



Git + Github

Marco Robles

@marcoantoniorob 
@MarcoRob 

Diego Castro

@diego-osorno-castro 



Marco Robles

- **Software Engineer** who loves working with JS or Java (any POO).
- Swimmer and MTB Biker (noob).
- **Contributor to Apache Beam** open-source project.



Diego Castro

- **Software Engineer with JAVA as main skill, AWS enthusiast.**
- Loves practicing outdoor sports (trail running, MTB)
- From "La tierra de la cajeta", Celaya, Guanajuato
- I don't like mole u_u

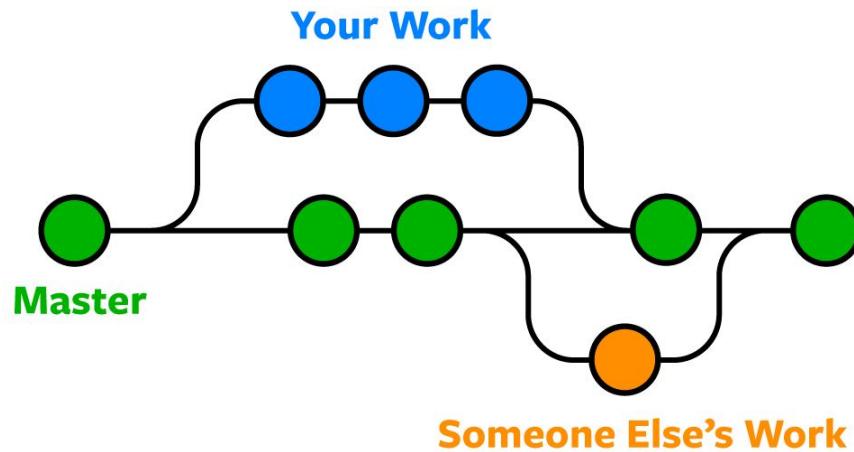
Contenido

- ¿Qué es Git? 
- Instalación 
- .gitignore 
- Comandos - Locales 
- Comandos - Remotos 
- GitHub 
- Pull Requests 
- Recursos 

■ ¿Qué es Git?

¿Qué es Git?

Git es controlador de versiones, diseñado para manejar desde proyectos pequeños hasta proyectos más grandes, con rapidez y eficiencia.



Contenido

¿Qué es Git?

Instalación

.gitignore

Comandos - Locales

Comandos - Remotos

GitHub

Pull Requests

Recursos

Instalación

 Instalación.

- **Homebrew**

```
$ brew install git
```

** Si no tienes homebrew, puedes instalarlo desde el siguiente [link](#);

- **MacPorts**

```
$ sudo port install git
```

** Si no tienes MacPorts, puedes instalarlo desde el siguiente [link](#).

Windows

- Instalador

- <https://git-scm.com/download/win>

Contenido

¿Qué es Git?

Instalación

.gitignore

Comandos - Locales

Comandos - Remotos

GitHub

Pull Requests

Recursos

.gitignore

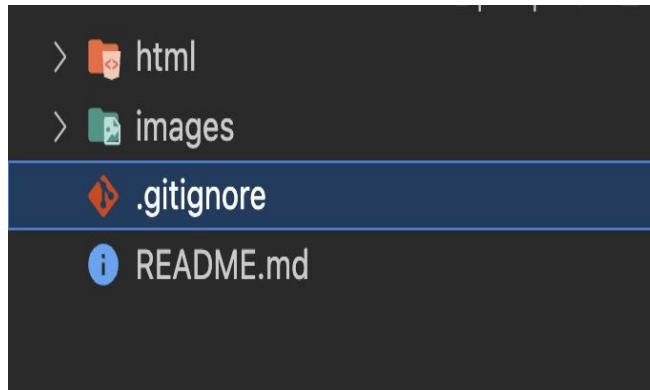
■ ¿Qué es .gitignore?

.gitignore es un archivo de texto en donde le indicamos a git qué archivos y carpetas ignorar.

