

Universidad de Costa Rica
Facultad de Ingeniería
Escuela de Ingeniería Eléctrica
IE0117 – Programación Bajo Plataformas Abiertas
II ciclo 2023

Laboratorio Git - GitHub

Oscar Porras Silesky C16042
Grupo 001

Profesor: Ing. Julián Gairaud

Asistente: Felipe Agüero Peralta

2 de diciembre de 2023

1. Pregunta 1: Teoría e investigación

1.1. Describa la diferencia entre un commit y un push en Git.

En Git, un "commit" es una agrupación de cambios acompañados de información como la fecha y hora, el autor y un identificador SHA. Por su parte, el "push" es el proceso de transferir estos commits del repositorio local a uno remoto. Fundamentalmente, la diferencia clave es que el commit almacena las modificaciones en el repositorio local, mientras que el push se encarga de enviar esas alteraciones al repositorio remoto.

1.2. ¿Qué es un conflicto de fusión (merge conflict) en Git y cómo se resuelve?

Un conflicto de fusión en Git surge cuando existen modificaciones contradictorias en las mismas partes de un archivo en distintas ramas. Este problema ocurre habitualmente durante una operación de fusión (merge), y se manifiesta cuando Git no logra determinar de manera automática qué cambios deben mantenerse. Para solucionarlo, hay una serie de pasos a seguir. Inicialmente, se debe verificar el estado actual mediante el comando `$ git status`, que ofrece detalles sobre la rama activa y el estado de los cambios. Posteriormente, se requiere abrir los archivos afectados y resolver los conflictos de forma manual, seleccionando los cambios a retener. Una vez resueltos los conflictos, se deben agregar los archivos ajustados con `git add` y realizar un commit mediante `git commit`. Este procedimiento termina el proceso de fusión.

1.3. ¿Cuál es la diferencia entre git merge y git rebase? Proporcione un escenario en el que sería preferible usar uno sobre el otro.

La distinción principal entre `git merge` y `git rebase` en Git radica en cómo manejan los cambios entre ramas. Mientras `git merge` une las líneas temporales de dos ramas, creando un nuevo commit de fusión que señala el punto de convergencia, `git rebase` reordena los commits de una rama sobre otra. Al ejecutar `$ git rebase origin/main` desde una rama local, por ejemplo, los commits exclusivos de la rama local se sitúan sobre los últimos commits de `origin/main`. Aunque `git rebase` puede complicar la historia en ramas que se comparten públicamente al reescribir la historia de commits. En términos de uso preferido, `git merge` es más adecuado cuando es importante mantener el historial completo, como en ramas públicas o compartidas, por ejemplo, al integrar una rama de características a la rama principal (`main` o `master`).

1.4. ¿Cuál es el propósito principal de GitHub en el contexto de Git?

La función principal de GitHub en relación con Git es la de ser una plataforma de hospedaje para control de versiones y colaboración. Git, como un sistema de control de versiones distribuido, facilita el seguimiento y la coordinación de cambios en los archivos. GitHub, en cambio, ofrece un espacio en línea para almacenar, compartir y manejar estos repositorios de Git. Además, GitHub enriquece las capacidades de Git con características adicionales, como una interfaz de usuario gráfica en la web, control de acceso y seguimiento de errores. Estas funcionalidades hacen que GitHub sea de gran ayuda para el trabajo colaborativo en equipo, proporcionando medios para contribuir a proyectos, revisar código y manejar modificaciones. En resumen, Git se enfoca en el manejo de versiones de archivos, y GitHub sirve como una plataforma integral para alojar y colaborar, potenciando las funcionalidades de Git.

1.5. Describa a detalle el proceso de enviar una solicitud de extracción (pull request) en GitHub.

Para iniciar una solicitud de extracción (pull request) en GitHub, se comienza desde la interfaz web, en la sección "Pull Requests". En este espacio, se selecciona que los cambios hechos en una nueva rama, que ya ha sido creada y subida al repositorio remoto (origin), se integren con la rama principal (main). Se muestra un resumen de los cambios efectuados, y es esencial agregar un título y una descripción detallada a la solicitud. GitHub compara automáticamente las ramas involucradas, señalando cualquier conflicto potencial o indicando si la fusión es viable sin problemas. Después de que la solicitud de extracción se complete y se realice la fusión, es aconsejable eliminar la rama secundaria del repositorio remoto para prevenir la acumulación de ramas superfluas, ayudando así a mantener el orden.

