

### UNIVERSIDAD DON BOSCO FACULTAD DE INGENIERÍA

### ANALISIS Y DISEÑO DE SISTEMAS

### DESAFIO 1 - GITHUB

### PRESENTADO POR:

### Ayala Zelaya, Alejandro René AZ160404 Rebecca Gabrielle Pineda Molina PM141471 Franklin Alejandro López Ramírez LR192720

### David Guillermo Ramírez RR192745

### Catedrático:

### ING. Henry Porfirio Avalos Cardoza

### Soyapango San Salvador, 02/05/2020

Índice

[Introducción 1](#_Toc39280788)

[Objetivos 2](#_Toc39280789)

[Marco teórico 3](#_Toc39280790)

[1. ¿Qué es Git? 3](#_Toc39280791)

[2. Objetos de Git 3](#_Toc39280792)

[3. ¿Qué es GitHub? 4](#_Toc39280793)

[4. ¿Qué es una metodología de desarrollo de software en el mundo de la programación y de qué forma vamos a aplicarla? 4](#_Toc39280794)

[5. Git&GitHub, Setup 5](#_Toc39280795)

[6. Set Up clave SSH 8](#_Toc39280796)

[7. Set Up Config 9](#_Toc39280797)

[8. Crear – Clonar un repositorio git 9](#_Toc39280798)

[9. Realizar cambios, revertirlos y subirlos al repositorio remoto 10](#_Toc39280799)

[Conclusiones 17](#_Toc39280800)

[Anexos 18](#_Toc39280801)

[Bibliografía 19](#_Toc39280802)

# Introducción

El presente trabajo tiene como finalidad informar, lo esencial referente a Git y GitHub como herramienta para la coordinación de un trabajo en equipo, para la realización de un software.

Para ello, se utilizó una estructura sencilla, que ayudará al lector, a comprender de la manera más clara y precisa, la información relacionada a la conceptualización de Git y GitHub, además de su configuración inicial y posterior uso para el desarrollo de software.

En primer lugar, se plantea una parte aclaratoria o de definición de conceptos, donde damos definiciones y datos importantes que debemos conocer para proceder con la parte aplicada, es decir, la parte teórica o de conocimientos previos.

Posteriormente nos adentramos a una parte más práctica, donde ejemplificamos y mostramos partes concretas relacionadas al tema, con ejemplos propios y material de apoyo visual.

Todo esto bajo la idea de un aprendizaje completo que te lleve desde no conocer la temática, hasta un dominio que te permita adéntrate y empezar a examinar.

# Objetivos

* **Objetivo general:** Dominar el uso de GitHub para el desarrollo del proyecto de catedra.
* **Objetivo específico 1:** Conocer los conceptos esenciales para el manejo de Git y GitHub
* **Objetivo específico 2:** Generar una guía completa, pero entendible para el uso de GitHub
* **Objetivo específico 3:** Aplicar los conocimientos en un ejemplo práctico.

# Marco teórico

## ¿Qué es Git?

Es un software diseñado para el trabajo en equipo durante el desarrollo de software, a través del control de versiones, teniendo como prioridad la eficiencia y confiabilidad para mantener diversas versiones, a pesar de que se cuente con múltiples archivos en de código.

Actualmente es un software libre distribuido bajo la Licencia Publica General de GNU en su versión 2 y tiene contribuciones en su desarrollo por más de 200 programadores.

## Objetos de Git

Git es fundamentalmente un almacén de clave-valor. Cuando agrega datos a git, construye un object y usa el hash SHA-1 del contenido del object como una clave.

Por lo tanto, cualquier contenido en git puede ser buscado por su hash:

git cat-file -p 4bb6f98

Hay 4 tipos de Object:

* Blob: Es el tipo de objeto utilizado para almacenar los contenidos de cada archivo en un repositorio.
* Tree: Es el tipo de objeto que crea jerarquía entre diversos archivos en un repositorio Git.
* Commit: Es el tipo de objeto utilizado para capturar el estado actual de un proyecto, este no puede ser cambiado a menos que sea especificado directamente
* Tag: Las etiquetas son referencias que apuntan a puntos específicos en el historial de Git.

## ¿Qué es GitHub?

GitHub es una plataforma de desarrollo colaborativo para alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas de computador.

El código de los proyectos alojados en GitHub se almacena típicamente de forma pública, aunque utilizando una cuenta de pago, también permite hospedar repositorios privados.

Desde enero de 2010, GitHub opera bajo el nombre de GitHub, Inc. Anteriormente era conocida como Logical Awesome LLC, el 4 de junio de 2018, Microsoft compró GitHub por la cantidad de 7.500 millones de dólares.

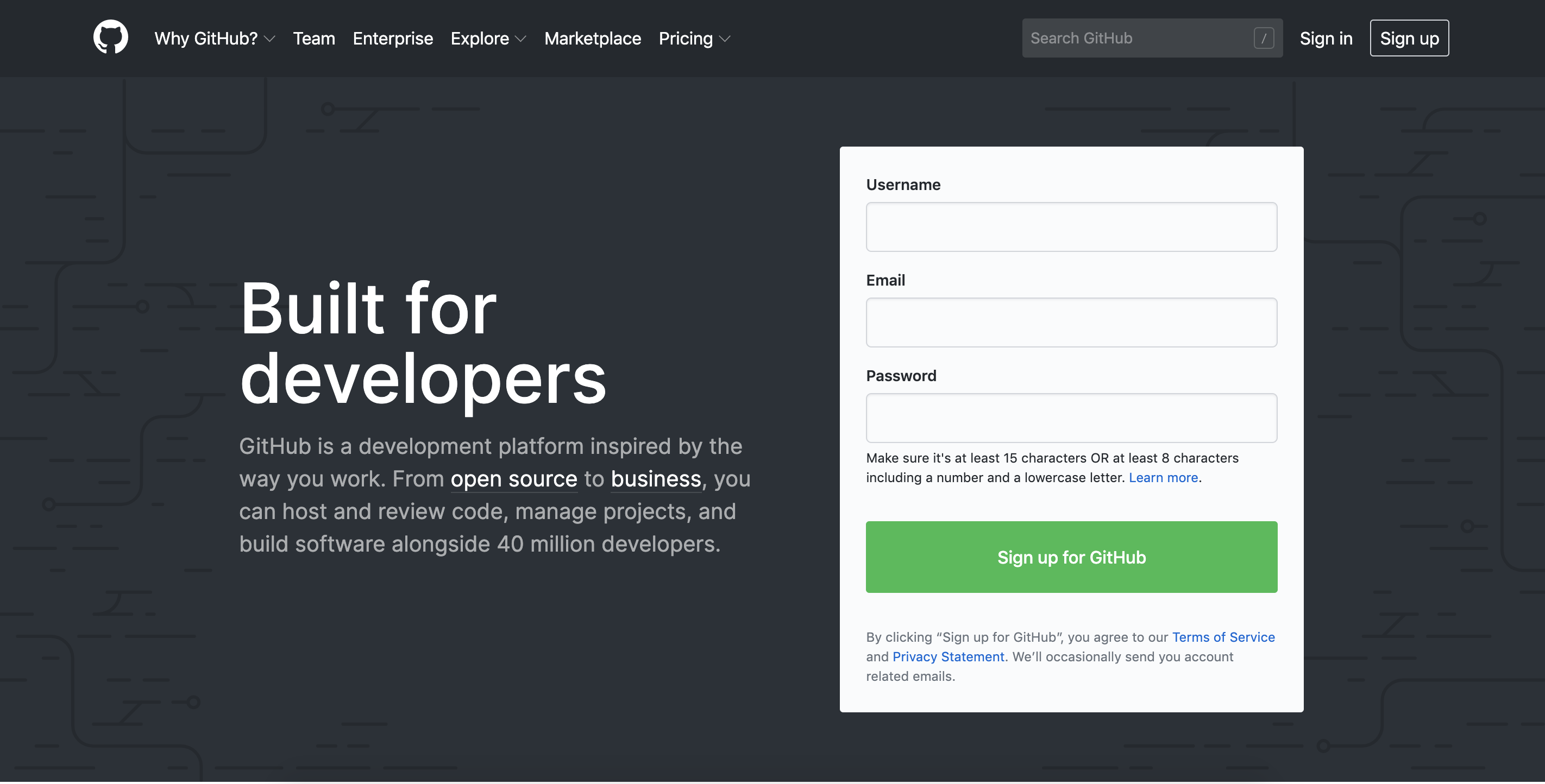
## ¿Qué es una metodología de desarrollo de software en el mundo de la programación y de qué forma vamos a aplicarla?