Este archivo de texto generalmente está en la **raíz del repositorio**.

Para crear un archivo **.gitignore local**, crea un archivo de texto y asígnale el nombre **“.gitignore”**.

Cada nueva línea debe incluir un archivo o carpeta adicional que quieras que Git lo ignore.





Las entradas de este archivo también pueden seguir un patrón coincidente:

- **'*' (ej. *.txt)**

Se utiliza como una coincidencia comodín.

- **'/' (ej. /bin)**

Se usa para ignorar las rutas relativas al archivo .gitignore.

- **'#' (ej. # Esto es un comentario :)**

Es usado para agregar comentarios

```
# Ignora archivos del sistema Mac  
.DS_Store
```

```
# Ignora la carpeta node_modules  
node_modules
```

```
# Ignora todos los archivos de texto  
.txt
```

```
# Ignora los archivos relacionados a API keys  
.env
```

```
# Ignora archivos de configuración SASS  
.sass-cache
```

Contenido

¿Qué es Git?

Instalación

.gitignore

Comandos - Locales

Comandos - Remotos

GitHub

Pull Requests

Recursos

■ Comandos – Locales

Primeros comandos.

- **git --version**

Muestra la versión instalada de git.

- **git help**

Muestra la información/descripción de los comandos.

- **git help commit**

- **git config**

Sirve para agregar configuraciones a la cuenta de git.

- **git config --global user.name "user name"**

Configura el nombre de usuario.

- **git config --global user.email "user.name@email.com"**

Configura el correo electrónico

- **git config --global init.defaultBranch main**

Configuración para que el nombre de la rama principal sea "main" en lugar de "master".

■ Comandos fundamentales

- **git init**

Crea un repositorio de git vacío o reinicializa uno ya existente.

- **git status**

Muestra el estatus del ambiente de trabajo. (working tree)

- **git add**

Agrega uno o más archivos al área de preparación.(staging area)

- `git add nombre_archivo.ext`
- `git add *.ext`
- `git add carpeta/*.ext`
- `git add carpeta/`
- `git add carpeta/.gitkeep`

- **git reset nombre_archivo.ext**

Remueve los archivos indicados del área de preparación al área de trabajo.

Comandos fundamentales

- **git commit -m “initial commit”**

Registra los cambios del repositorio; crea una captura del proyecto en un momento dado.

- **git commit -am “new message”**

Tiene la funcionalidad de un “git add.” y un “git commit” combinadas. (solo para archivos con seguimiento)

- **git commit --amend -m “message correction”**

Corrige el mensaje del último commit

- **git commit --amend -C HEAD**

Agrega los cambios en el área de preparación al último commit en la línea del tiempo.

- **git log**

Muestra el historial de commits que se han realizado en el branch.

Comandos fundamentales

- **git branch <nombre_branch>**

Te crea una nueva versión de la rama en la que estás situada, por lo general main.

- **git branch -m <nuevo_nombre>**

Con este comando puedes renombrar la rama(branch) en el que estás situado.

- **git branch -d <nombre_branch>**

Elimina el branch que le indicas, para esto no debes de estar situado en el branch que quieras eliminar

OTRAS FORMAS DE CREAR UN BRANCH:

- **git checkout -b <nombre_branch>**
- **git switch -c <nombre_branch>**

Nota: con ambos comandos se genera un nuevo branch y te mueves a este en un solo paso.

COMO MOVERSE ENTRE DISTINTOS BRANCHES

- **git checkout <nombre_branch>**
- **git switch <nombre_branch>**

Contenido

¿Qué es Git?

Instalación

.gitignore

Comandos - Locales

Comandos - Remotos

GitHub

Pull Requests

Recursos

■ Comandos – Remotos



Comandos remotos

- **git clone /path/repository/**
Clona un repositorio en un nuevo directorio.
- **git push origin <branch_name>**
Actualiza las referencias remotas junto con los objetos asociados
- **git pull origin <branch_name>**
Obtiene e integra las referencias y los cambios del repositorio o rama remota al local.
- **git fetch origin**
Descarga los cambios y las referencias de un repositorio o rama remota al local.

