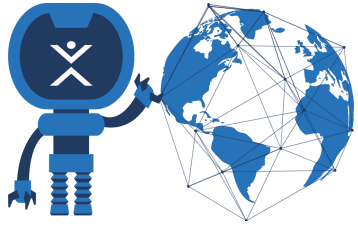


CUMBRE VIRTUAL

Comunidades Microsoft del Ecuador





CUMBRE VIRTUAL

Comunidades Microsoft del Ecuador

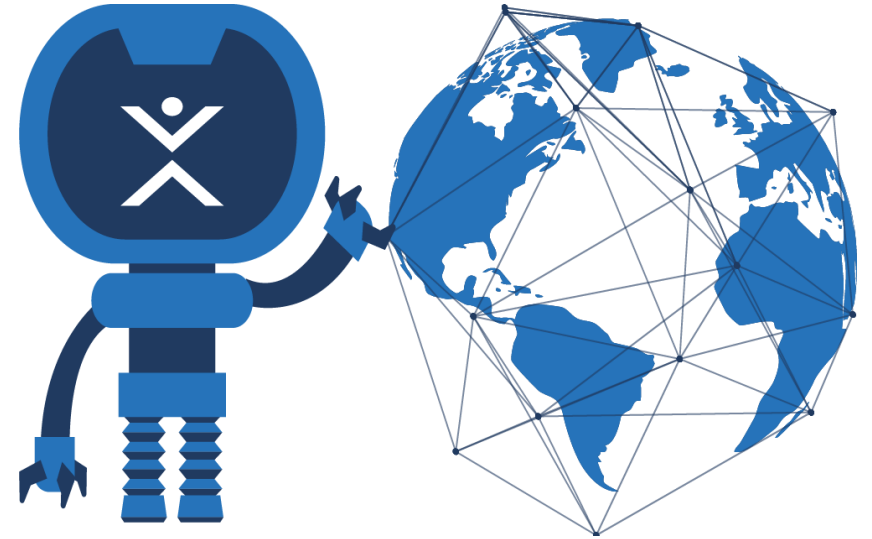
Introducción a GitKraken

Andy Enríquez

Anderson Pozo



¿Por qué estamos aquí?



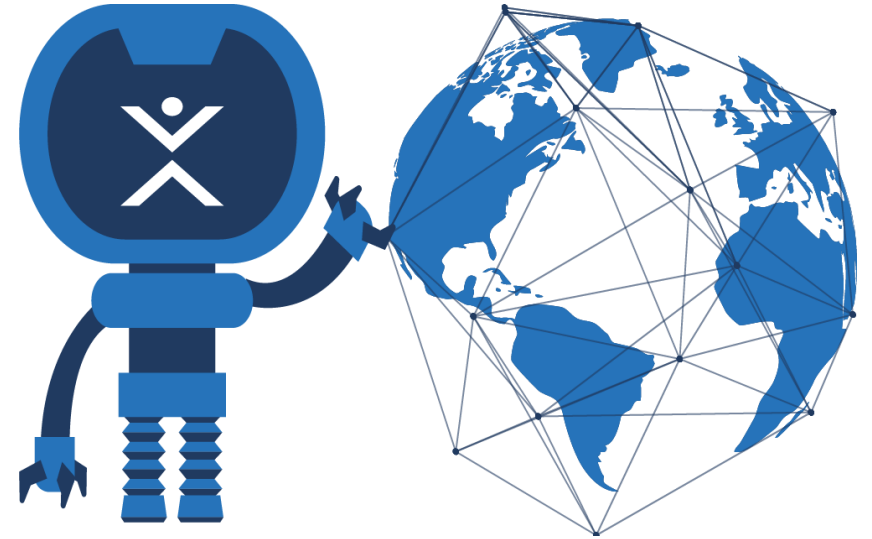
¿Por qué estamos aquí?



- ¿Cuánta gente de aquí ha tenido que colaborar en un proyecto de código antes y no estaba segura de cómo hacerlo?
- ¿Cuántos de nosotros hemos usado Git/GitHub u otras herramientas para colaborar y hemos sentido que era un proceso realmente difícil?
- Tal vez has trabajado en un proyecto con otros usando Git y te ha costado mucho trabajo averiguar quién ha hecho qué.
- ¿O tal vez te sientes cómodo con Git, pero no estás seguro de cómo rastrear quién está trabajando en qué?



¿Qué es Git?