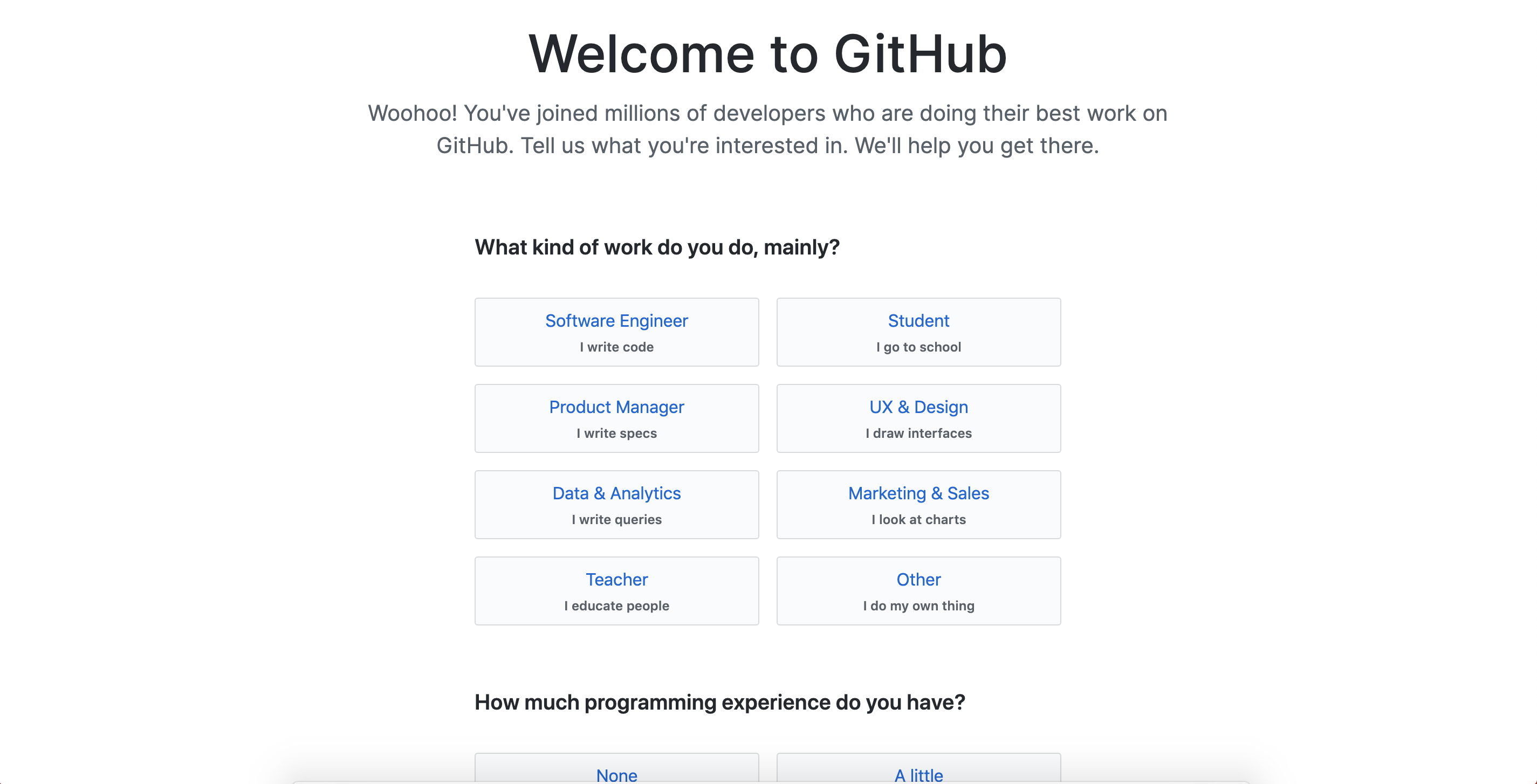
Metodología de desarrollo de software, en ingeniería de software, es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información.

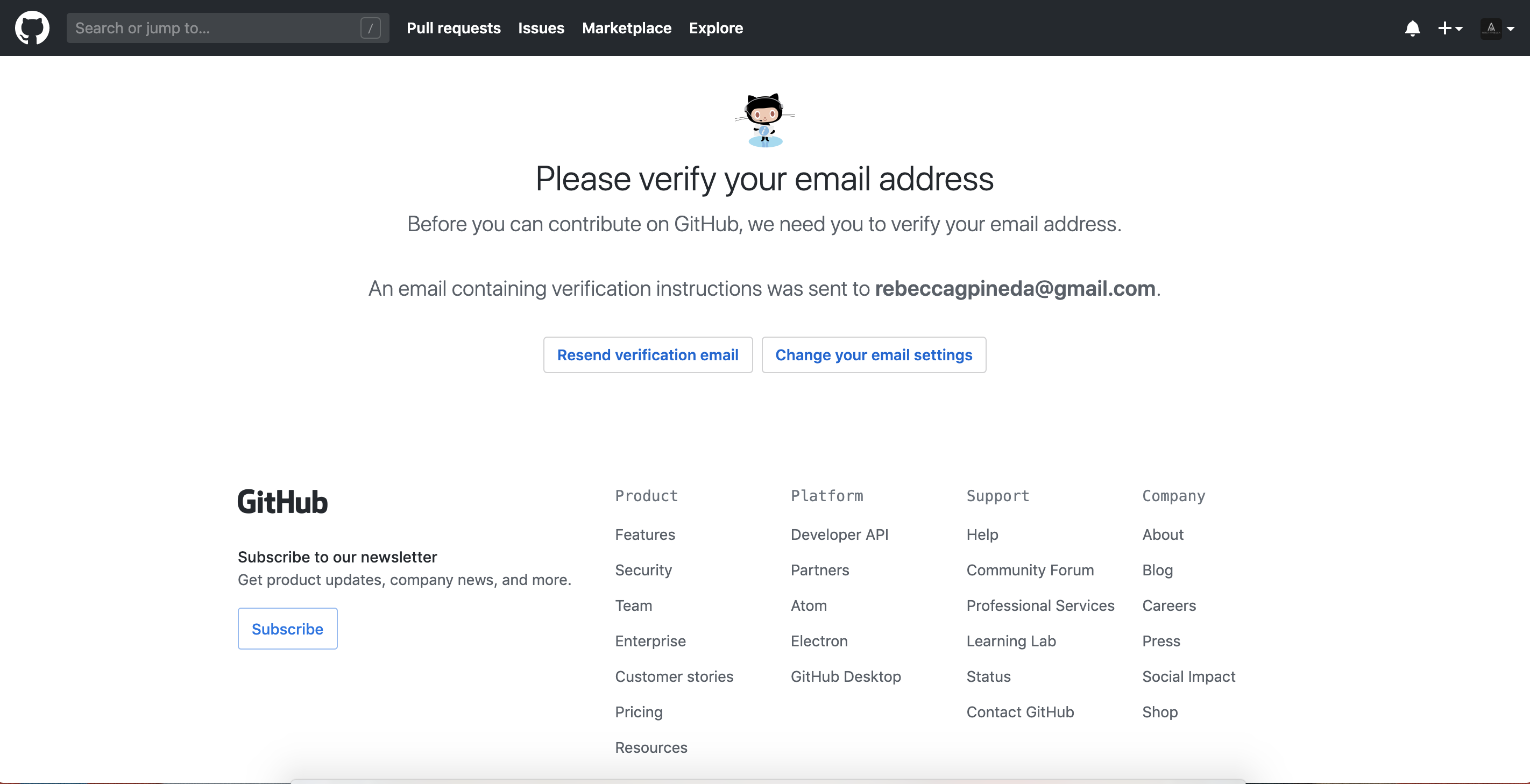
El objetivo es controlar el proceso de desarrollo para que sea ordenado y de esta forma lo más productivo posible.

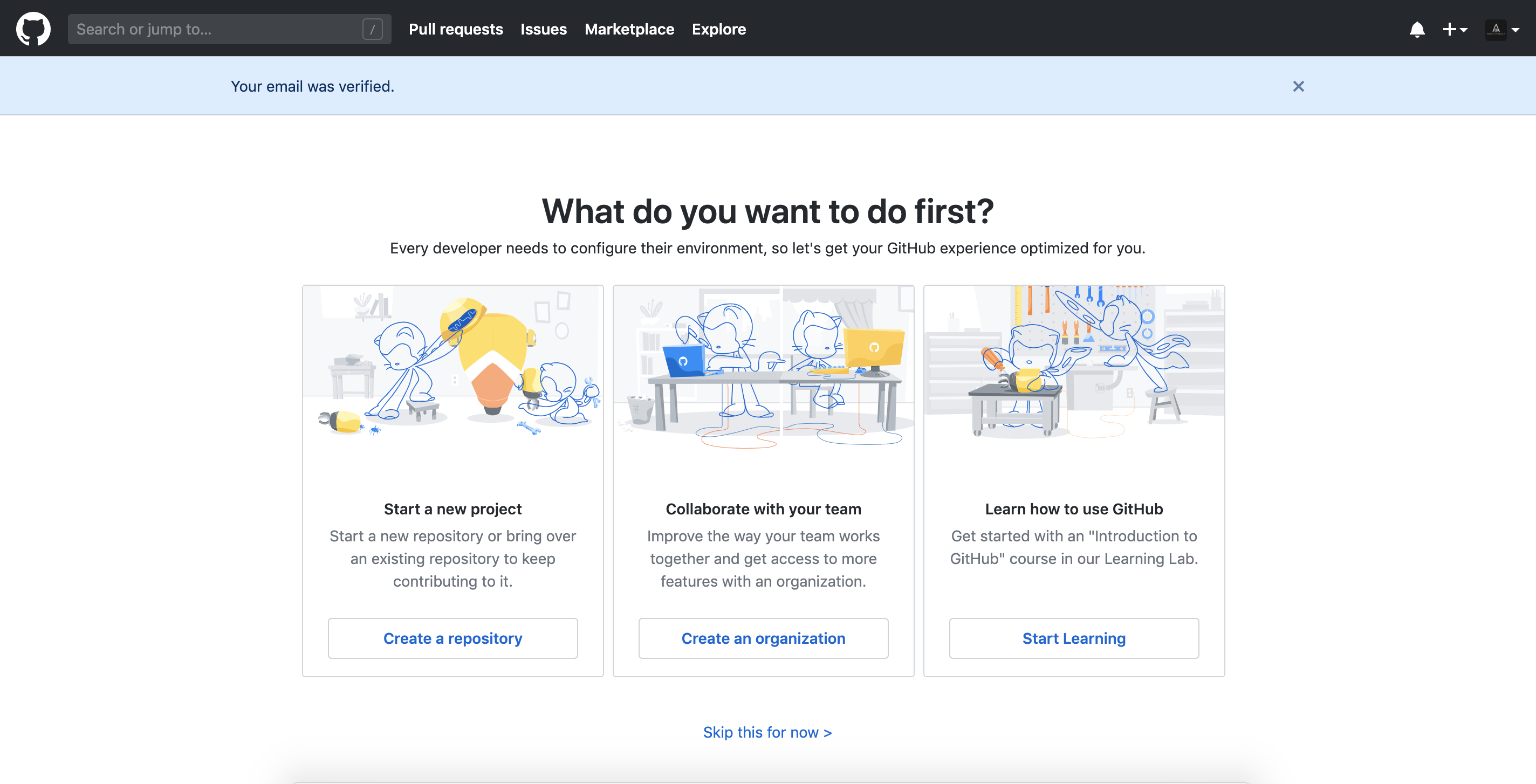
Este método de desarrollo estaría basado en Git y GitHub ofreciéndonos una enorme versatilidad y posibilitando que todo el equipo de desarrollo esté al día en cuanto a novedades en el desarrollo, así como multitud de ventajas. Esta metodología estaría basada en las branches y pull request.