2. Pregunta 2: Parte Práctica

2.1. Cree un nuevo repositorio Git en su máquina local.

```

oscar@oscar-vbox:~$ git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:     git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:     git branch -m <name>
Initialized empty Git repository in /home/oscar/.git/
oscar@oscar-vbox:~$ ls -a
.          Downloads      .python_history-02155.tmp      .vboxclient-draganddrop-tty2-control.pid
..         error.txt       salida_error.txt               .vboxclient-draganddrop-tty2-service.pid
.bash_history .git                  salida_errorv2.txt            .vboxclient-hostversion-tty2-control.pid
.bash_logout .lessshst            salida.txt                    .vboxclient-seamless-tty2-control.pid
.bashrc      .local               .selected_editor              .vboxclient-seamless-tty2-service.pid
bitacora.log miprograma.bash      shared                        .vboxclient-vmvga-session-tty2-control.pid
.cache       Music                snap                          .vboxclient-vmvga-session-tty2-service.pid
.config      Pictures             .ssh                          Videos
core.1762    .pkg                 .sudo_as_admin_successful    .vscode
Desktop      .profile             Templates                     wget-hsts
Documents    Public               .vboxclient-clipboard-tty2-control.pid
.dotnet      .python_history      .vboxclient-clipboard-tty2-service.pid

```

Figura 1: Creación y confirmación del repositorio Git en la máquina local.

Se utilizó el comando `git init` para la creación del repositorio.

2.2. Realice al menos 5 commits con mensajes descriptivos, puede ser arreglando error o similar.

```

oscar@oscar-vbox:~$ mkdir dirGit
oscar@oscar-vbox:~$ ls -a
.          .dotnet      .python_history      .vboxclient-clipboard-tty2-service.pid
..         Downloads .python_history-02155.tmp .vboxclient-draganddrop-tty2-control.pid
.bash_history error.txt       salida_error.txt     .vboxclient-draganddrop-tty2-service.pid
.bash_logout .git           salida_errorv2.txt   .vboxclient-hostversion-tty2-control.pid
.bashrc      .lessshst      salida.txt            .vboxclient-seamless-tty2-control.pid
bitacora.log .local         .selected_editor     .vboxclient-seamless-tty2-service.pid
.cache       miprograma.bash shared                .vboxclient-vmvga-session-tty2-control.pid
.config      Music          snap                  .vboxclient-vmvga-session-tty2-service.pid
core.1762    Pictures       .ssh                  Videos
Desktop      .pkg           .sudo_as_admin_successful .vscode
dirGit       .profile       Templates              wget-hsts
Documents    Public         .vboxclient-clipboard-tty2-control.pid
oscar@oscar-vbox:~$ cd dirGit/
oscar@oscar-vbox:~/dirGit$ touch arch1.txt
oscar@oscar-vbox:~/dirGit$ touch arch2.txt
oscar@oscar-vbox:~/dirGit$ touch arch3.txt
oscar@oscar-vbox:~/dirGit$ touch arch4.txt
oscar@oscar-vbox:~/dirGit$ touch arch5.txt

```

Figura 2: Creación de 5 archivos para los commits.

```

oscar@oscar-vbox:~/dirGit$ git add arch1.txt
oscar@oscar-vbox:~/dirGit$ git commit -m "Commit del archivo numero 1"
[master (root-commit) 23eb662] Commit del archivo numero 1
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 dirGit/arch1.txt
oscar@oscar-vbox:~/dirGit$ git add arch2.txt
oscar@oscar-vbox:~/dirGit$ git commit -m "Commit del archivo numero 2"
[master f3127ce] Commit del archivo numero 2
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 dirGit/arch2.txt
oscar@oscar-vbox:~/dirGit$ git add arch3.txt
oscar@oscar-vbox:~/dirGit$ git commit -m "Commit del archivo numero 3"
[master 5671c18] Commit del archivo numero 3
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 dirGit/arch3.txt
oscar@oscar-vbox:~/dirGit$ git add arch4.txt
oscar@oscar-vbox:~/dirGit$ git commit -m "Commit del archivo numero 4"
[master 3081865] Commit del archivo numero 4
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 dirGit/arch4.txt
oscar@oscar-vbox:~/dirGit$ git add arch5.txt
oscar@oscar-vbox:~/dirGit$ git commit -m "Commit del archivo numero 5"
[master b2bb471] Commit del archivo numero 5
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 dirGit/arch5.txt
oscar@oscar-vbox:~/dirGit$

