeis/11. Integracion, Entrega y Despliegue Continuo/practica-git (main)
$ git reset HEAD~1

```

```

Noel@DESKTOP-FJ0CUOU MINGW64 /d/mdeis/11. Integracion, Entrega y Despliegue Continuo/practica-git (main)
$ git checkout 13d23736eb5bfff3f30a1c241a21c250604ca8b53
Note: switching to '13d23736eb5bfff3f30a1c241a21c250604ca8b53'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

    git switch -c <new-branch-name>

Or undo this operation with:

    git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at 13d2373 Cambio para revert

Noel@DESKTOP-FJ0CUOU MINGW64 /d/mdeis/11. Integracion, Entrega y Despliegue Continuo/practica-git ((13d2373...))
$ git revert HEAD
[detached HEAD 8633cf1] Revert "Cambio para revert"
1 file changed, 1 insertion(+), 2 deletions(-)

```



9. Introducción a cherry-pick

```

Noel@DESKTOP-FJ0CUOU MINGW64 /d/mdeis/11. Integracion, Entrega y Despliegue Continuo/practica-git (bugFix)
$ git log -2 --stat
commit e2cd67e5204390a2a356f9f156ee5deb5924cccd (HEAD -> bugFix, origin/bugFix)
Author: Noel Vaca <noel.vm.2501@gmail.com>
Date: Mon Nov 14 01:47:36 2022 -0400

    Para cherry-pic 2

    README.md | 3 ++-
    1 file changed, 2 insertions(+), 1 deletion(-)

commit c8629c8497526cd6a6517d02a0b9cf00f820b000
Author: Noel Vaca <noel.vm.2501@gmail.com>
Date: Mon Nov 14 01:44:12 2022 -0400

    Para cherry-pic

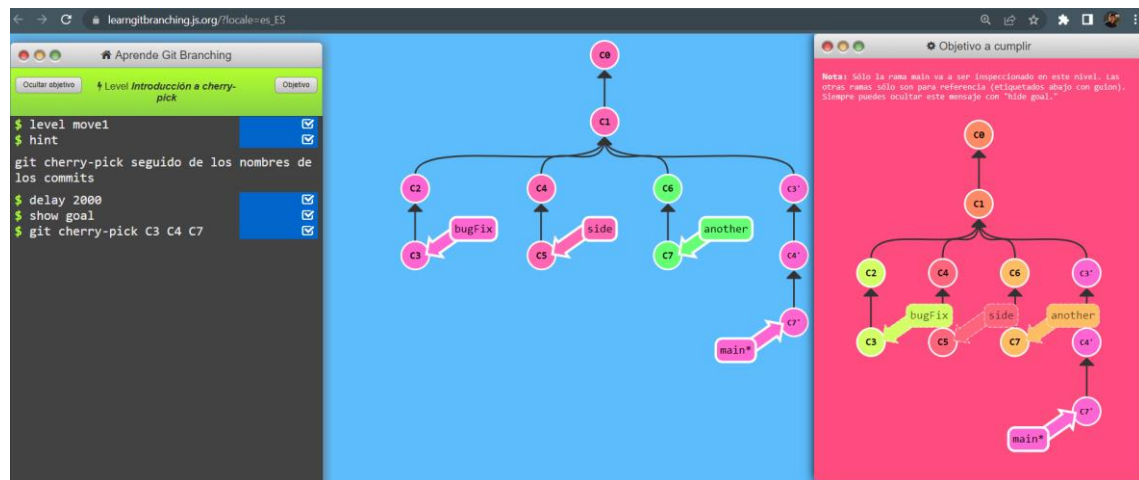
    README.md | 3 ++-
    1 file changed, 2 insertions(+), 1 deletion(-)

Noel@DESKTOP-FJ0CUOU MINGW64 /d/mdeis/11. Integracion, Entrega y Despliegue Continuo/practica-git (bugFix)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