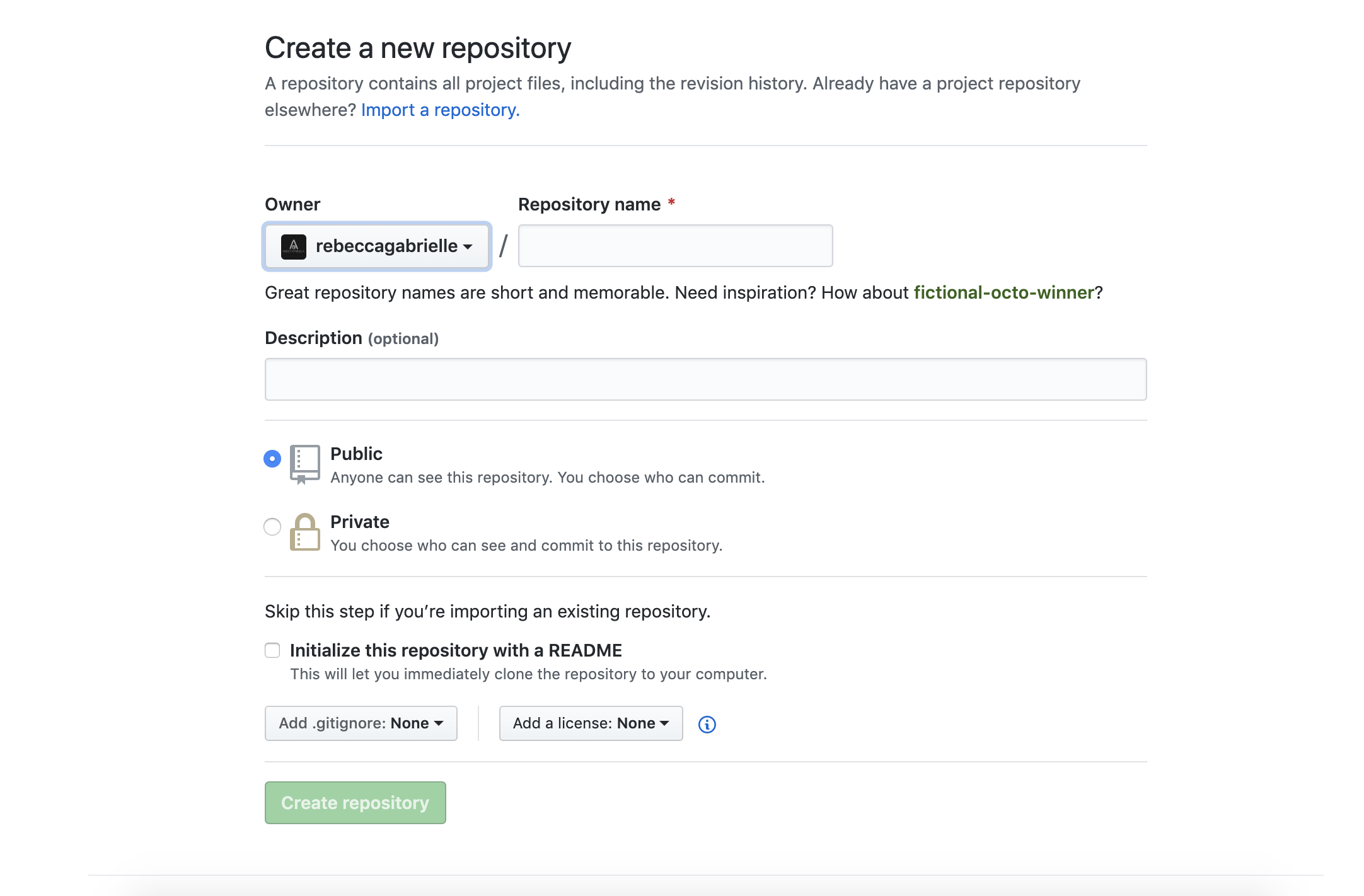
## Git&GitHub, Setup

1. Entrar a github.com y crear una cuenta. 

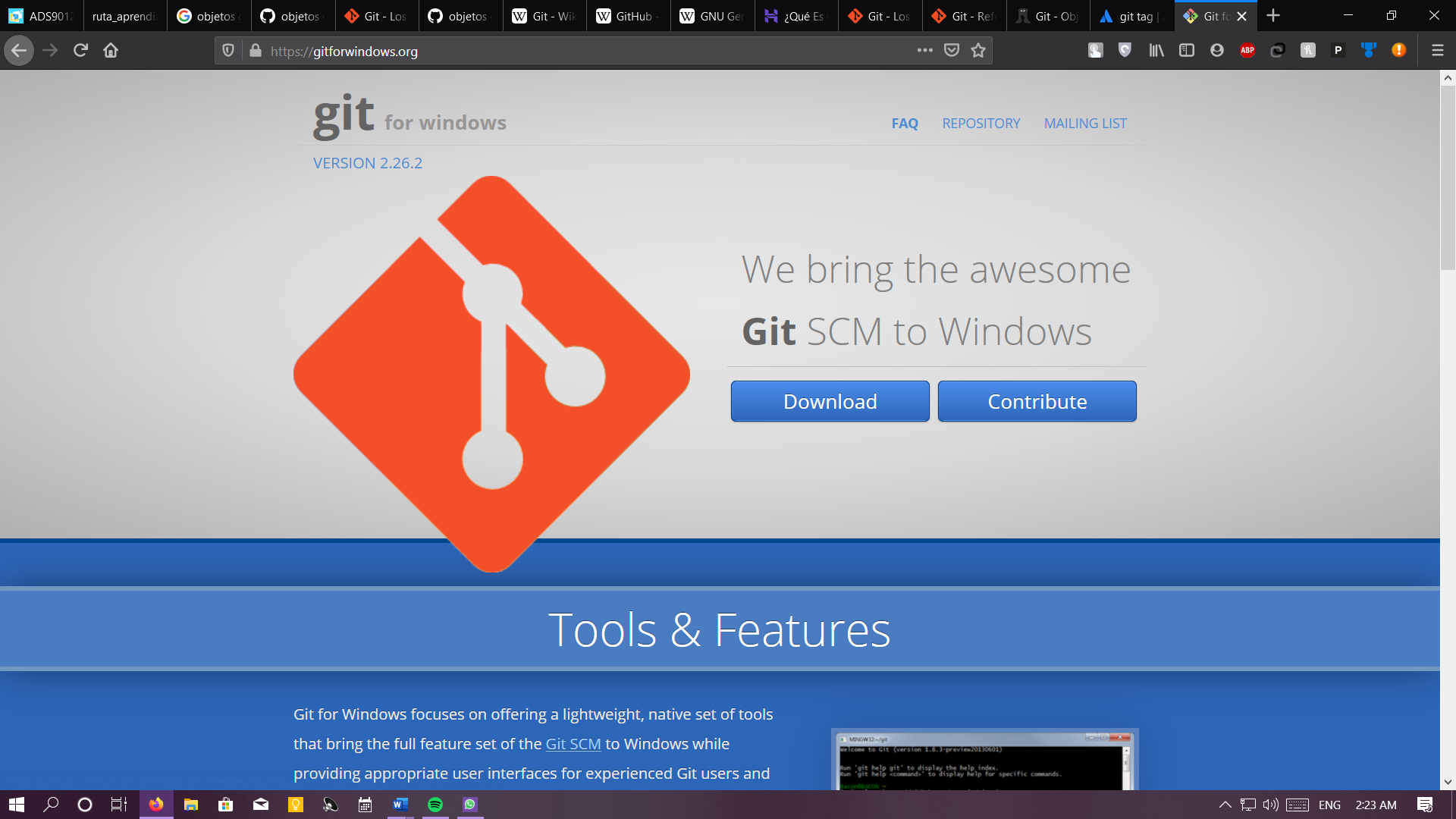
2. Personalizar tu experiencia, respondiendo las preguntas iniciales.

3. Verificar tu correo electrónico.

4. Elegir la opción crear un repositorio.

5. Crear repositorio

1. El siguiente paso será instalar git.



Git es un programa que se instala en nuestro pc, pero hay que tener en cuenta que no todos los programas tienen una interfaz gráfica (gui o Graphical User Interface). En el caso de git en mac no hay gui (aunque sí hay programas externos que permite manejar git a través de una gui) y en el caso de windows git sí viene incluido con una gui. Sin embargo, tanto en mac como en windows, no es recomendable trabajar con git mediante una interfaz gráfica. En vez de eso se trabajará mediante línea de comandos por terminal (en mac) / cmd (en windows).

## Set Up clave SSH

Como contexto hay que definir lo que es una clave SSH y como nos puede ayudar usando Git.

Las claves SSH son una forma de acceso continuo sin necesidad de una contraseña a un servidor. Este método a diferencia del método tradicional por contraseña permite realizar conexiones automáticas sin tener que enviar ninguna contraseña, limitando así la posibilidad de robo.

Git usa claves SSH para la seguridad del contenido gestionando en una empresa y realiza conexión automática a las personas con acceso.

El par de claves SSH son generadas por un programa nativo en el sistema operativo ssh-keygen (Linux), MSysGit (Windows). Ejemplo de creación de claves:

$ ssh-keygen

Generating public/private rsa key pair.

Enter file in which to save the key (/home/schacon/.ssh/id\_rsa):

Created directory '/home/schacon/.ssh'.

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in /home/schacon/.ssh/id\_rsa.

Your public key has been saved in /home/schacon/.ssh/id\_rsa.pub.

The key fingerprint is:

d0:82:24:8e:d7:f1:bb:9b:33:53:96:93:49:da:9b:e3 schacon@mylaptop.local

Una vez Generada la clave publica esta debe mandarse a todos los que administren el servidor git.

## Set Up Config

La configuración principal de git es muy importante para identificar variables operativas o de gestión del usuario, estas se cambian con el comando “Set Up Config” el comando más usado que veremos es la configuración de nuestra identidad.