Contenido

¿Qué es Git?

Instalación

.gitignore

Comandos - Locales

Comandos - Remotos

GitHub

Pull Requests

Recursos



Creación de una cuenta

- Requerimientos:
 - Correo electrónico
- PASOS A SEGUIR:
 1. Ir al sitio oficial de [GitHub](https://github.com) (github.com)
 2. Ingresar tu **correo electrónico**.
 3. Ingresar una **contraseña** de al menos 15 caracteres u 8 incluyendo un número y una minúscula.
 4. Ingresar un **nombre de usuario** único
 5. Indicar si quieres recibir anuncios y actualización por email con una “Y” para sí y una “N” para no.
 6. Verificar tu cuenta.
 7. Dar clic en crear la cuenta.
 8. Ingresar el código que se te envió al correo que diste de alta.

Doble autenticación (2FA)

Dentro de tu cuenta de GitHub sigue los siguientes pasos para habilitar la doble autenticación:

1. Ir a la configuración de tu cuenta:
 - a. Dirígete a la imagen de tu usuario para desplegar un menú
 - b. Selecciona configuración (Settings)
2. Dentro de las configuraciones selecciona la opción de “Contraseña y autenticación” (Password and authentication)
3. En la sección de “Doble autenticación” (Two-Factor Authentication) dar clic en “Habilitar doble autenticación” (Enable two factor authentication)
4. Selecciona la opción que mejor te convenga:
 - a. Usando una aplicación
 - i. [1Password](#), [Authy](#), [LastPass Authenticator](#), or [Microsoft Authenticator](#).
 - b. Por mensaje de texto (SMS)

Doble autenticación (2FA)

5. Una vez seleccionada una opción vas a tener que seguir la configuración que se te presenta
 - a. Para una aplicación tienes que escanear un QR
 - b. SMS: tienes que ingresar un número de teléfono válido
6. Guardar los códigos que se te muestran en la pantalla.
7. Si ya llegaste hasta este paso tienes activada la doble autenticación.



Clave SSH

Primero tienes que crear una clave SSH, sigue los siguientes pasos para conseguirlo.

1. Ve a la terminal de tu SO
2. Pega el siguiente comando en tu terminal y sustituye el correo por el que usas en tu cuenta de GitHub
`> $ ssh-keygen -t ed25519 -C "your_email@example.com"`
3. Te va a pedir ingresar en donde quieras almacenar tu clave, por defecto es en `"/Usuarios/TuUsuario/.ssh/id"`

Clave SSH

4. Te va a pedir un frase de seguridad, puedes dejarla vacía
5. Te va a aparecer un mensaje indicando donde se almaceno tu clave, algo así:
6. Listo ya tienes un clave de SSH generada.

```
Your identification has been saved in /Users/georgina.fraire/.ssh/id_ed25519
Your public key has been saved in /Users/georgina.fraire/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:rt3KW:A7WjZLesbKA/WJu5MDTDqqccCDYOS9UozytYU your_email@example.com
The key's randomart image is:
+--[ED25519 256]--+
| . |
| o + . |
| o+ +E . |
| *....o.. |
| o+...+ .So . |
| ..o o +..oo |
| . . . . o.+B |
| o. =*B=+ |
| .. . @%+. |
+----[SHA25519]-----+
```

Clave SSH

En tu terminal realiza lo siguiente:

1. Copia el contenido de tu clave pública al portapapeles con el siguiente comando, recuerda reemplazar el **id_ssh** por el que tu generaste.

```
> pbcopy < ~/.ssh/id_ssh.pub
```

2. También puedes abrir el archivo con algún editor de texto y copiar el contenido; por ejemplo, nano con el siguiente comando:

```
> nano < ~/.ssh/id_ssh.pub
```



Clave SSH

Ahora dentro de tu cuenta de GitHub haz lo siguiente:

1. Ir a la configuración de tu cuenta:
 - a. Dirígete a la imagen de tu usuario para desplegar un menú
 - b. Selecciona configuración (Settings)
2. Dentro de las configuraciones selecciona la opción de “Claves SSH y GPU” (SSH Y GPU Keys)
3. Dentro de la sección SSH da clic en el botón “Nueva Clave SSH” (New SSH Key)
4. Agrega un título
5. Pega el contenido de la clave SSH dentro de la sección “Clave” (key)
6. Da clic en “Aregar clave ssh” (Add ssh key)
7. LISTO.