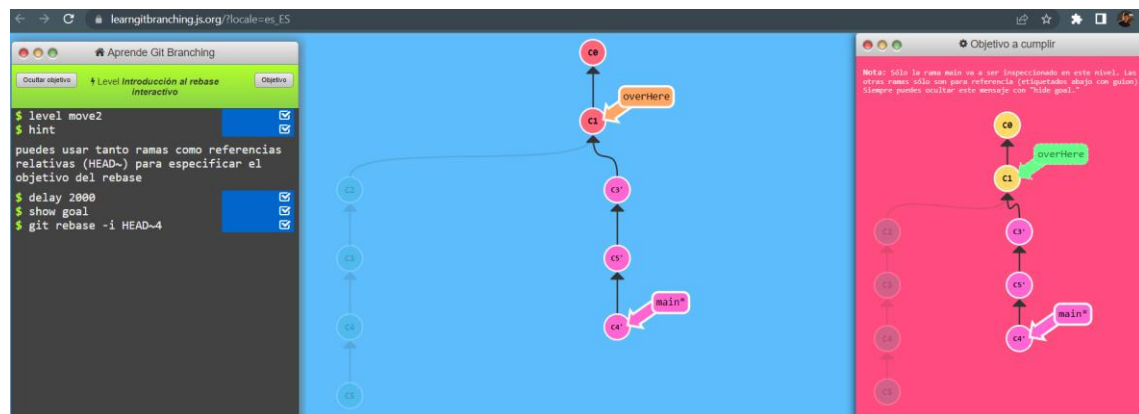
Noel@DESKTOP-FJ0CUOU MINGW64 /d/mdeis/11. Integracion, Entrega y Despliegue Continuo/practica-git (main)
$ git cherry-pick c8629c8497526cd6a6517d02a0b9cf00f820b000 e2cd67e5204390a2a356f9f156ee5deb5924cccd
[main f067abe] Para cherry-pic
Date: Mon Nov 14 01:44:12 2022 -0400
1 file changed, 2 insertions(+), 1 deletion(-)
[main ce5d054] Para cherry-pic 2
Date: Mon Nov 14 01:47:36 2022 -0400
1 file changed, 2 insertions(+), 1 deletion(-)

Noel@DESKTOP-FJ0CUOU MINGW64 /d/mdeis/11. Integracion, Entrega y Despliegue Continuo/practica-git (main)
$

```



10. Introducción al rebase interactivo




```

MINGW64:/d/mdeis/11. Integraci...
pick adf77325 Para rebase interactivo commit 1
pick 967c551 Para rebase interactivo commit 1
pick c1ce6d3 Para rebase interactivo commit 1

# Rebase 7ad5fe2..967c551 onto 7ad5fe2 (3 commands)
#
# Commands:
# p, pick <commit> = use commit
# r, reword <commit> = use commit, but edit the commit message
# e, edit <commit> = use commit, but stop for amending
# s, squash <commit> = use commit, but meld into previous commit
# f, fixup [-C | -c] <commit> = like "squash" but keep only the previous
#                               commit's log message, unless -C is used, in which case
#                               keep only this commit's message; -c is same as -C but
#                               opens the editor
# x, exec <command> = run command (the rest of the line) using shell
# b, break = stop here (continue rebase later with 'git rebase --continue')
# d, drop <commit> = remove commit
# l, label <label> = label current HEAD with a name
# t, reset <label> = reset HEAD to a label
# m, merge [-C <commit> | -c <commit>] <label> [# <oneline>]
#     create a merge commit using the original merge commit's
#     message (or the oneline, if no original merge commit was
#     specified); use -c <commit> to reword the commit message
#
# These lines can be re-ordered; they are executed from top to bottom.
#
# If you remove a line here THAT COMMIT WILL BE LOST.
< Entrega y Despliegue Continuo/practica-git/.git/rebase-merge/git-rebase-todo[+] [unix] (18:03 14/11/2022)
-- INSERTAR --

```

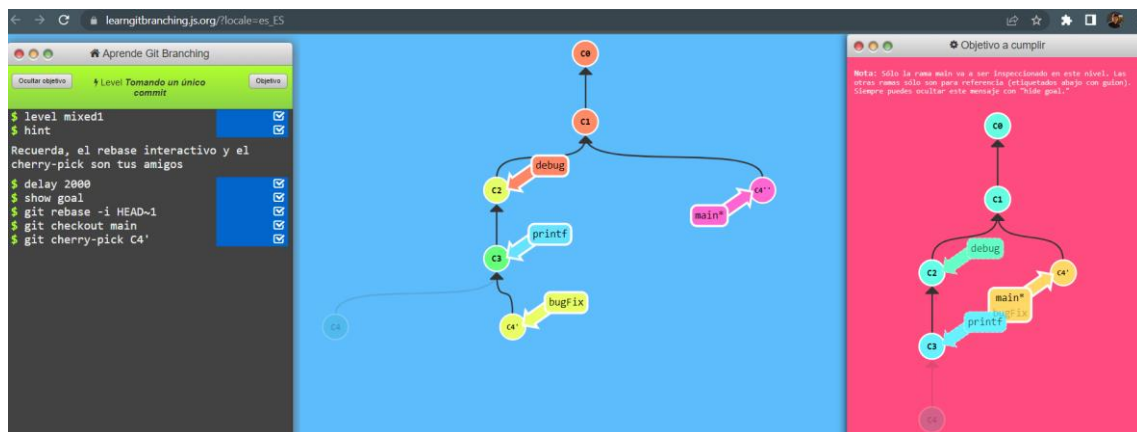
```