$ git config --global user.name "John Doe"

$ git config --global user.email johndoe@example.com

Para ver todos los posibles comandos a editar git nos ayuda desplegando la lista de ellos con el comando:

git config –list

## Crear – Clonar un repositorio git

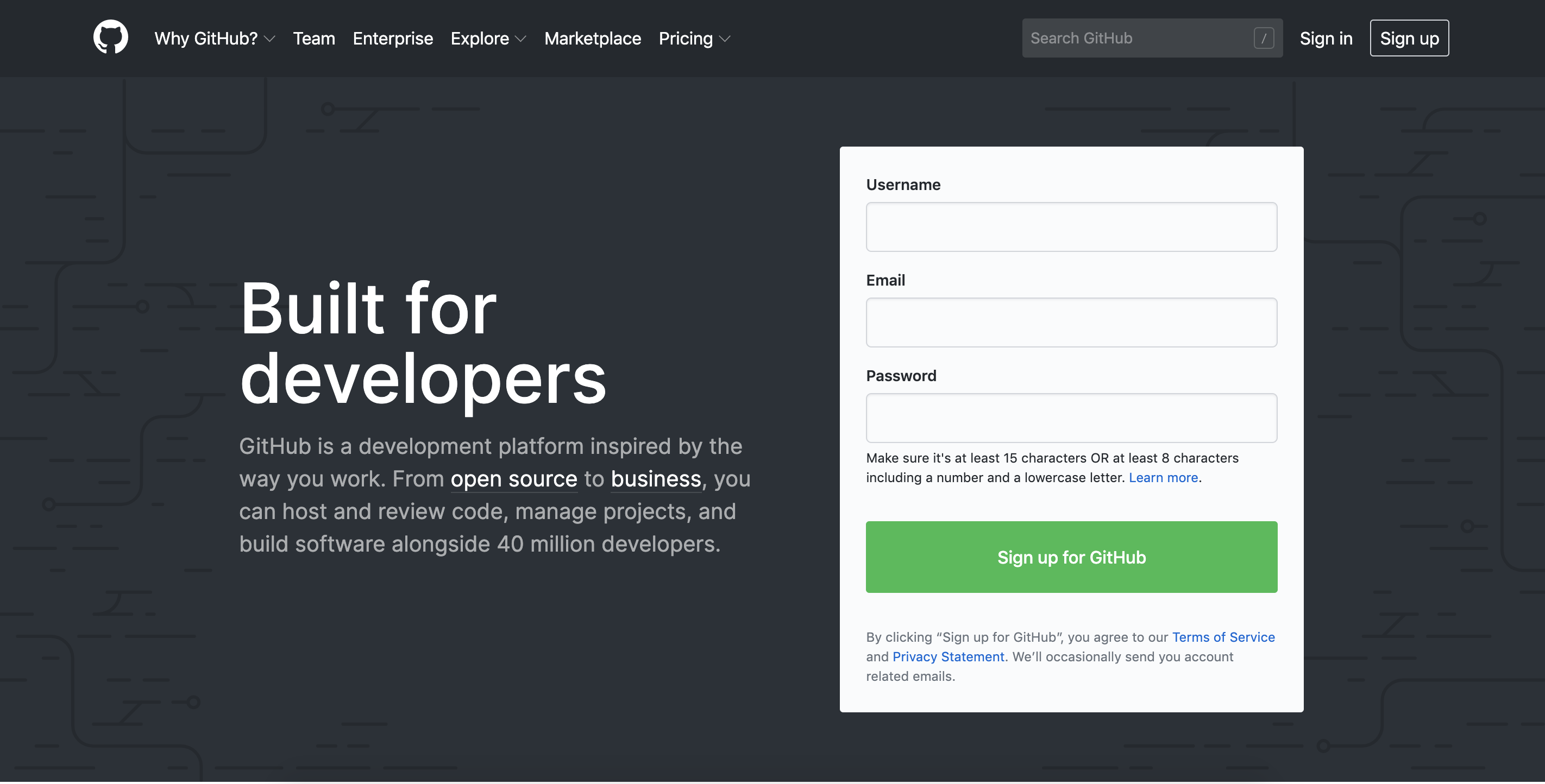
La creación de un repositorio se hace con el comando “Git init” este automáticamente crea un nuevo proyecto y una nueva ramificación master.

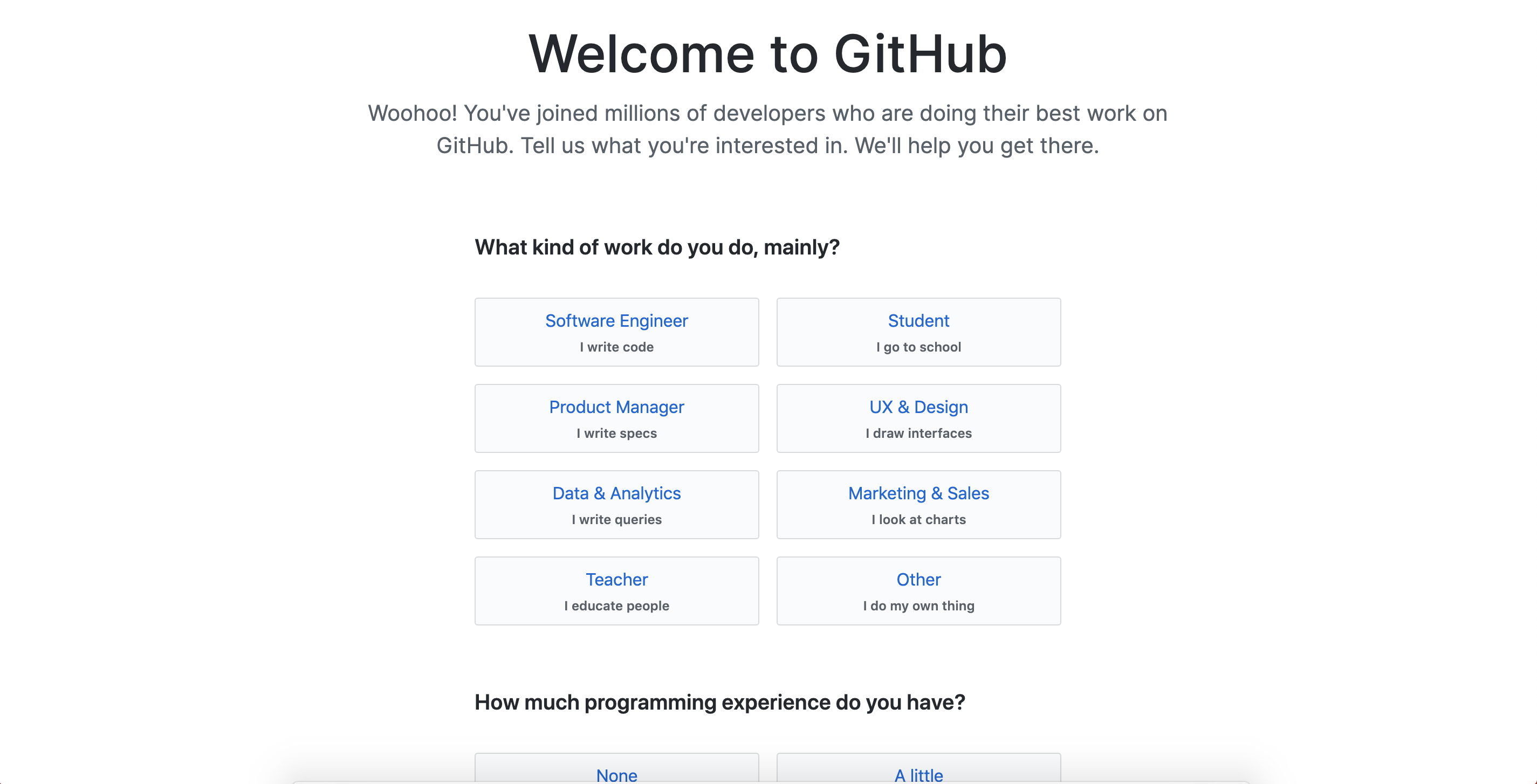
La clonación de un repositorio git nos permite obtener todos los archivos subidos a una rama de un repositorio ya existente el comando que nos permite esto es:

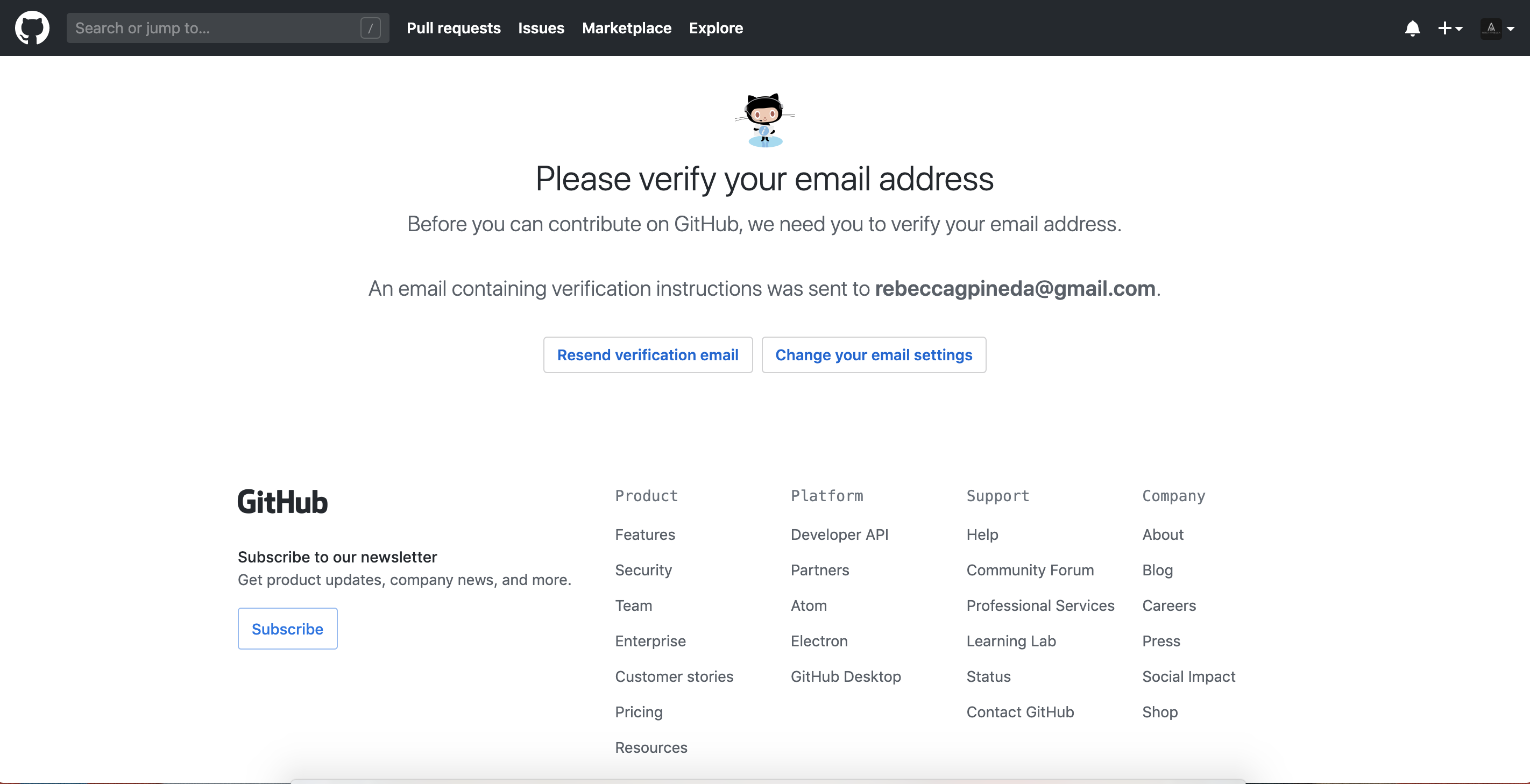
“git clone \*\*\*Dirección de repositorio que queremos clonar \*\*\*”