```

Figura 3: Realización de commits.

```

oscar@oscar-vbox:~/dirGit$ git log
commit b2bb471f4b6687440a5dd849a31b8c7131bbad02 (HEAD -> master)
Author: Kroenenn <gabrielp200325@gmail.com>
Date: Sat Dec 2 11:54:51 2023 -0600

    Commit del archivo numero 5

commit 3081865f189ae3627978e6020633bcf66537a3d9
Author: Kroenenn <gabrielp200325@gmail.com>
Date: Sat Dec 2 11:54:43 2023 -0600

    Commit del archivo numero 4

commit 5671c1845c528c8555a5777d979c05c494b0e7ce
Author: Kroenenn <gabrielp200325@gmail.com>
Date: Sat Dec 2 11:54:35 2023 -0600

    Commit del archivo numero 3

commit f3127ce765a70d6ae9bf61c13d17bb965b18d140
Author: Kroenenn <gabrielp200325@gmail.com>
Date: Sat Dec 2 11:54:27 2023 -0600

    Commit del archivo numero 2

commit 23eb6624fd46e1fdec6d193be08f9f7ea5654bbb
Author: Kroenenn <gabrielp200325@gmail.com>
Date: Sat Dec 2 11:54:13 2023 -0600

    Commit del archivo numero 1
oscar@oscar-vbox:~/dirGit$

```

Figura 4: Confirmación de los 5 commits.

Para los 5 commits, se decidió hacer la carpeta "dirGit" donde se crearon 5 archivos "txt" vacíos con el comando "touch". Luego se usaron los comandos "git add", y finalmente, con "git commit -m" se subió el commit al repositorio local master.

2.3. Cree una rama llamada “finaldesemestre” y realice al menos 2 commits en esa rama.

```

oscar@oscar-vbox:~/dirGit$ git branch finaldesemestre
oscar@oscar-vbox:~/dirGit$ git checkout finaldesemestre
Switched to branch 'finaldesemestre'
oscar@oscar-vbox:~/dirGit$ git branch
* finaldesemestre
  master
oscar@oscar-vbox:~/dirGit$ touch commit1finaldesemestre.txt
oscar@oscar-vbox:~/dirGit$ touch commit2finaldesemestre.txt
oscar@oscar-vbox:~/dirGit$ git add commit1finaldesemestre.txt
oscar@oscar-vbox:~/dirGit$ git commit -m "Commit 1 finaldesemestre"
[finaldesemestre f7c7339] Commit 1 finaldesemestre
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 dirGit/commit1finaldesemestre.txt
oscar@oscar-vbox:~/dirGit$ git add commit2finaldesemestre.txt
oscar@oscar-vbox:~/dirGit$ git commit -m "Commit 2 finaldesemestre"
[finaldesemestre d31223c] Commit 2 finaldesemestre
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 dirGit/commit2finaldesemestre.txt

```

Figura 5: Creación de rama "finaldesemestre" junto con los commits.

```

oscar@oscar-vbox:~/dirGit$ git log
commit d31223c3cde17cd4a879cea3dda08818cab146c6 (HEAD -> finaldesemestre)
Author: Kroenenn <gabrielp200325@gmail.com>
Date: Sat Dec 2 11:58:37 2023 -0600

    Commit 2 finaldesemestre

commit f7c7339519771aa370ad96f53a70a9aeb1460524
Author: Kroenenn <gabrielp200325@gmail.com>
Date: Sat Dec 2 11:58:19 2023 -0600

    Commit 1 finaldesemestre

commit b2bb471f4b6687440a5dd849a31b8c7131bbad02 (master)
Author: Kroenenn <gabrielp200325@gmail.com>
Date: Sat Dec 2 11:54:51 2023 -0600

    Commit del archivo numero 5

commit 3081865f189ae3627978e6020633bcf66537a3d9
Author: Kroenenn <gabrielp200325@gmail.com>
Date: Sat Dec 2 11:54:43 2023 -0600

    Commit del archivo numero 4

commit 5671c1845c528c8555a5777d979c05c494b0e7ce
Author: Kroenenn <gabrielp200325@gmail.com>
Date: Sat Dec 2 11:54:35 2023 -0600

    Commit del archivo numero 3

commit f3127ce765a70d6ae9bf61c13d17bb965b18d140
Author: Kroenenn <gabrielp200325@gmail.com>
Date: Sat Dec 2 11:54:27 2023 -0600
...skipping...
commit d31223c3cde17cd4a879cea3dda08818cab146c6 (HEAD -> finaldesemestre)
Author: Kroenenn <gabrielp200325@gmail.com>
Date: Sat Dec 2 11:58:37 2023 -0600

    Commit 2 finaldesemestre