Noel@DESKTOP-FJ0CUOU MINGW64 /d/mdeis/11. Integracion, Entrega y Despliegue Continuo/practica-git (main)
$ git rebase -i HEAD~3
Successfully rebased and updated refs/heads/main.

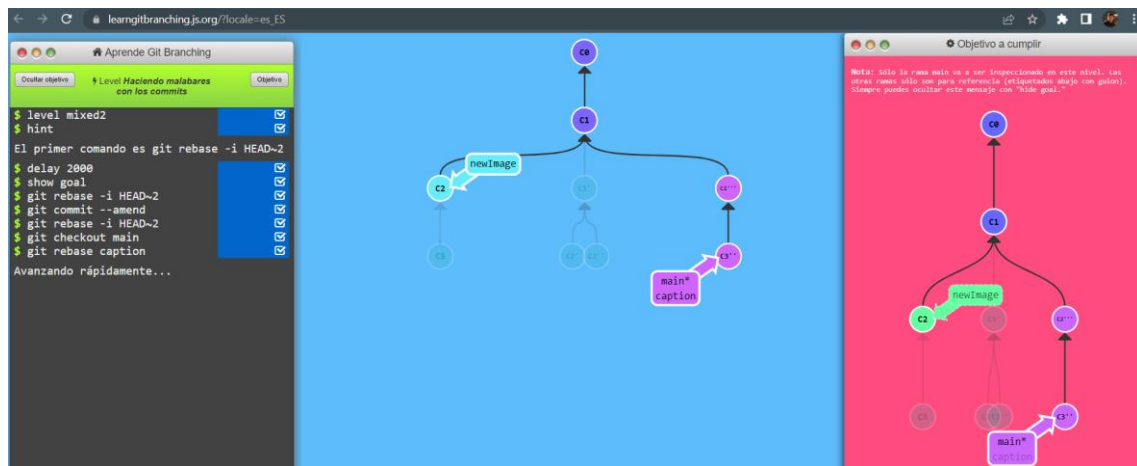
Noel@DESKTOP-FJ0CUOU MINGW64 /d/mdeis/11. Integracion, Entrega y Despliegue Continuo/practica-git (main)
$ |

```

11. Tomando un único commit



12. Haciendo malabares con los commits



13. Haciendo malabares con los commits # 2

Objetivo a cumplir

Nota: Sólo la rama main va a ser inspeccionada en este nivel. Las otras ramas sólo son para referencia (¡disparadas! ¡deja con guita!). Siempre puedes ocultar este mensaje con "hide goal."

14. Tags en Git

Objetivo a cumplir

Puedes ocultar esta ventana con "hide goal."

15. Git Describe

Objetivo a cumplir

Puedes ocultar esta ventana con "hide goal."

16. Rebaseando más de 9000 veces

Aprende Git Branching

Objetivo: Rebaseando más de 9000 veces