¿Qué es Git?

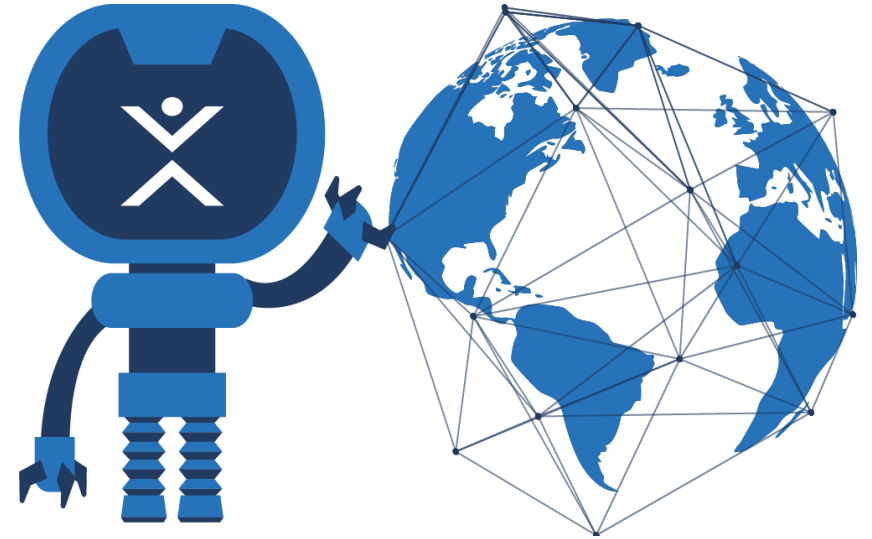


Git es un sistema de control de versiones que te ayuda a colaborar en proyectos con otros. Hay dos características principales que te ayudan a conseguirlo.

1. Registra los cambios de manera que se puedan buscar, en lugar de renombrar un archivo para cada versión.
2. Los colaboradores pueden trabajar en paralelo y fusionar sus cambios automáticamente, en lugar de comparar manualmente las diferencias entre un archivo



Git de la vieja escuela





Algunos desarrolladores utilizan la línea de comandos para interactuar con Git y GitHub siguiendo el flujo de trabajo de mejores prácticas descrito en las siguientes diapositivas.

1. Primero, podrían bifurcar el código de otra persona. Esto crea su propia copia del proyecto de otra persona.
2. Luego crearían una rama. La rama es una versión paralela de su bifurcación, donde pueden probar los cambios sin editar la copia maestra.

Términos clave

fork: tu propia copia del depósito de otra persona.

branch: una versión paralela de la copia maestra de un repo. Hacer una rama permite editar el código sin romper accidentalmente una versión de trabajo.



3. Escenifican sus cambios a medida que avanzan. Cuando están contentos con ellos, realizan un commit
4. Para añadir su versión de trabajo (rama) a la rama maestra, abren un pull request
5. Entonces, ellos (o el dueño del proyecto) fusionarán la rama en la rama maestra.

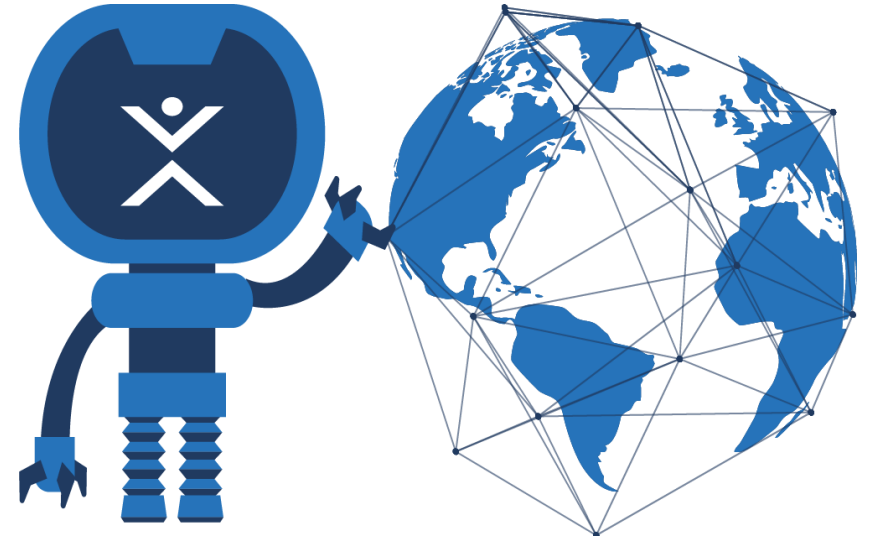
Términos clave

stage: añadir a un grupo cohesivo/un conjunto de revisiones

commit: un grupo de revisiones que quieres añadir oficialmente a tu rama

merge: para añadir oficialmente los cambios de su rama en la rama maestra (u otra rama)

¿Cómo lo hacen?