```

Figura 6: Verificación de la creación de la rama con los commits.

```

commit f7c7339519771aa370ad96f53a70a9aeb1460524
Author: Kroenenn <gabriels200325@gmail.com>
Date: Sat Dec 2 11:58:19 2023 -0600

    Commit 1 finaldesemestre

commit b2bb471f4b6687440a5dd849a31b8c7131bbad02 (master)
Author: Kroenenn <gabriels200325@gmail.com>
Date: Sat Dec 2 11:54:51 2023 -0600

    Commit del archivo numero 5

commit 3081865f189ae3627978e6020633bcf66537a3d9
Author: Kroenenn <gabriels200325@gmail.com>
Date: Sat Dec 2 11:54:43 2023 -0600

    Commit del archivo numero 4

commit 5671c1845c528c8555a5777d979c05c494b0e7ce
Author: Kroenenn <gabriels200325@gmail.com>
Date: Sat Dec 2 11:54:35 2023 -0600

    Commit del archivo numero 3

commit f3127ce765a70d6ae9bf61c13d17bb965b18d140
Author: Kroenenn <gabriels200325@gmail.com>
Date: Sat Dec 2 11:54:27 2023 -0600
:

```

Figura 7: Segunda parte de verificación.

Se utilizó el comando "git branch finaldesemestre" y luego "git checkout finaldesemestre" donde luego con el comando "touch", "git add" y "git commit -m" se realizaron los commits en esa rama.

2.4. Fusione la rama "finaldesemestre" con la rama principal usando un merge.

```

oscar@oscar-vbox:~/dirGit$ git checkout master
Switched to branch 'master'
oscar@oscar-vbox:~/dirGit$ git merge finaldesemestre
Updating b2bb471..d31223c
Fast-forward
 dirGit/commit1finaldesemestre.txt | 0
 dirGit/commit2finaldesemestre.txt | 0
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 dirGit/commit1finaldesemestre.txt
 create mode 100644 dirGit/commit2finaldesemestre.txt
oscar@oscar-vbox:~/dirGit$ git log
commit d31223c3cde17cd4a879cea3dda08818cab146c6 (HEAD -> master, finaldesemestre)
Author: Kroenenn <gabriels200325@gmail.com>
Date: Sat Dec 2 11:58:37 2023 -0600

    Commit 2 finaldesemestre

commit f7c7339519771aa370ad96f53a70a9aeb1460524
Author: Kroenenn <gabriels200325@gmail.com>
Date: Sat Dec 2 11:58:19 2023 -0600

    Commit 1 finaldesemestre

```

Figura 8: Fusión de la rama finaldesemestre con la principal.

Se utilizó el comando "git checkout master", "git merge finaldesemestre" y "git log" para confirmar.

2.5. Cree una nueva rama llamada “verano” y realice al menos 3 commits en esa rama.

```

oscar@oscar-vbox:~/dirGit$ git branch verano
oscar@oscar-vbox:~/dirGit$ git checkout verano
Switched to branch 'verano'
oscar@oscar-vbox:~/dirGit$ touch verano1.txt
oscar@oscar-vbox:~/dirGit$ touch verano2.txt
oscar@oscar-vbox:~/dirGit$ touch verano3.txt
oscar@oscar-vbox:~/dirGit$ git add verano1.txt
oscar@oscar-vbox:~/dirGit$ git commit -m "Commit verano1"
[verano 153c25c] Commit verano1
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 dirGit/verano1.txt
oscar@oscar-vbox:~/dirGit$ git add verano2.txt
oscar@oscar-vbox:~/dirGit$ git commit -m "Commit verano2"
[verano 376924e] Commit verano2
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 dirGit/verano2.txt
oscar@oscar-vbox:~/dirGit$ git add verano3.txt
oscar@oscar-vbox:~/dirGit$ git commit -m "Commit verano3"
[verano cfa2d7a] Commit verano3
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 dirGit/verano3.txt
oscar@oscar-vbox:~/dirGit$

