

La última guía de colaboración Github

Fuente: <https://medium.com/@jonathanmines/the-ultimate-github-collaboration-guide-df816e98fb67>



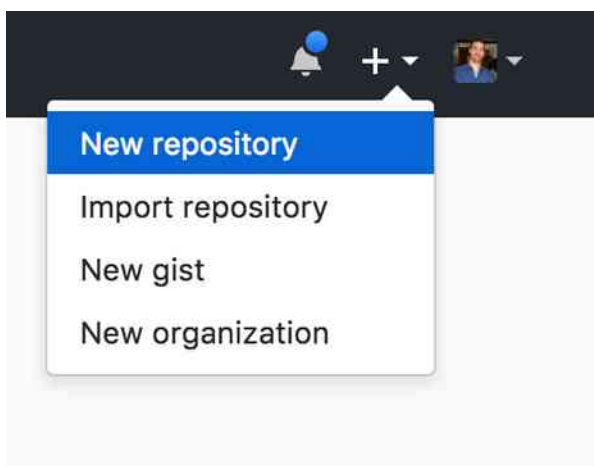
Esta es sólo una de las muchas maneras de colaborar en un proyecto usando GitHub. Pero sería uno que le sugiero si empezara a trabajar con un equipo y no haya establecido un flujo de git todavía o sepa por dónde empezar a establecer uno.

Paso 1: Inicializar un nuevo proyecto

Crear un nuevo proyecto/directorio desde la línea de comandos

```
$ mkdir github-guide
```

Ve a Github y haz clic en el botón de la esquina de la rop derecha y selecciona .



A continuación, rellene el nombre del Repositorio y los campos Descripción. Manténlo público, y no inicies este repositorio con un README. No cambies nada más. Haga clic en el repositorio de Create.

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner: MinesJA / Repository name: github_guide ✓

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

Description (optional): A guide to github

☒ Public
Anyone can see this repository. You choose who can commit.

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

☐ Initialize this repository with a README
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: None Add a license: None ⓘ

Create repository

A continuación verás la página de configuración. Estas son las instrucciones para conectar el Repo que acabas de crear en Github (Remote) con el directorio que creó en tu terminal (Local).

MinesJA / github_guide Watch 0 Star 0 Fork 0

<> Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

Quick setup — if you've done this kind of thing before

Set up in Desktop or HTTPS SSH git@github.com:MinesJA/github_guide.git

We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# github_guide" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin git@github.com:MinesJA/github_guide.git
git push -u origin master
```

...or push an existing repository from the command line

```
git remote add origin git@github.com:MinesJA/github_guide.git
git push -u origin master
```

Pega las líneas de la caja roja en el terminal, empezando con Echo en el terminal mientras estás en el directorio que acabas de crear localmente. El terminal debería verse así cuando haya terminado:

```

[13:51:37] github_guide
// ♥ echo "# github_guide" >> README.md
[13:51:49] github_guide
// ♥ git init
Reinitialized existing Git repository in /Users/jmines/Development/code/practice/github_guide/.git/
[13:51:54] github_guide
// ♥ git add README.md
[13:52:00] github_guide
// ♥ git commit -m "first commit"
[master (root-commit) 168e94f] first commit
1 file changed, 25 insertions(+)
create mode 100644 README.md
[13:52:04] (master) github_guide
// ♥ git remote add origin git@github.com:MinesJA/github_guide.git
[13:52:08] (master) github_guide
// ♥ git push -u origin master
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 465 bytes | 465.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To github.com:MinesJA/github_guide.git
* [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
[13:52:15] (master) github_guide
// ♥

```

Esto añade una carpeta de .git a su repo, le conecta a su remoto Github Repo y también le da un archivo ".gitignore".

Y si vas a tu página de Github Repo, verás la ReadMe con la que inicializó y la referencia al primer commit que hiciste.

Ahora vamos a poner este Repo al día. Vuelve a tu terminal y git add, git commit y git push:

```
$ git add .
```

```
$ git commit -m "Segundo commit"
```

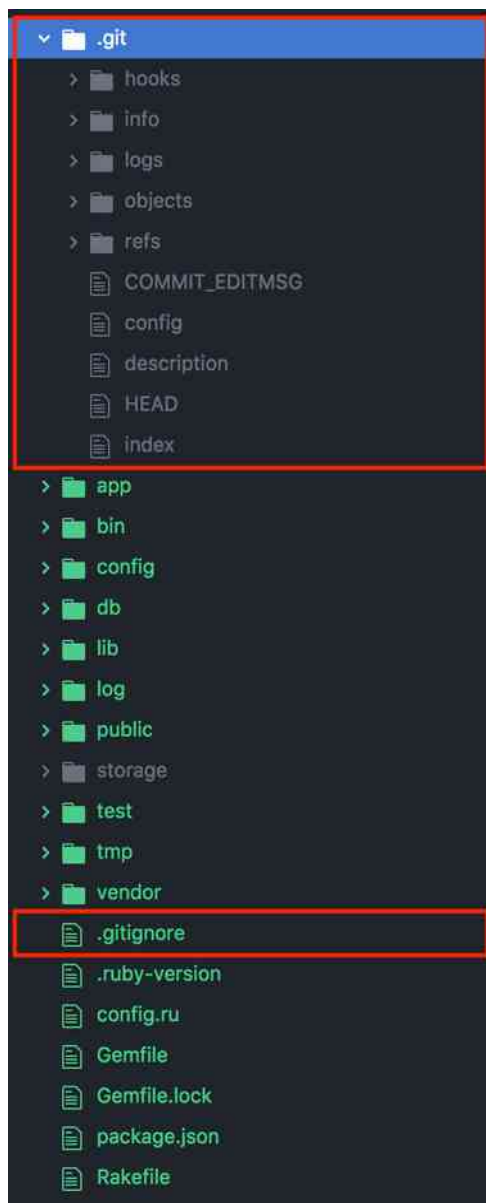
```
$ git push
```

Ahora mira tu repo. Debe tener todos los archivos con los que creó su directorio local junto con un nuevo commit id (9c2e2f6):

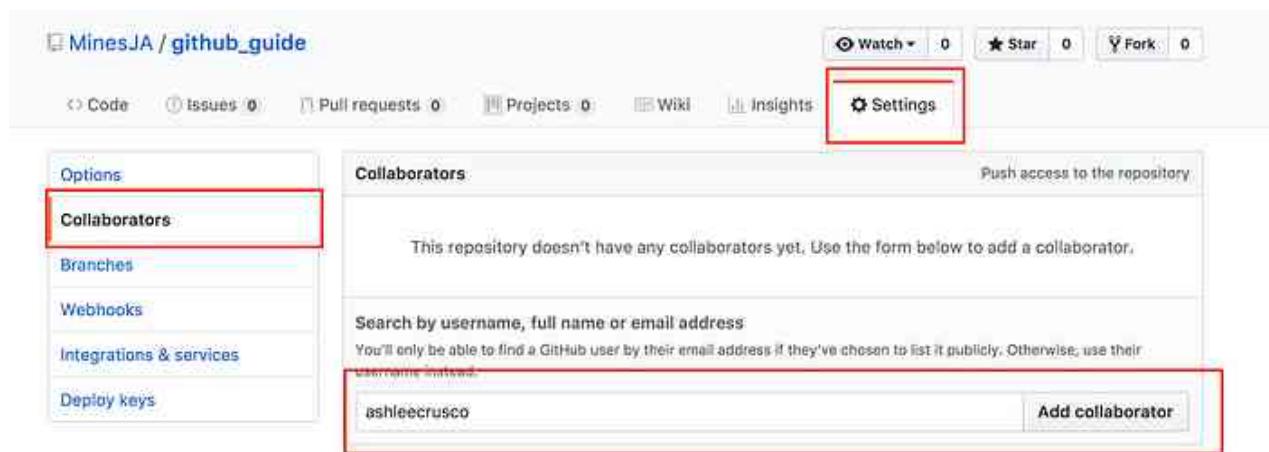
Estás inicializado y listo para empezar a trabajar.

Paso 2: Configurar tu equipo

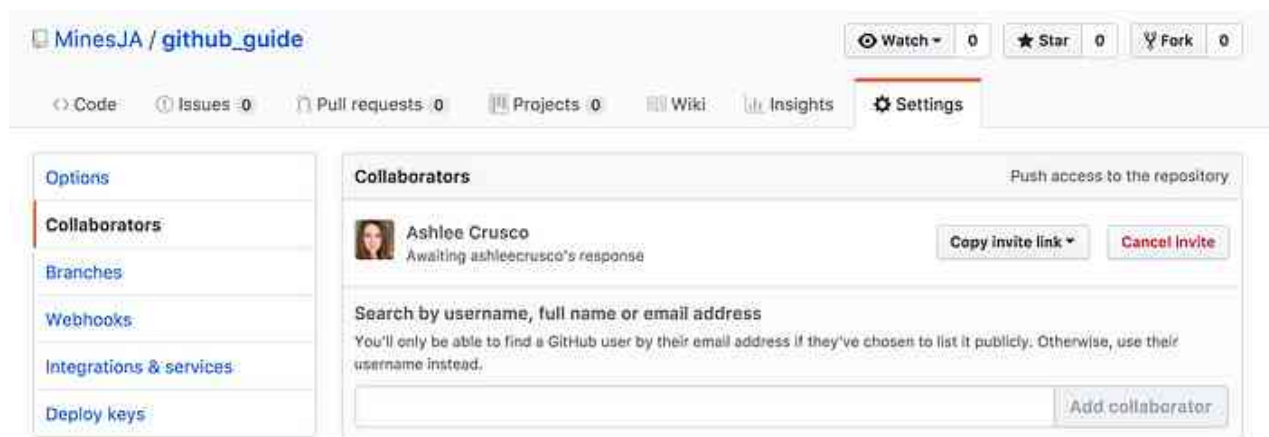
Eres un miembro del equipo, así que vas a tener que añadir tu equipo a tu repo para que puedan colaborar. Una vez que su equipo se añada como colaboradores, podrán empujar, fusionarse y una tonelada de otras cosas destructivas, así que asegúrese de añadir sólo a sus compañeros de equipo.