```
$ level advanced1
$ hint
Recuerda, la manera más eficiente podría
ser actualizar main sólo al final...
$ delay 2000
$ show goal
$ git checkout bugfix
$ git rebase main
$ git checkout side
$ git rebase bugfix
$ git checkout side
$ git checkout another
$ git reset C7
$ git rebase side
$ git checkout main
$ git rebase main
La rama ya se ha actualizado.
$ undo
$ git rebase another
Avanzando rápidamente...
```

Objetivo a cumplir

Nota: Sólo la rama main va a ser inspeccionada en este nivel. Las otras ramas sólo son para referencia (etiquetadas, pero con guión). Siempre puedes ocultar este mensaje con "hide goal".

17. Múltiples padres

Aprende Git Branching

Objetivo: Múltiples padres

```
$ level advanced2
$ hint
Usa 'git branch bugwork' sobre algún
commit para crear la referencia que falta
$ delay 2000
$ show goal
$ git checkout HEAD~2~1
$ git branch bugwork
$ git checkout main
```

Objetivo a cumplir

Puedes ocultar esta ventana con "hide goal".

18. Ensalada de ramas

Aprende Git Branching

Objetivo: Ensalada de ramas

```
$ level advanced3
$ hint
¡Asegúrate de hacer las cosas en el orden
correcto! Crea primero la rama 'one',
después 'two', y después 'three'.
$ delay 2000
$ show goal
$ git checkout one
$ git cherry-pick C4 C3 C2
$ git checkout two
$ git cherry-pick C5 C4 C3 C2
$ git checkout three
$ git rebase C2
Avanzando rápidamente...
$ objective
```

Objetivo a cumplir

Puedes ocultar esta ventana con "hide goal".

REMOTE

The screenshot shows the 'git clone' level of the 'Aprende Git Branching' game. The terminal on the left displays the following commands and output:

```
$ level remote1
$ hint
¡Simplemente escribe 'git clone'!
$ delay 2000
$ show goal
$ git clone
local branch "main" set to track remote
branch "o/main"
```

The central diagram illustrates the initial state: a local 'main' branch (pink circle) pointing to a commit 'c1' (pink circle), which in turn points to a remote 'main' branch (pink circle) pointing to a commit 'c0' (pink circle). The right panel shows the goal state, which is identical to the initial state.

The screenshot shows the 'git fetch' level. The terminal on the left displays the following commands and output:

```
$ level remote3
$ hint
Simplemente ¡escribe git fetch!
$ delay 2000
$ show goal
$ git fetch
```

The central diagram illustrates the state after a 'git fetch' operation. The local 'main' branch (pink circle) now points to a commit 'c1' (pink circle), which points to a remote 'main' branch (pink circle) pointing to a commit 'c0' (pink circle). The right panel shows the goal state, which is identical to the initial state.

The screenshot shows the 'git pull' level. The terminal on the left displays the following commands and output:

```
$ level remote4
$ hint
Simplemente ¡ejecuta git pull!
$ delay 2000
$ show goal
$ git pull
```

The central diagram illustrates the state after a 'git pull' operation. The local 'main' branch (pink circle) now points to a commit 'c1' (pink circle), which points to a remote 'main' branch (pink circle) pointing to a commit 'c0' (pink circle). The right panel shows the goal state, which is identical to the initial state.

The screenshot shows the 'Push y Pull -- Git Remotes!' level. The terminal on the left displays the following commands and output:

```
$ level remote4
$ hint
Simplemente ¡ejecuta git pull!
$ delay 2000
$ show goal
$ git pull
```

The central diagram illustrates the state after a 'git pull' operation. The local 'main' branch (pink circle) now points to a commit 'c1' (pink circle), which points to a remote 'main' branch (pink circle) pointing to a commit 'c0' (pink circle). The right panel shows the goal state, which is identical to the initial state.

Selecciona un nivel

Principal Remota

Push y Pull -- Git Remotes!

Hora de compartir vuestros 1's y 0's, chicos; programar se volvió social!

1: Introducción a clone

✓ ✓ ✓ ✓ ✓ ✓ ✓

Hasta el origen y más allá -- Git Remotes avanzado!