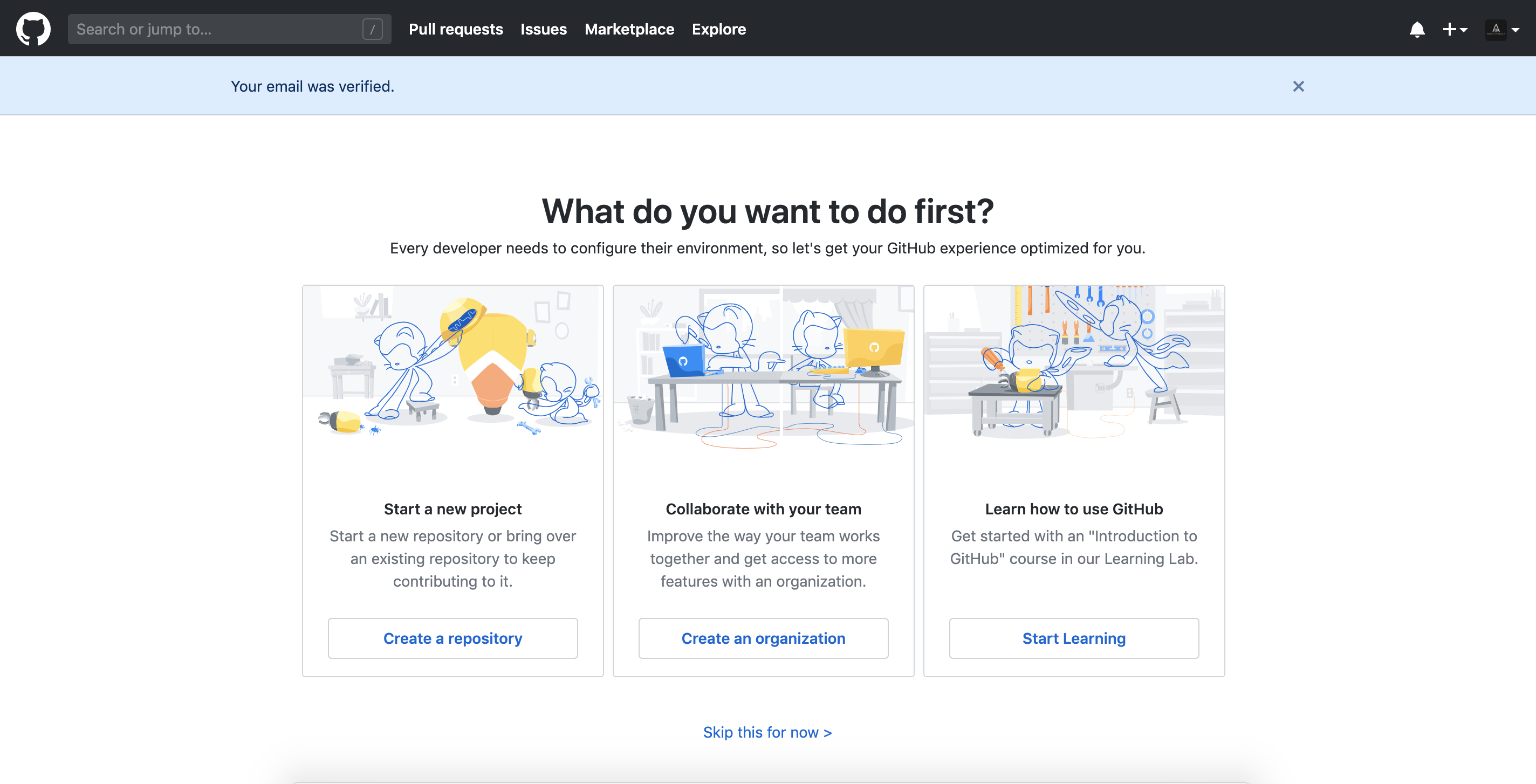
Ejemplo de repositorio: <https://github.com/rebeccagabrielle/Webproject>

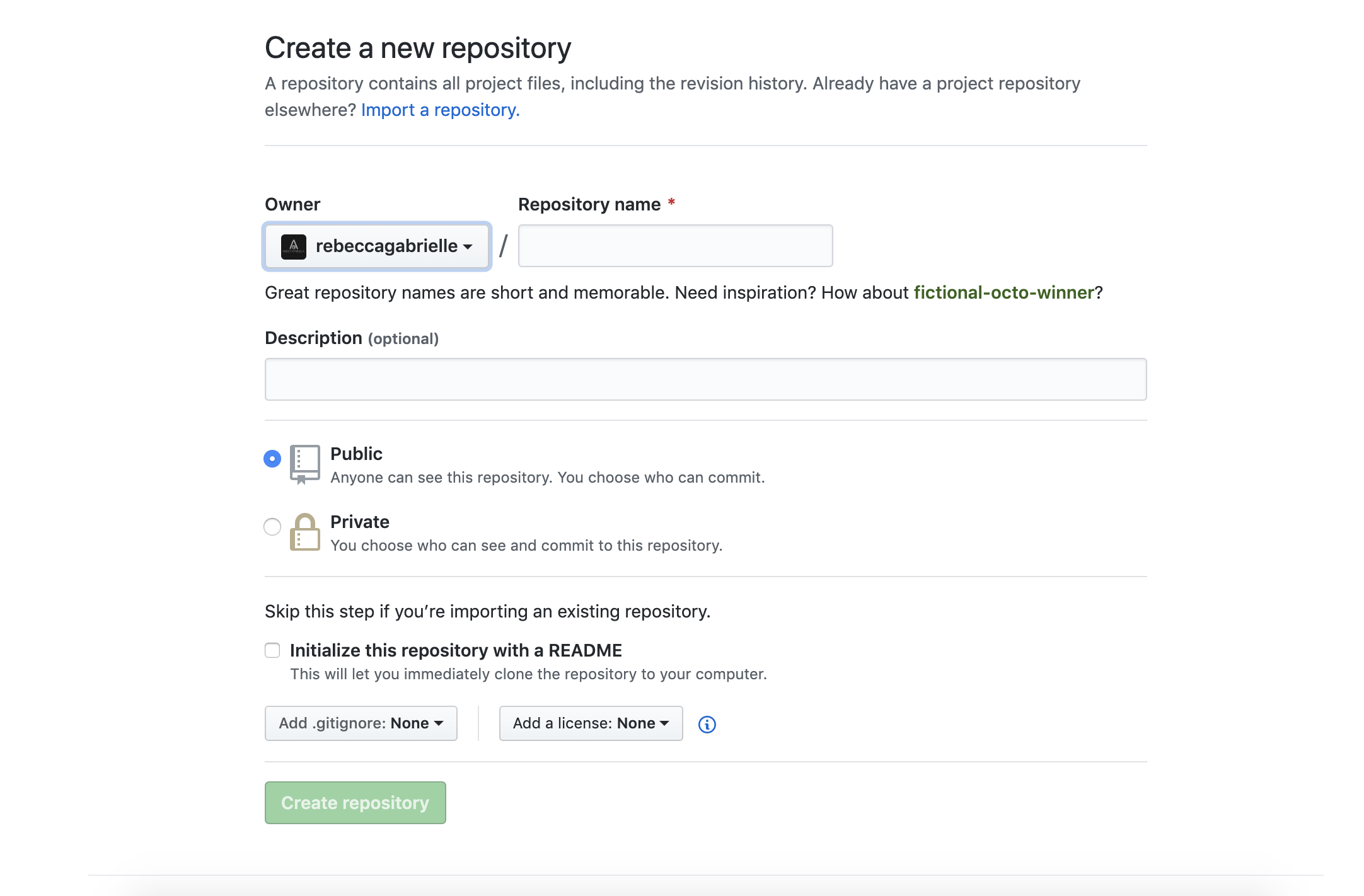
## Realizar cambios, revertirlos y subirlos al repositorio remoto

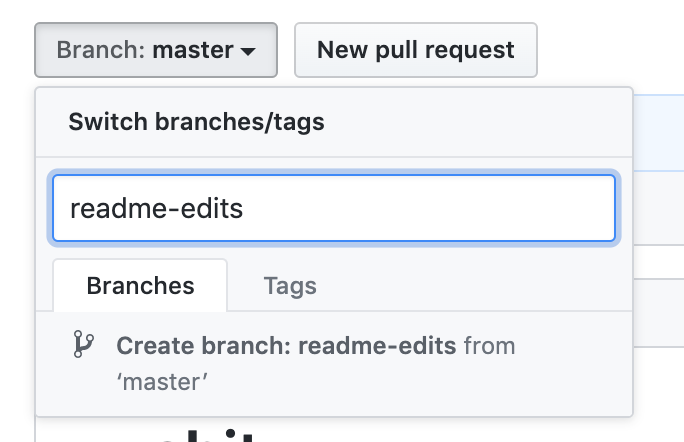
1. Entrar a github.com y crear una cuenta. 