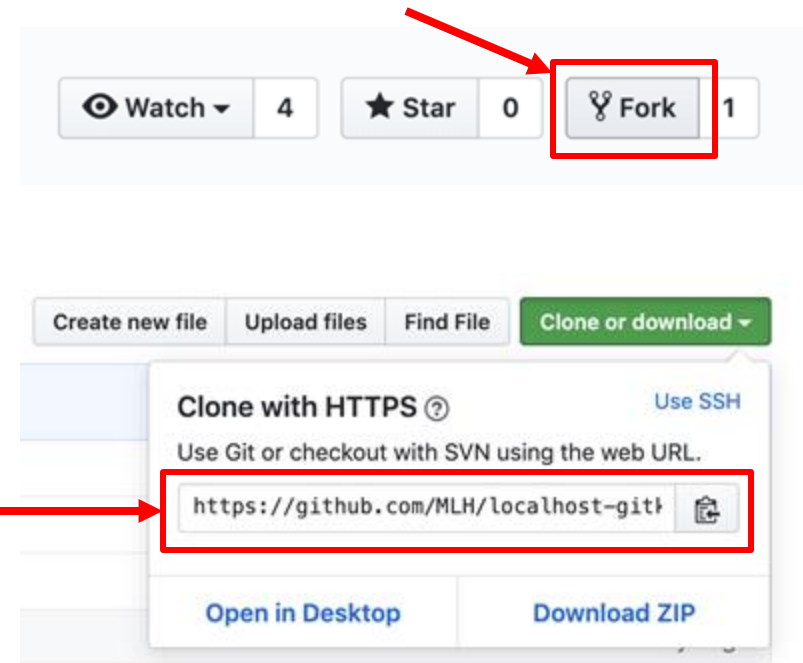


¿Cómo lo hacen?



Si un desarrollador está usando la línea de comandos para completar estos pasos, su flujo de trabajo de la vieja escuela podría parecerse a esto:

1. Ve a GitHub para encontrar el proyecto con el que quieren trabajar y haz clic en fork.
2. Ir a su propio perfil de GitHub para ver su bifurcación del repositorio, y copiar la URL del clon.



¿Cómo lo hacen?



3. Abre su terminal e introduce estos comandos para clonar el repo y crear una rama.

```
@localhost > git clone https://github.com/MLH/localhost-gitkraken.git
Cloning into 'localhost-gitkraken'...
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 7 (delta 1), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (7/7), done.
@localhost > cd localhost-gitkraken/
@localhost > git checkout -b "mlh-add-bio"
Switched to a new branch 'mlh-add-bio'
@localhost > █
```

¿Cómo lo hacen?



- Después de hacer los cambios usando su editor de texto, introduzca estos comandos para escenificar sus cambios y confirmarlos.

```
@localhost > git status
On branch mlh-add-bio
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
@localhost > git add README.md
@localhost > git status
On branch mlh-add-bio
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   README.md

@localhost > git commit -m "modify readme"
[mlh-add-bio a60f4e6] modify readme
 1 file changed, 1 insertion(+), 1 deletion(-)
@localhost > █
```

¿Cómo lo hacen?