```

Figura 9: Creación de la rama llamada verano.

```

oscar@oscar-vbox:~/dirGit$ git log
commit cfa2d7a1eea0a3359af7059d142398ead5c751cc (HEAD -> verano)
Author: Kroenenn <gabrielp200325@gmail.com>
Date: Sat Dec 2 12:04:21 2023 -0600

    Commit verano3

commit 376924e839050db0ee70fc2f914aa5fa1072695d
Author: Kroenenn <gabrielp200325@gmail.com>
Date: Sat Dec 2 12:04:14 2023 -0600

    Commit verano2

commit 153c25ca04a4814f98f2636fad70f5b1f91170
Author: Kroenenn <gabrielp200325@gmail.com>
Date: Sat Dec 2 12:04:06 2023 -0600

    Commit verano1

```

Figura 10: Confirmación de la creación de la rama .

Se utilizó el comando "git branch verano", "git checkout verano", se utilizó el comando "touch" para crear archivos para los commits. Se usó "git add" y "git commit -m".

2.6. Rebase la rama “verano” en la rama principal.

```

oscar@oscar-vbox:~/dirGit$ git checkout master
Switched to branch 'master'
oscar@oscar-vbox:~/dirGit$ git rebase verano
Successfully rebased and updated refs/heads/master.
oscar@oscar-vbox:~/dirGit$ git log
commit cfa2d7a1eea0a3359af7059d142398ead5c751cc (HEAD -> master, verano)
Author: Kroenenn <gabrielp200325@gmail.com>
Date: Sat Dec 2 12:04:21 2023 -0600

    Commit verano3

commit 376924e839050db0ee70fc2f914aa5fa1072695d
Author: Kroenenn <gabrielp200325@gmail.com>
Date: Sat Dec 2 12:04:14 2023 -0600

    Commit verano2

```

Figura 11: .

Se utilizó "git checkout master", "git rebase verano" y "git log" para la confirmación.

2.7. Cree un repositorio en GitHub.

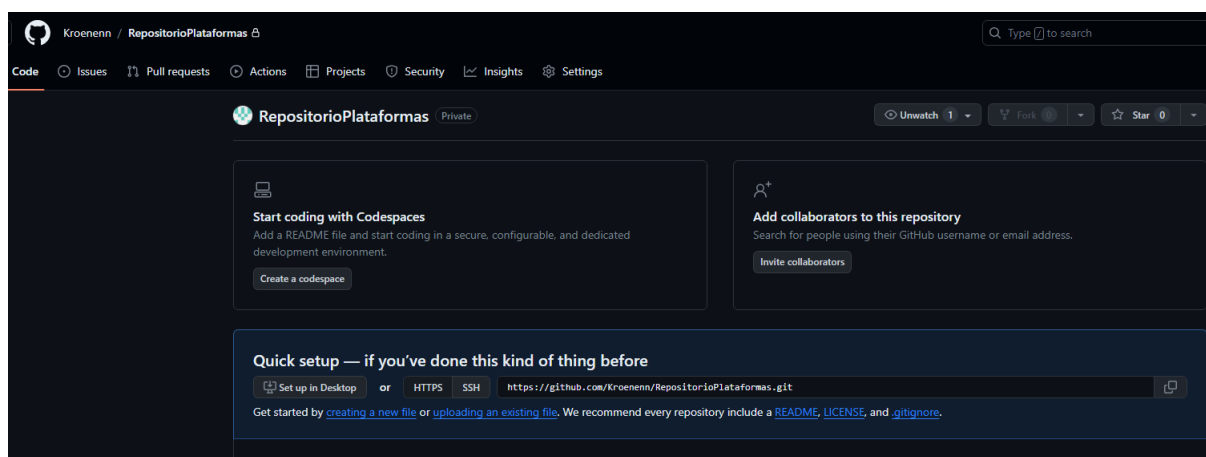


Figura 12: Creación de repositorio en GitHub.