2. Personalizar tu experiencia, respondiendo las preguntas iniciales.

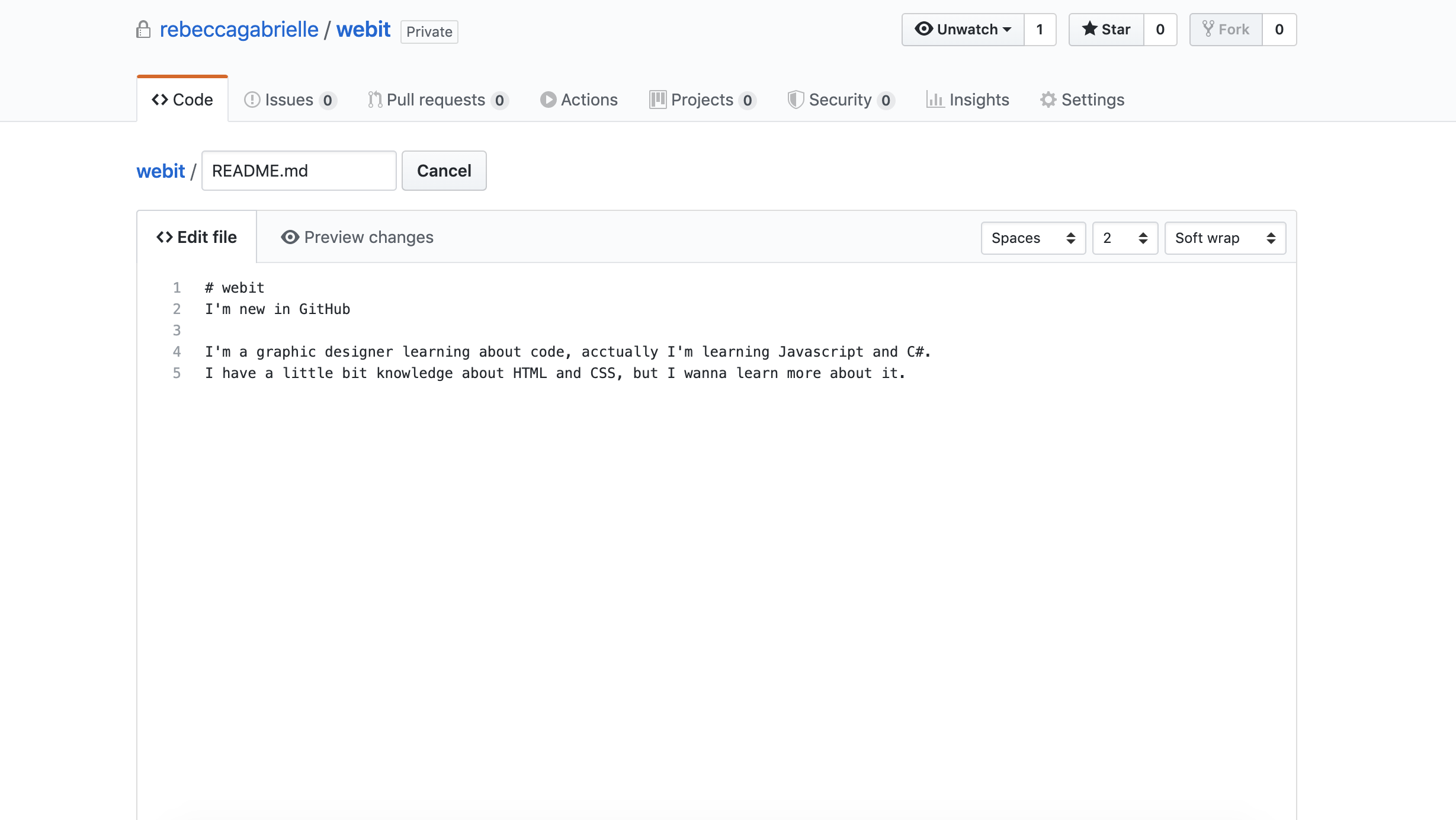
3. Verificar tu correo electrónico.

4. Elegir la opción crear un repositorio.

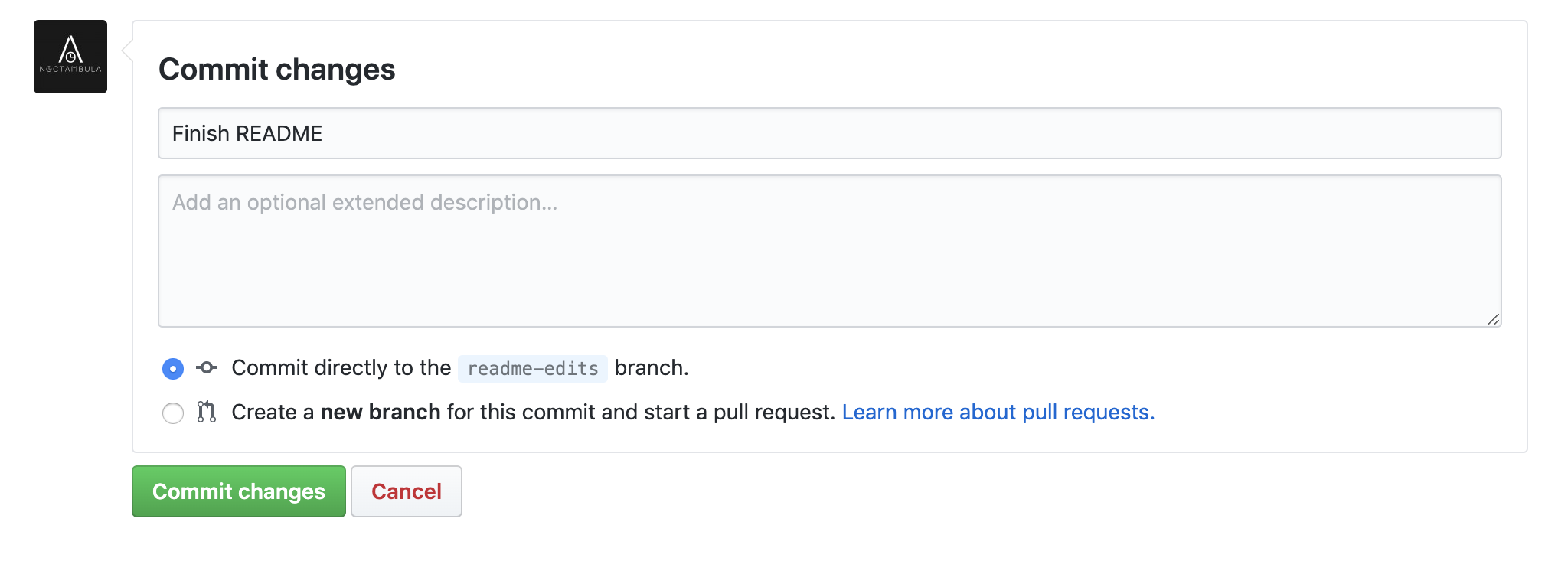
5. Crear repositorio

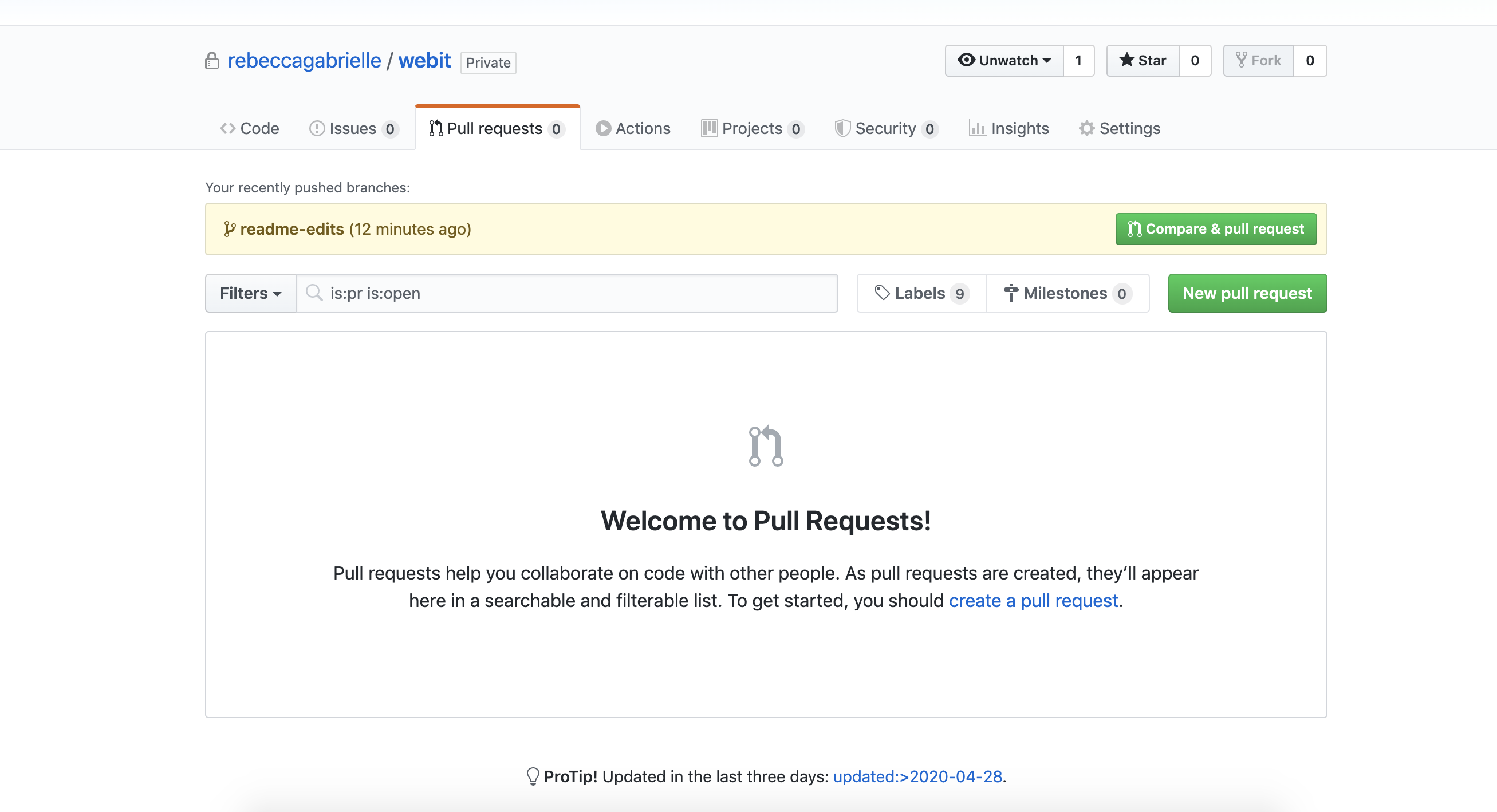
6. Crear una nueva rama para poder realizar un commit.