Contenido

¿Qué es Git?

Instalación

.gitignore

Comandos - Locales

Comandos - Remotos

GitHub

Pull Requests

Recursos

Pull Requests

■ Repositorio Entregables

https://github.com/WIZEACADEMY-BAZ/BAZJAVA_BLOQUE2

<https://forms.gle/i8zpZ3pCFiYJaGMy7>



Pull Requests

Comenzamos en la interfaz de Github, en la pestaña "Code".

Aquí podemos visualizar el contenido del repositorio para la rama "main", así como el contenido del archivo "README.md".

The screenshot shows the GitHub repository interface for 'RafaelWize / Basics'. The 'Code' tab is selected. The main branch is 'main'. There is 1 branch and 2 tags. The repository contains several files and commits:

File	Commit Message	Time Ago
.gitignore	Added Project Structure (No content on files)	5 months ago
html	Added Project Structure (No content on files)	5 months ago
js	Added Project Structure (No content on files)	5 months ago
ruby	Added Project Structure (No content on files)	5 months ago
README.md	new push	4 months ago
languages.md	Fix: #2, agregué los lenguajes que me pidieron	5 days ago
teammates.md	closes #3 closes #4	5 days ago

At the bottom, the 'README.md' file is shown with the content 'push'.

Pull Requests

En la misma sección, sobre el nombre de la rama (main), tenemos la opción de crear una rama a partir de otra; en esta ocasión vamos a crear una rama llamada "from-main" a partir de la rama "main".

Esta rama contendrá el mismo historial de cambios y versiones que la rama "main".

The screenshot shows a GitHub repository interface. At the top, there are buttons for 'Go to file', 'Add file', and 'Code'. Below that, a dropdown menu is open, showing the current branch 'main' and a list of branches and tags. One branch, 'from-main', is highlighted with a blue border. The main content area displays a list of commits:

Commit Message	Date
0073c67 5 days ago 32 commits	
Added Project Structure (No content on files)	5 months ago
Added Project Structure (No content on files)	5 months ago
Added Project Structure (No content on files)	5 months ago
Added Project Structure (No content on files)	5 months ago
README.md new push	4 months ago
languages.md Fix: #2, agregué los lenguajes que me pidieron	5 days ago
teammates.md closes #3 closes #4	5 days ago

At the bottom, there is a section for the 'README.md' file with a 'push' button and an edit icon.

Pull Requests

Podemos verificar que estamos posicionados sobre la rama:

- "**from-main**" y la leyenda "**This branch is up to date with main**", puesto que contienen el mismo historial de cambios.

This branch is up to date with main.

RafaelWize closes #3 closes #4 0073c67 5 days ago 32 commits

File	Commit Message	Time Ago
html	Added Project Structure (No content on files)	5 months ago
js	Added Project Structure (No content on files)	5 months ago
ruby	Added Project Structure (No content on files)	5 months ago
.gitignore	Added Project Structure (No content on files)	5 months ago
README.md	new push	4 months ago
languages.md	Fix: #2, agregué los lenguajes que me pidieron	5 days ago
teammates.md	closes #3 closes #4	5 days ago

README.md

push

Pull Requests

Podemos desde la interfaz simular un nuevo commit en nuestra rama "**from-main**", modificando el contenido del archivo **README.md**; basta con hacer click en el nombre del archivo y posteriormente en el ícono del lápiz que se encuentra en el menú del lado derecho.