2.8. Suba el repositorio local al repositorio de GitHub.

```

oscar@oscar-vbox:~/dirGit$ git checkout master
Already on 'master'
oscar@oscar-vbox:~/dirGit$ git remote add origin git@github.com:Kroennn/RepositorioPlataformas.git
error: remote origin already exists.
oscar@oscar-vbox:~/dirGit$ git push -u origin master
Username for 'https://github.com': ^Z
[4]+  Stopped                  git push -u origin master
oscar@oscar-vbox:~/dirGit$ git remote set-url origin git@github.com:Kroennn/RepositorioPlataformas.git
oscar@oscar-vbox:~/dirGit$ git checkout master
Already on 'master'
oscar@oscar-vbox:~/dirGit$ git push -u origin master
The authenticity of host 'github.com (140.82.114.4)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvC0QU.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
Enumerating objects: 31, done.
Counting objects: 100% (31/31), done.
Delta compression using up to 4 threads
Compressing objects: 100% (19/19), done.
Writing objects: 100% (31/31), 2.30 KiB | 785.00 KiB/s, done.
Total 31 (delta 8), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (8/8), done.
To github.com:Kroennn/RepositorioPlataformas.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
oscar@oscar-vbox:~/dirGit$

```

Figura 13: Subida del repositorio local a GitHub.

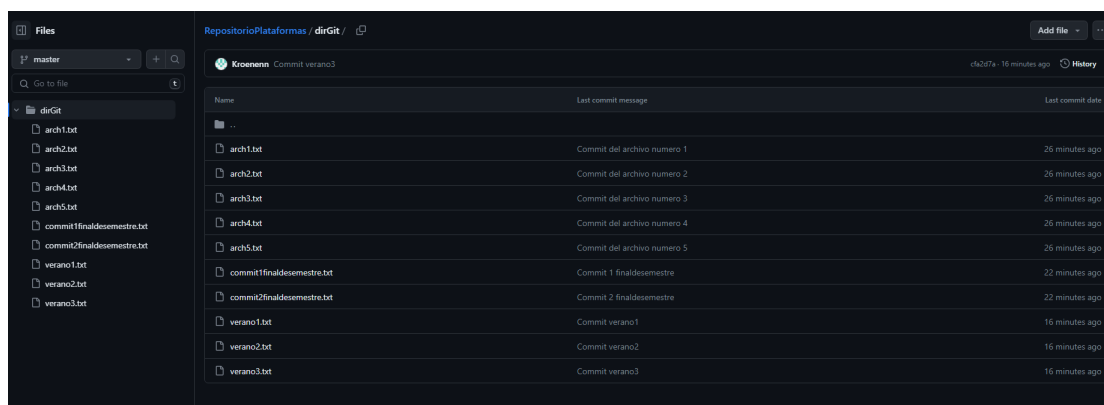


Figura 14: Confirmación de subida de archivos.

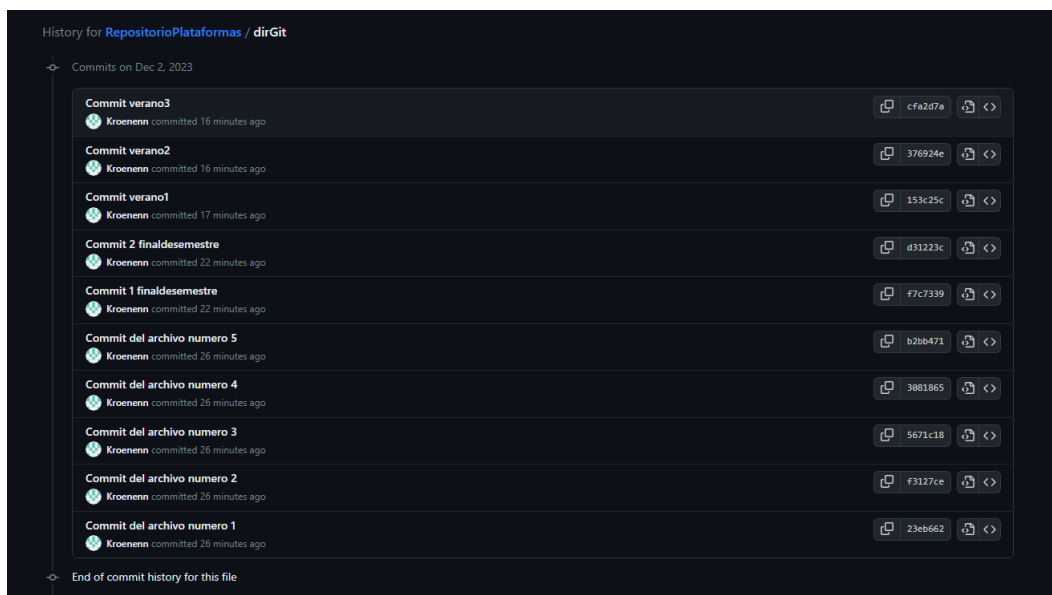


Figura 15: Confirmación de historial.