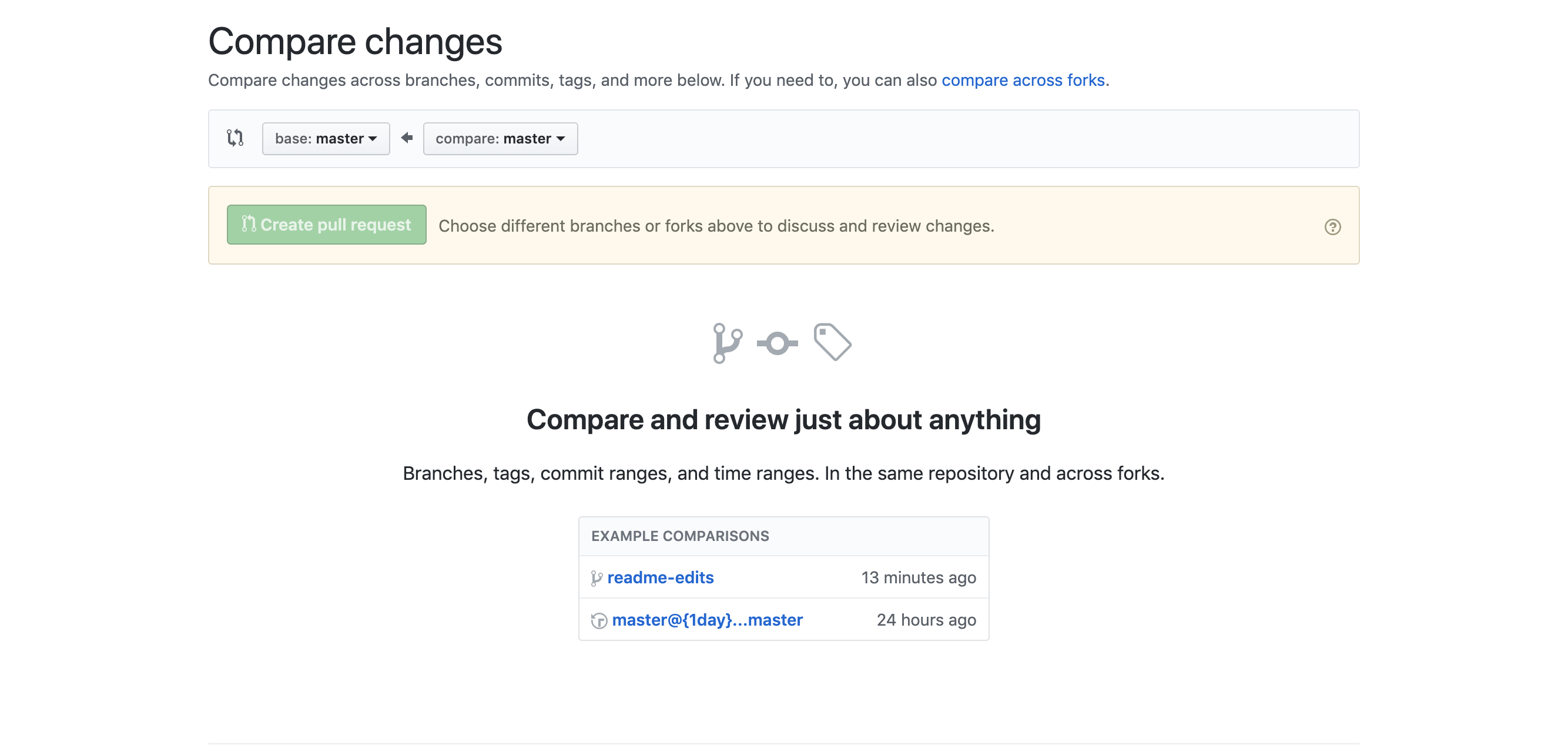
7. Editar el archivo en la nueva rama creada.



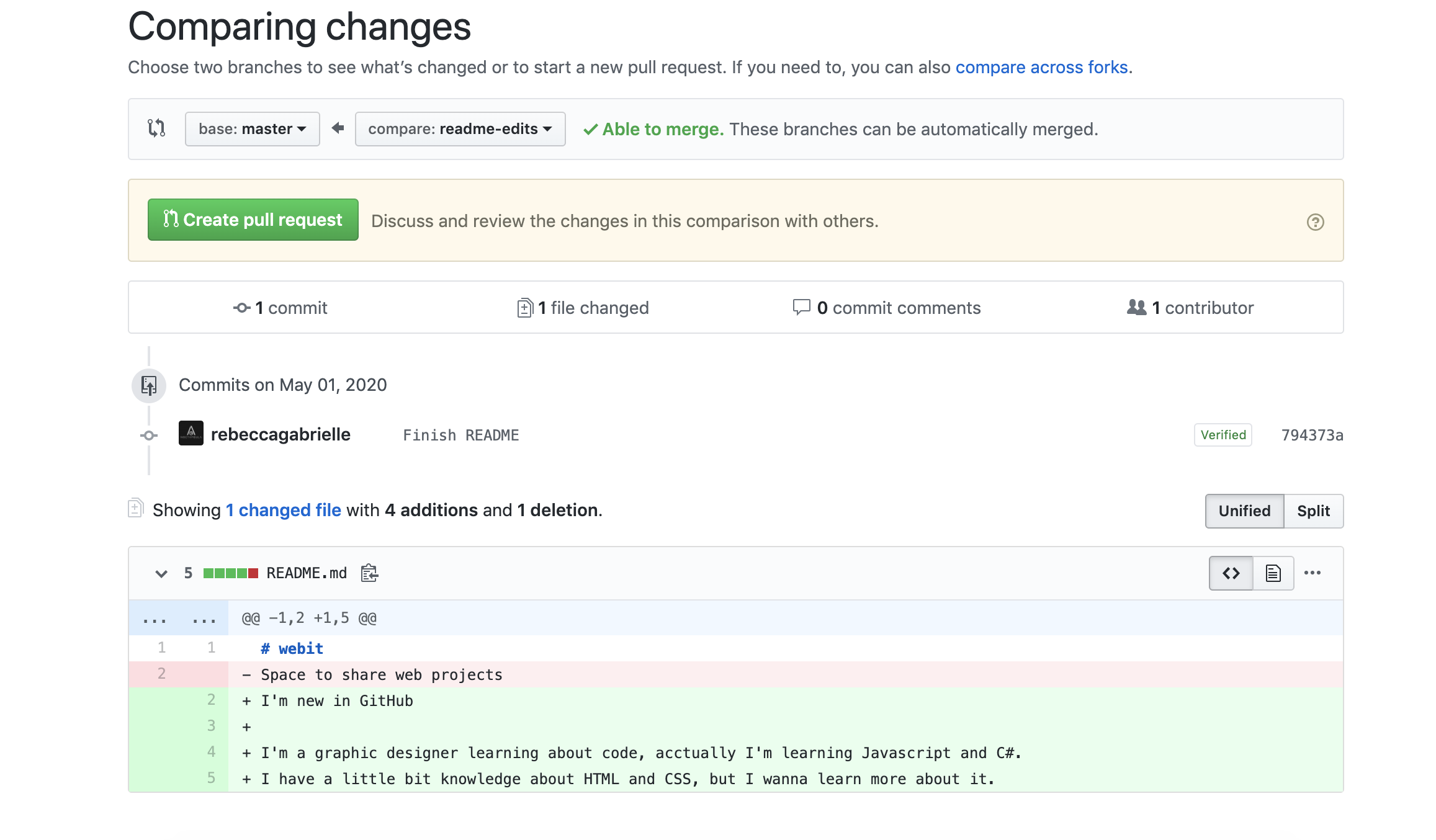
8. Realizar el commit de los cambios (es decir salvar los cambios en la nueva rama).