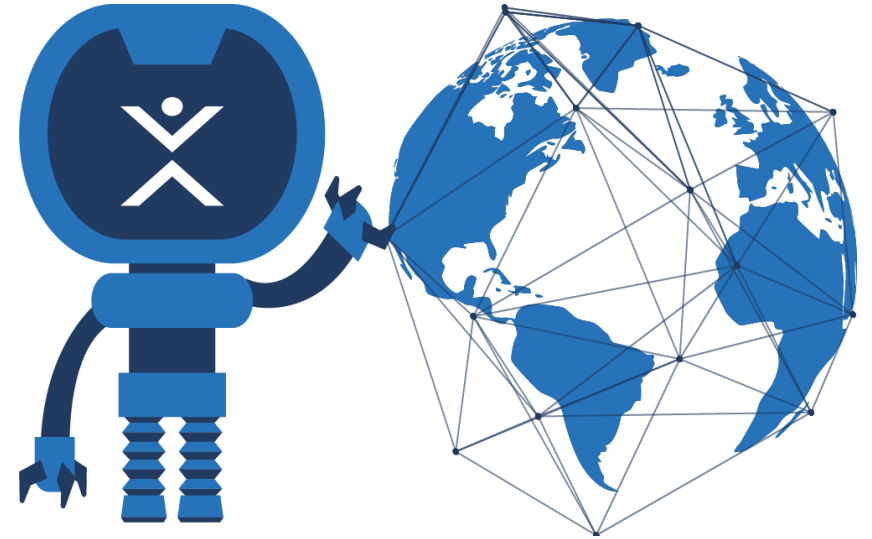


5. Entonces ellos harán un push de los cambios hasta el repositorio de GitHub.

```
@localhost > git push --set-upstream origin mlh-add-bio
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 312 bytes | 312.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
remote:
remote: Create a pull request for 'mlh-add-bio' on GitHub by visiting:
remote:   https://github.com/MLH/localhost-gitkraken/pull/new/mlh-add-bio
remote:
To https://github.com/MLH/localhost-gitkraken.git
 * [new branch]      mlh-add-bio -> mlh-add-bio
Branch 'mlh-add-bio' set up to track remote branch 'mlh-add-bio' from 'origin'.
@localhost > █
```

6. Entonces podrían ir a GitHub para revisar y fusionar el pull request

¿Por qué GitKraken?



¿Por qué GitKraken?