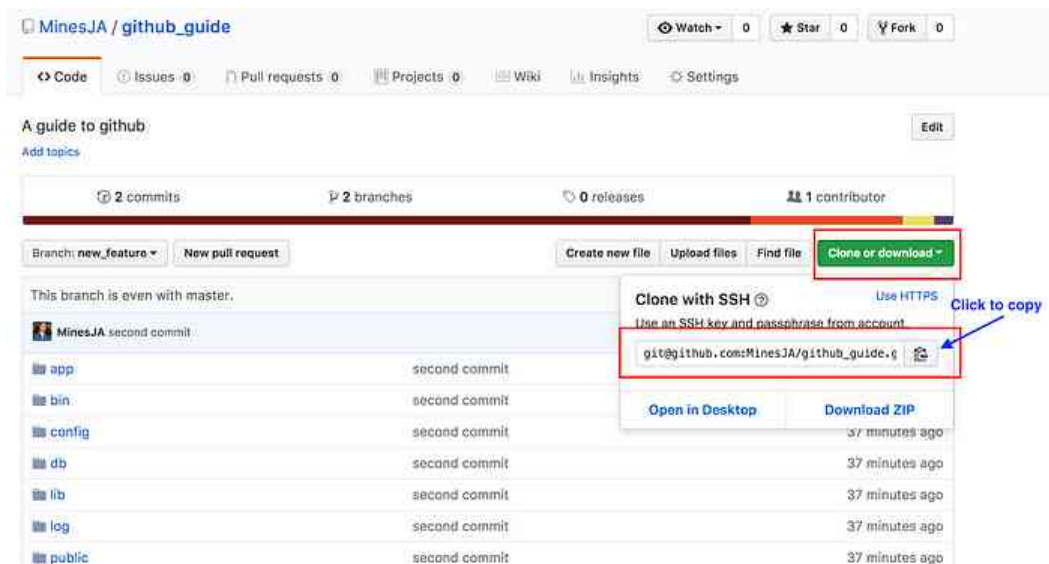
Haz clic en la pestaña "Configuración" de tu repositorio, luego Colaboradores, busca usuarios de Github y agregalos haciendo clic en .Añadir Colaborador:



El usuario recibirá un correo electrónico haciéndole saber que le añadiste y que será incluido como colaborador.



Si eres colaborador, ve a la página Github Repo, Git Clone el proyecto, y cd en el directorio. **No hagas un FORK.** Un Fork lo copiará en una nueva página de Repo a tu página de Github, pero lo que quieres es colaborar en el mismo Github Repo con tus compañeros de equipo.