Se utilizaron los comandos "git remote add origin", "git remote set-url origin" y "git push -u origin master".

2.9. Cree una rama en GitHub, el nombre que usted desee y realice al menos 2 cambios en esa rama.

```
oscar@oscar-vbox:~/dirGit$ git branch nueva
oscar@oscar-vbox:~/dirGit$ git checkout nueva
Switched to branch 'nueva'
oscar@oscar-vbox:~/dirGit$ touch nueva1cambio.txt
oscar@oscar-vbox:~/dirGit$ touch nueva2cambio.txt
oscar@oscar-vbox:~/dirGit$ git add nueva1cambio.txt nueva2cambio.txt
oscar@oscar-vbox:~/dirGit$ git commit -m "dos cambios en la rama"
[nueva 4c6bea4] dos cambios en la rama
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 dirGit/nueva1cambio.txt
create mode 100644 dirGit/nueva2cambio.txt
oscar@oscar-vbox:~/dirGit$ git push
fatal: The current branch nueva has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin nueva

oscar@oscar-vbox:~/dirGit$ git push origin nueva
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 298 bytes | 298.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'nueva' on GitHub by visiting:
remote:   https://github.com/Kroenenn/RepositorioPlataformas/pull/new/nueva
remote:
To github.com:Kroenenn/RepositorioPlataformas.git
 * [new branch]      nueva -> nueva
```

Figura 16: Creación de rama "nueva" con los 2 cambios.

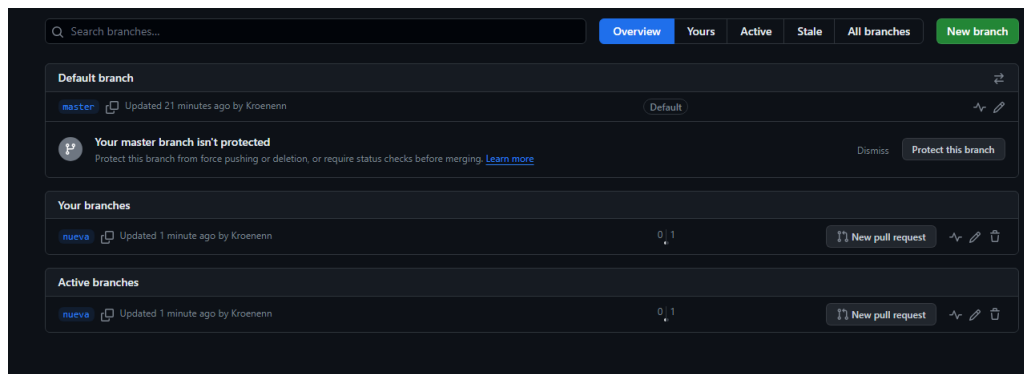


Figura 17: Confirmación de creación de la rama.

Se utilizó los comandos "git branch nueva", "git checkout nueva", "touch" para creación de archivos, "git add", "git commit -m" y "git push origin nueva".