The screenshot shows a GitHub repository interface. At the top, there's a navigation bar with a dropdown for 'from-main', the link 'Basics / README.md', and buttons for 'Go to file' and '...'. Below the header, a message from 'RafaelWize' indicates a 'new push'. To the right, it says 'Latest commit c33c3e6 on Jun 16' and has a 'History' button. A note below the message says '1 contributor'. The main content area displays the file's content: '1 lines (1 sloc) | 4 Bytes' followed by the word 'push'. At the bottom of the content area are several icons: a diff icon, a copy icon, 'Raw', 'Blame', an edit icon, a dropdown menu, a clipboard icon, and a trash bin icon. At the very bottom of the page, there's a footer with links: GitHub logo, '© 2022 GitHub, Inc.', and links for 'Terms', 'Privacy', 'Security', 'Status', 'Docs', 'Contact GitHub', 'Pricing', 'API', 'Training', 'Blog', and 'About'.

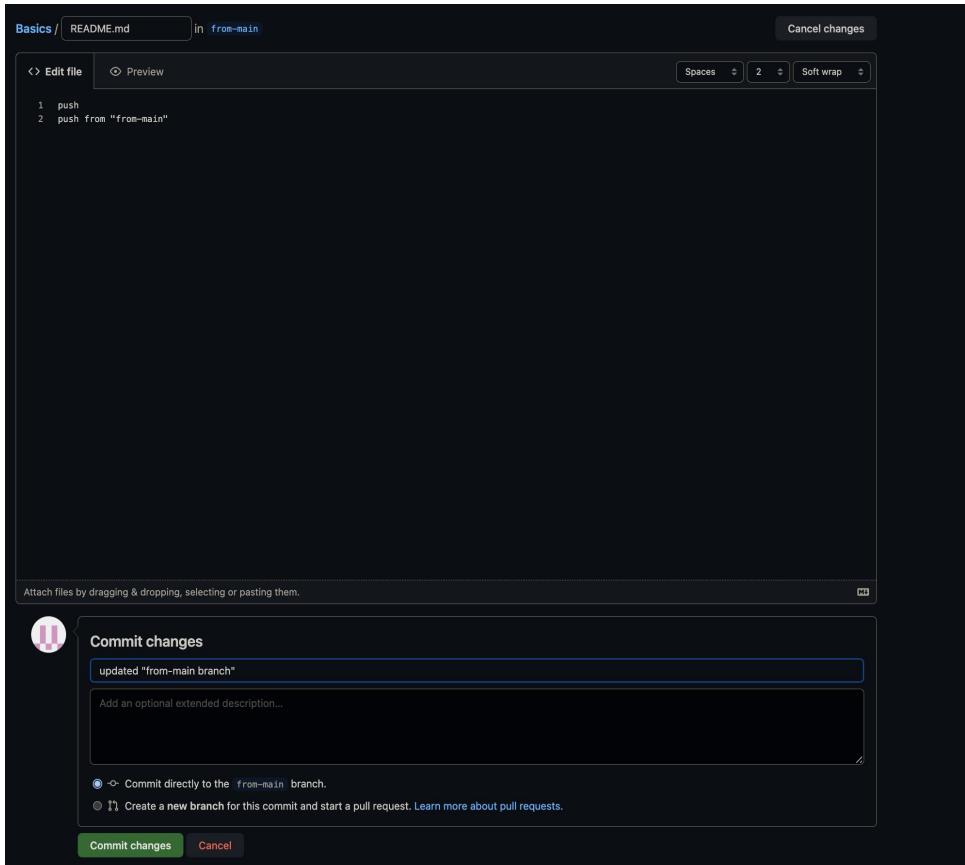
Pull Requests

Una vez dentro de la característica que nos ofrece Github para modificar el archivo, modificamos el contenido (en esta ocasión agregamos una línea nueva para simular un cambio).