```
$ git clone git.github.com:MinesJA/github_guide.git
```

```
$ cd github-guide/
```

Y ahora estás listo para colaborar.

Paso 3: Colaborar

Cuando estés usando git para trabajar en el mismo proyecto con varias personas, hay una regla central que debes seguir:

THE MASTER BRANCH SHOULD ALWAYS BE DEPLOYABLE

La manera de mantener al Master implementable es crear nuevas ramas para nuevas características y fusionarlas en Master cuando se completen. Así es cómo funciona esto.

Paso 3a: Ramas

Para empezar, las **ramas** siempre deben representar **características**. Por ejemplo, si desea agregar la capacidad de un usuario para iniciar sesión probablemente debería crear una rama llamada "user_authentication", y en esa rama solo debe actualizar lo que necesita para permitir que un usuario inicie sesión.

También es importante cuando colabora que tu equipo elige características que no tienen código superpuesto. Por ejemplo, no deberías trabajar en una rama de "usuario-login", al mismo tiempo que tu compañero de equipo está trabajando en una rama de "usuario-logout", ya que las posibilidades de que trabajes en los mismos archivos y estés escribiendo código superpuestos son muy altas.

Así que digamos que quieres crear el modelo de Usuario. En su terminal, cree una nueva rama:

```
$ git co -b create_user
```

El "co" es la abreviatura de "checkout" que se utiliza para cambiar entre ramas. Añadir el "-b" y un nombre al final crea una nueva rama y luego se mueve a esa nueva rama.

Lo puedes verificar con el comando:

```
$ git branch
```

Que debería producir:

```
[14:48:31] (new_feature) github_guide
// ♥ git branch
    master
* new_feature
```

Ahora estás en tu nueva rama y puedes empezar a codificar.

Nota: Como regla general, debes usar “git add” y “git commit” cuando termines algo que permita que tu código funcione (termina siendo un par de veces por hora). Por ejemplo, cuando terminas un método y la base de código funciona, “git commit” así:

```
$ git commit -m "Añadida función para permitir a los usuarios decir 'Hola Mundo'"
```

Paso 3b: Atendiendo solicitudes de Pull Request

Su equipo pasó todo el día y la noche trabajando en sus características separadas en sus diversas ramas. Regresan al día siguiente con sus características completas y quieren fusionarlos de nuevo en Master para ser desplegados.

Determinar su flujo de git es una gran parte de trabajar en un equipo, pero aquí está un golpe de punto que podría adoptar por ahora:

Primero, determine quién va a encargarse de la fusión. Cuanta menos gente actúe independientemente en la fusión mejor, así que para un equipo de 4 probablemente te permitiría tener un oficial de Reviewer o "Merge Master".

A continuación, que hagan todos “git push” de sus ramas:

```
$ git push
```

Ahora ve a la página de Github Repo. Usted debe ver la rama que presionó en una barra amarilla en la parte superior de la página con un botón “Compare & pull request”.

Nota: Alternativamente, puede seleccionar la rama en el menú desplegable "Branch:" y seleccionar la rama que acaba de empujar hacia arriba. Tú, entonces tendrás una solicitud de “Pull Request” y el botón “Comparar” en el lado derecho.

MinesJA / github_guide

Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

A guide to github

3 commits 2 branches 0 releases 1 contributor

Your recently pushed branches:

create_user (11 minutes ago) Compare & pull request

Branch: create_user New pull request

Create new file Upload files Find file Clone or download

This branch is 1 commit ahead of master.

MinesJA Created User Latest commit a5d1fd3 11 minutes ago

File	Commit	Time
app	Created User	11 minutes ago
bin	second commit	an hour ago
config	second commit	an hour ago
db	Created User	11 minutes ago
lib	second commit	an hour ago
log	second commit	an hour ago

Haz clic en “Compare & pull request” y esto le llevará a la página de la petición de "Abrir un pull request". A partir de aquí, deberías escribir una breve descripción de lo que realmente cambiaste. Y deberías hacer clic en la pestaña "Reviewers". y seleccionar a quien tu equipo decidió que fuera el “Merge Master”. Cuando hayas terminado, haz clic en "Crear la solicitud de tirar".

MinesJA / github_guide

Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

base: master compare: create_user ✓ Able to merge. These branches can be automatically merged.

Created User

Write Preview

Added a user model, migration, and created a hello_world method.

Request a review