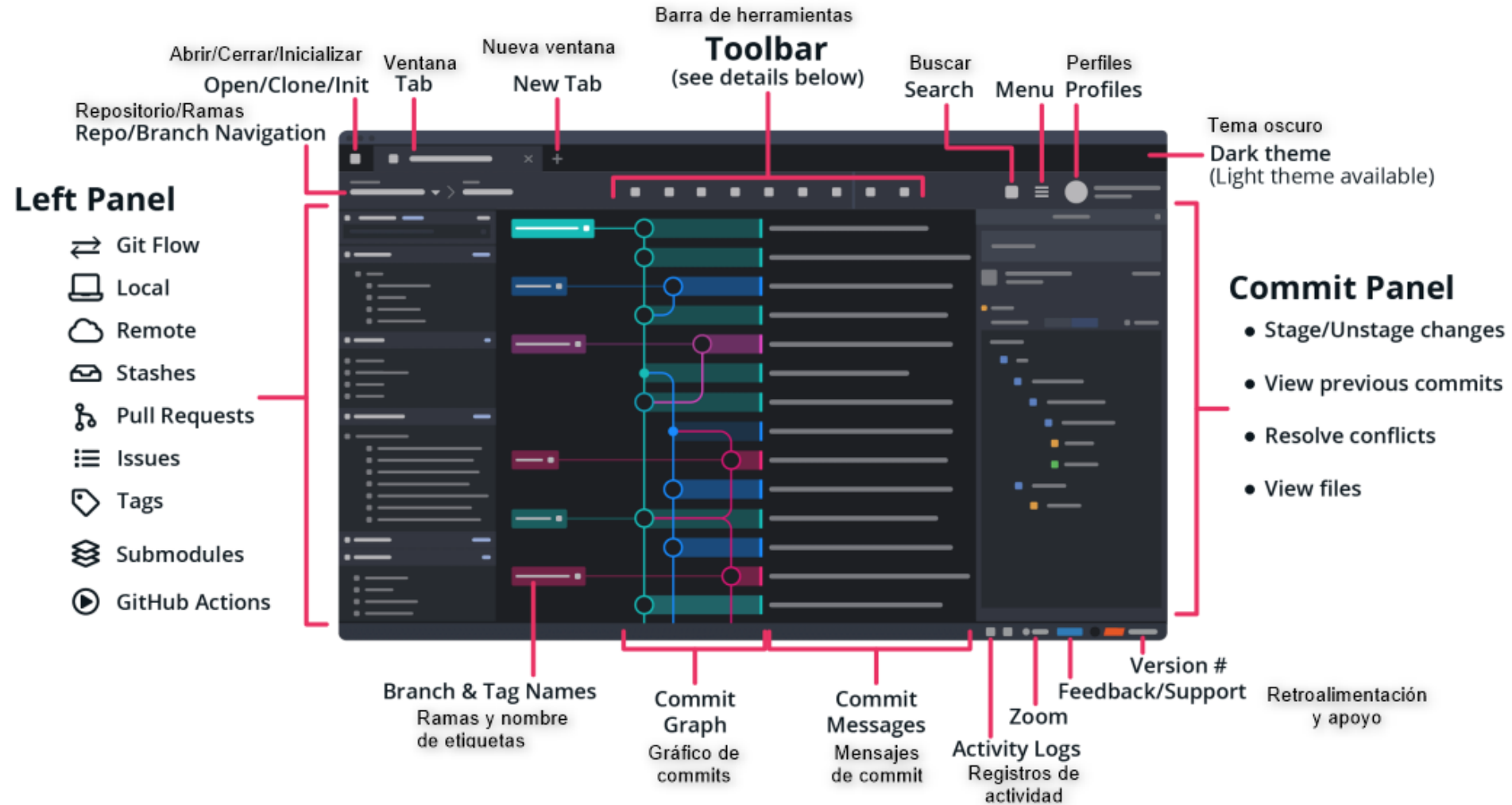


Git es poderoso, pero memorizar los comandos lleva mucho tiempo y la forma en que funciona Git puede ser difícil de visualizar.

GitKraken proporciona una interfaz visual que permite el mismo flujo de trabajo con sólo unos pocos clics.



Interfaz Visual Intuitiva



Interfaz Visual Intuitiva



The screenshot displays the GitHub Desktop application interface. The top bar shows the repository name 'flask' and the current branch 'master'. The left sidebar contains a list of local and remote branches, with 'master' selected. Below this, there are options to select an issue tracker for the repository, including GitHub, GitHub Enterprise, GitKraken Boards, GitLab, GitLab Self-Managed, Jira Cloud, Jira Server, and Trello. The main area shows a commit history graph with a list of commit messages. The right sidebar shows the details of the selected commit, including the commit message, the author (Grey Li), the date (23/11/2020 @ 21:51), and the files modified (1 modified). The file diff for '.pre-commit-config.yaml' is shown at the bottom right.

repository: flask branch: master

Undo Redo Pull Push Branch Stash Pop Boards Timelines

Viewing 37/37 Show All Filter (Ctrl + Alt + f)

LOCAL 1/1

REMOTE 4/4

PULL REQUESTS 4

ISSUES

Select an issue tracker for this repo

GitHub ✓

GitHub Enterprise →

GitKraken Boards ✓

GitLab ✓

GitLab Self-Managed →

Jira Cloud ✓

Jira Server →

Trello →

BRANCH / TAG GRAPH COMMIT MESSAGE

Merge pull request #3375 from pallets/ext-py-version

reword extension python version item

Merge branch '1.1.x'

Merge branch '1.0.x' into 1.1.x

Merge pull request #3374 from pallets/ext-docs

update patterns, snippets, extensions docs

Merge pull request #3349 from nathanmp/master

Update testing.rst

Updating contribution docs. Added the link with a g...

Merge pull request #3343 from gruentee/patch-1

fix typo: base on -> based on

Merge pull request #3332 from Eido95/patch-1

Merge pull request #3336 from arnavb/patch-1

Fix typo in pop documentation

docs: Change max content length value to megabyte

fixed typo in logging docstring (#3328)

Merge branch '1.1.x'

fix stylecheck

Merge branch '1.1.x'

Merge branch '1.0.x' into 1.1.x

commit: 28a3eb

Merge pull request #3842 from pallets/pre-commit-ci-update-config

[pre-commit.ci] pre-commit autoupdate

Grey Li authored 23/11/2020 @ 21:51 parent: 085525,cd3a93

GitHub committed 23/11/2020 @ 21:51

1 modified

Path Tree View all files

.pre-commit-config.yaml

Feedback PRO 7.4.0

¿Por qué GitKraken?



Ramificación y
fusión



Editor de código
integrado



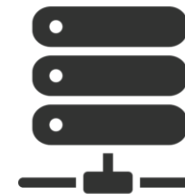
Integraciones de
alojamiento Git



Integraciones de
seguimiento de conflictos



Disponible para entornos
en línea o fuera de línea





¡GRACIAS!



Anderson Pozo

anderson.pozo@upec.edu.ec

@AndersonPozo12

Andy Enríquez

andy.enriquez@upec.edu.ec