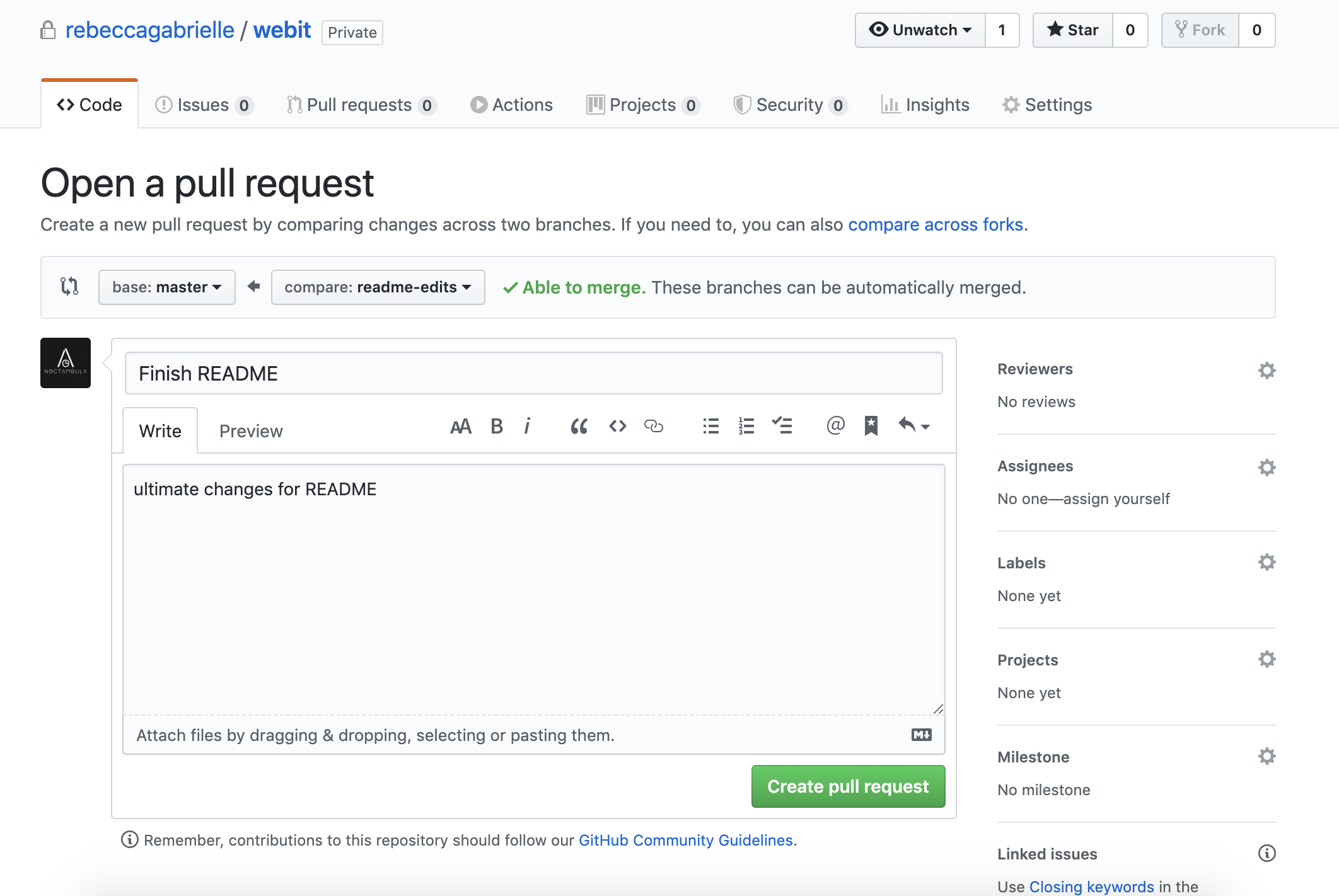


9. Realizar un pull request de los cambios realizados.

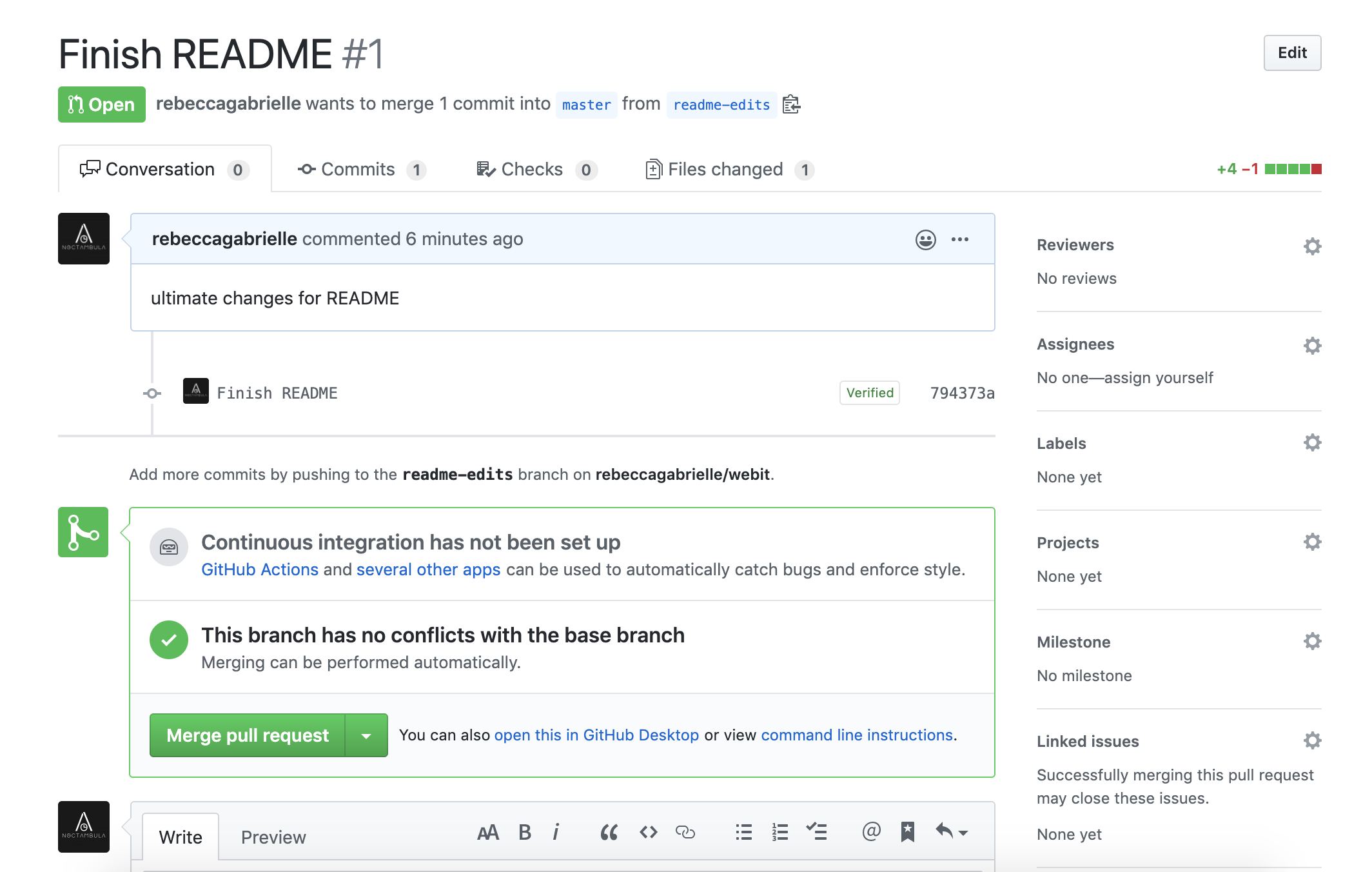
 10. Elegir la rama que queremos comparar con la rama master.

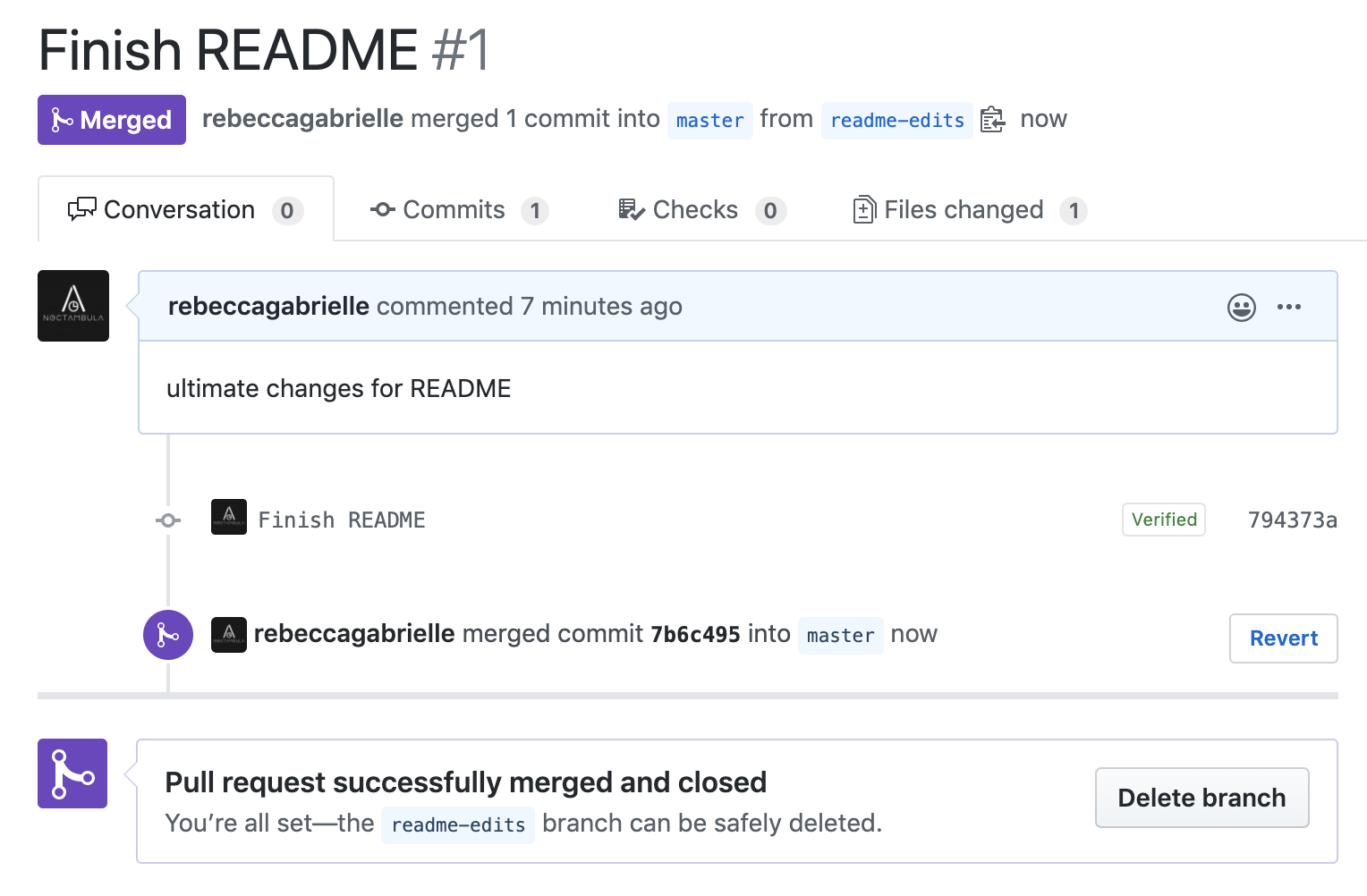
11. Comparar los cambios.



12. Cuando estamos de acuerdo con los cambios del commit, realizar el pull request.

13. Realizar merge del pull request a la rama master.



14. Una vez realizado el merge, eliminar la rama creada.

# 

# Conclusiones

El conocimiento de Git y GitHub en la era moderna es esencial para cualquier desarrollador de software, en primer lugar, por la popularidad que posee, por lo cual es muy posible que en algún momento debamos trabajar bajo estas herramientas, ya sea por las disposiciones de nuestro lugar de trabajo o por la comodidad y familiaridad de nuestros compañeros en los proyectos, convirtiendo en un conocimiento esencial para el desarrollo en equipo.

Además, es innegable la funcionalidad y facilidad que ofrece, con un manejo relativamente sencillo que no tiene una curva de aprendizaje demasiado elevado, es posible dominarlo en poco tiempo e incorporarlo a nuestro flujo de trabajo con facilidad.

Finalmente, basándonos en lo anterior, podemos concluir, que este aprendizaje es algo esencial para nuestro crecimiento como profesionales en el área de sistemas y desarrollo de software.

# Anexos

# Bibliografía

1. B., G. (2019, mayo 13). ¿Qué es GitHub y para qué se utiliza? Recuperado de <https://www.hostinger.es/tutoriales/que-es-github/>
2. Git - Objetos | git Tutorial. (s. f.). Recuperado 2 de mayo de 2020, de <https://riptutorial.com/es/git/example/8755/objetos>
3. colaboradores de Wikipedia. (2020, abril 18). Git. Recuperado de <https://es.wikipedia.org/wiki/Git>