Filter people

ashleecrusco Ashlee Crusco

Labels

None yet

Projects

None yet

Milestone

No milestone

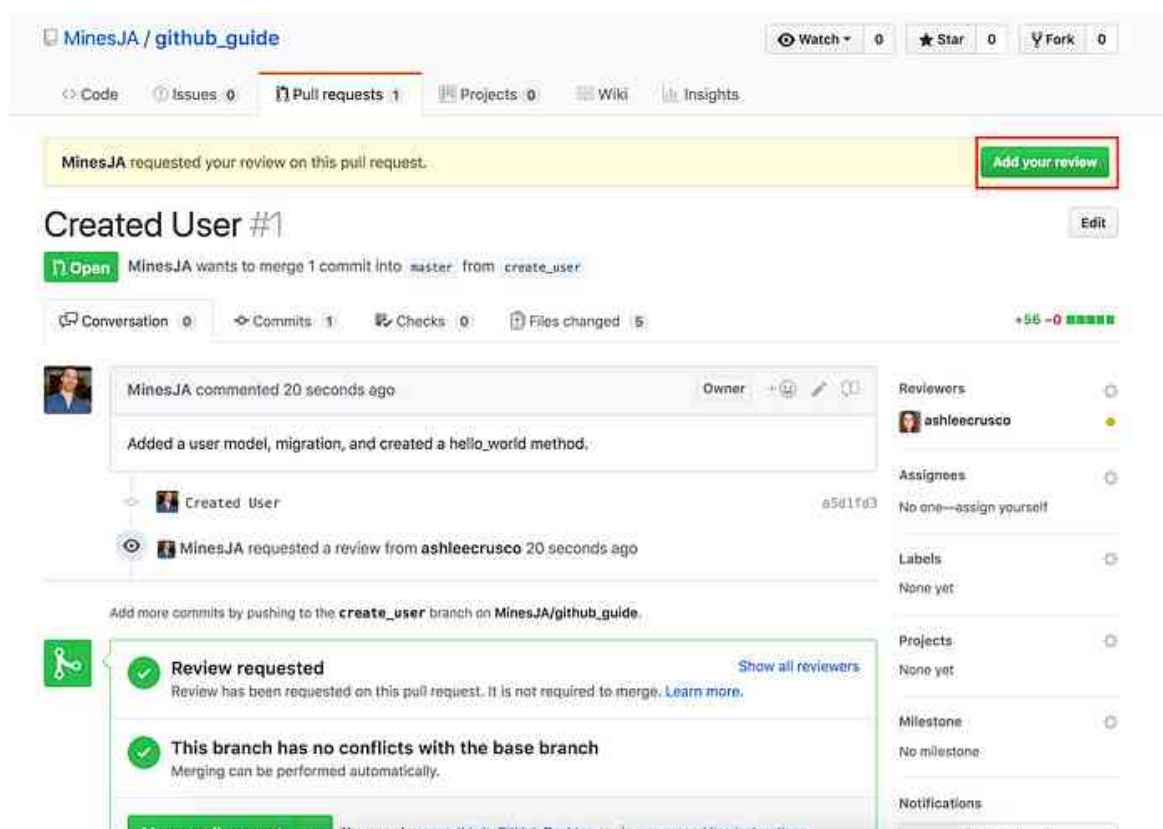
Create pull request

Paso 3c: Fusión de las solicitudes de Pull Request

Tenga en cuenta que si eres un colaborador, puedes fusionar tus propias solicitudes. Pero, de nuevo, si estás trabajando en un equipo, tiene más sentido tener a una persona haciendo toda la fusión y que todos los demás envíen peticiones y asignen al "Merge Master", como revisor para resolver los conflictos en la fusión.

Por lo tanto, asumiendo que usted es el encargado de cuidar de todos los merges y alguien le ha asignado como Revisor en una solicitud de los Pull, cuando usted entra en su Github usted notará que tiene una notificación que le hace saber que alguien le ha asignado como revisor. También te has dado cuenta de una barra amarilla que indica a uno de tus compañeros de equipo mientras le pides tu opinión sobre esta petición de Pull.

Adelante, haga clic en el botón Agregar tu opinión.



Esto le llevará a la página Pull Request. Si trabajas juntos usaría todas las mañanas para sentarme en equipo y pasar por solicitudes de Pull Request juntos. Si haces eso, no tendrás que dejar necesariamente comentarios detallados.

Sin embargo, si usted está trabajando de forma remota, esta será su herramienta principal para hacerle saber al solicitante si necesita hacer cambios o si va a fusionar su solicitud.

MinesJA / github_guide

Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 1 Projects 0 Wiki Insights

Created User #1

Open MinesJA wants to merge 1 commit into master from create_user

Conversation 0 Commits 1 Checks 0 Files changed 5

Changes from all commits Jump to... +56 -0