2.10. Finalmente envié una solicitud de extracción desde la rama que creó a la rama principal en GitHub.

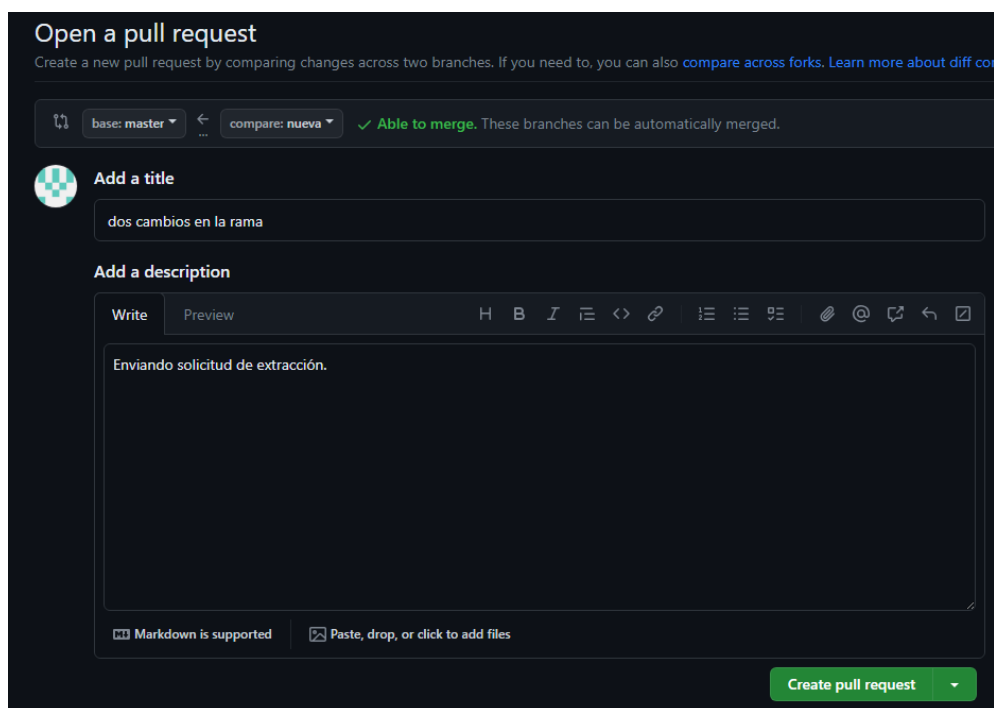


Figura 18: Envío de extracción desde GitHub.

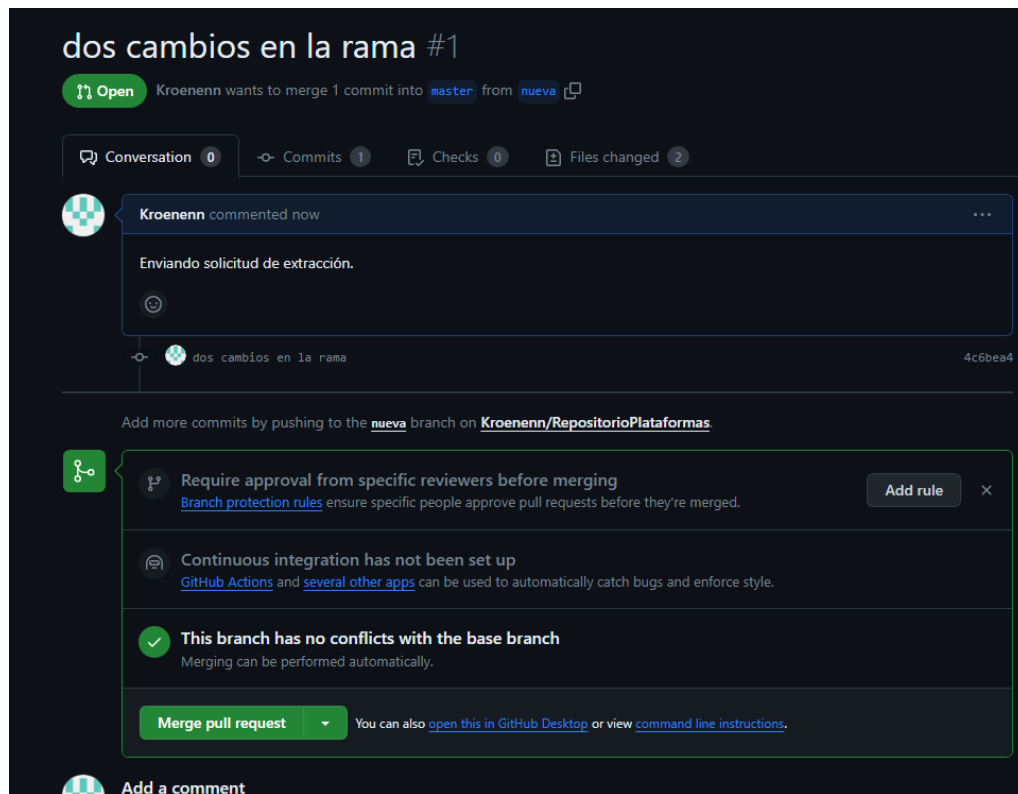


Figura 19: Confirmación del envío de extracción.

Se utilizó GitHub para llevar a cabo el proceso.