```

8 app/models/user.rb
... @@ -0,0 +1,8 @@
1 +class User < ApplicationRecord
2 +
3 +   def hello_world
4 +     puts "Hello world!"
5 +   end
6 +
7 +
8 +end

10 db/migrate/20180529191921_create_users.rb
... @@ -0,0 +1,10 @@
1 +class CreateUsers < ActiveRecord::Migration[5.2]
2 +   def change
3 +     create_table :users do |t|
4 +       t.string :name
5 +       t.integer :age

```

Diff settings Review changes

Submit your review

Review summary

Leave a comment

Comment
Submit general feedback without explicit approval.

Approve
Submit feedback and approve merging these changes.

Request changes
Submit feedback that must be addressed before merging.

Submit review

Cuando haga clic en la “Submit review” de “Review changes” puedes

ashleecrusco approved these changes a minute ago

View changes

Lock conversation

Looks good to me!

ashleecrusco assigned MinesJA 29 seconds ago

Add more commits by pushing to the create_user branch on MinesJA/github_guide.

Review requested

Review has been requested on this pull request. It is not required to merge. [Learn more.](#)

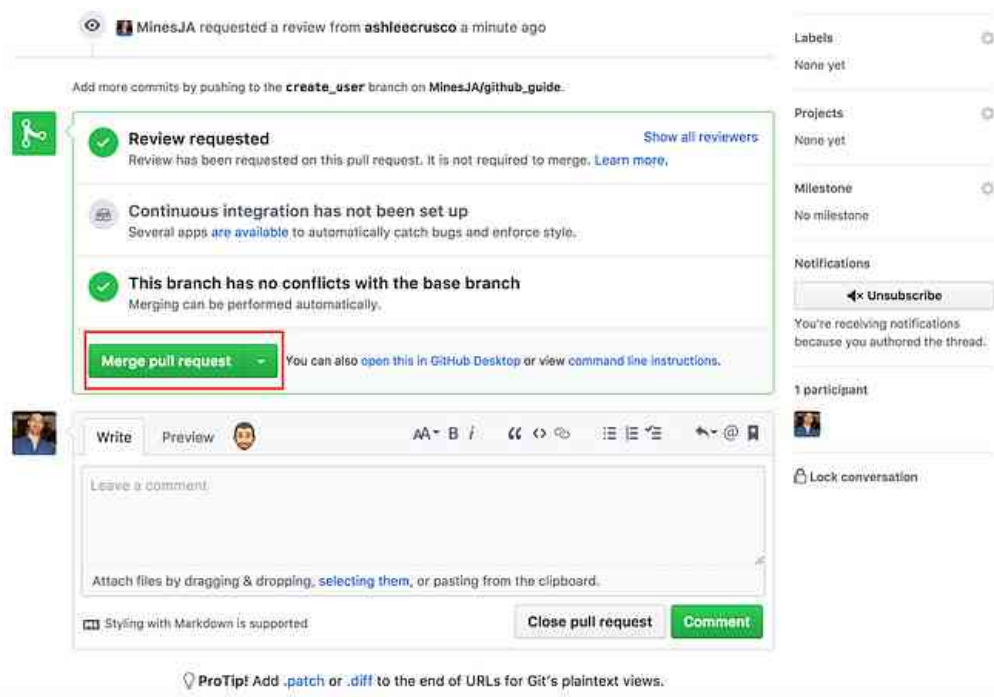
Continuous integration has not been set up

Several apps are available to automatically catch bugs and enforce style.

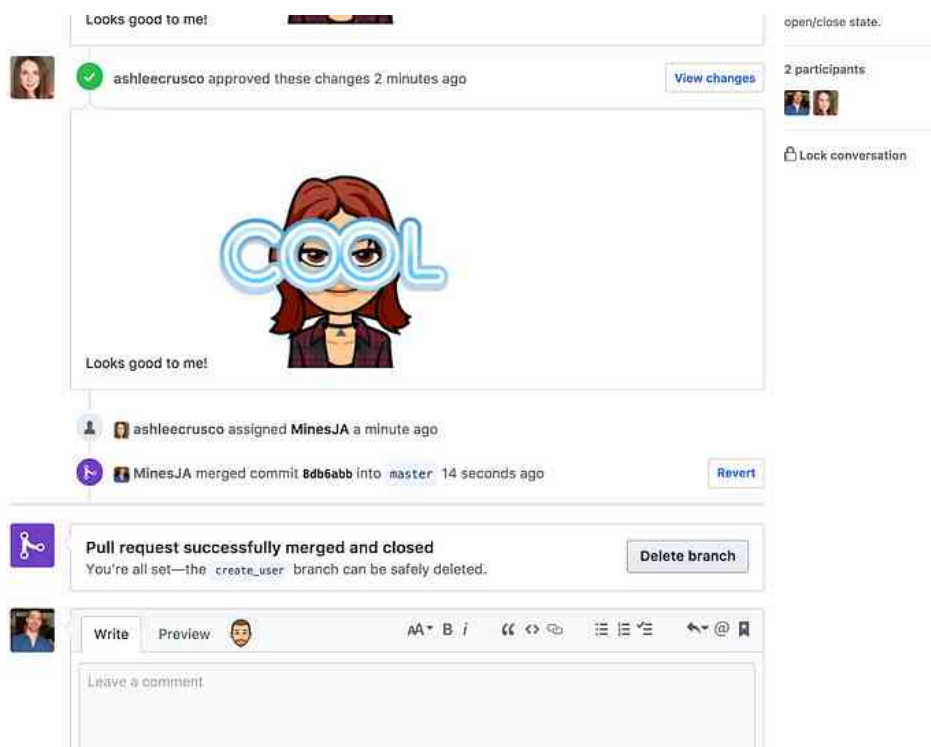
This branch has no conflicts with the base branch

Merging can be performed automatically.

Cuando esté satisfecho con la solicitud de Pull request, vaya a la parte inferior de la solicitud de Pull y haga clic en la solicitud de “Merge and Pull request”.



A continuación, verá una solicitud de Pull Request y mensaje cerrado y un botón a la rama de la que debe hacer clic.



Paso 4: Enjuague, Repita

Y eso es más o menos. Sigue añadiendo nuevas ramas para nuevas funciones y luego reune como equipo para fusionarlas en el Master. Mantenga el Master limpio y desplegable y no trate de fusionar más de una rama a la vez.