Después en la sección inferior colocamos el nombre con el que va a ser identificado nuestro cambio (commit) y tenemos dos opciones:

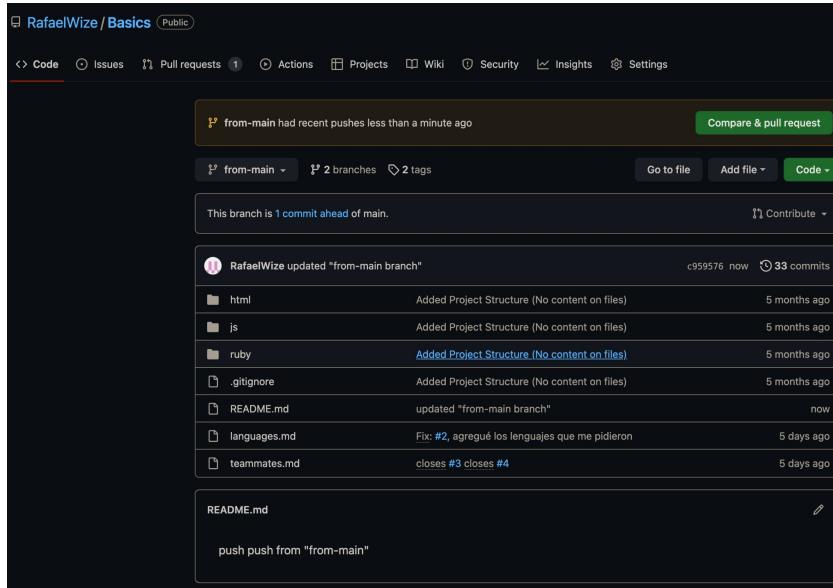
- 1) hacer un commit a la rama actual (from-main; es la opción que vamos a seleccionar)
- 2) crear una nueva rama a partir de este commit y hacer directamente la Pull Request

Una vez que elegimos nuestra opción presionamos el botón de color verde "Commit Changes".



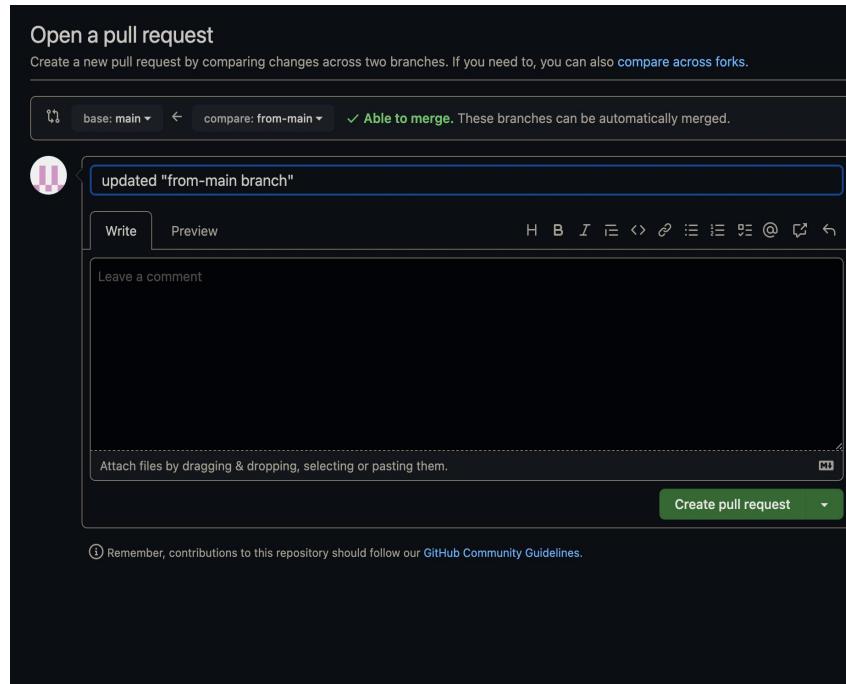
Pull Requests

Una vez que creamos nuestro nuevo commit en la rama, al regresar a la pestaña "Code" podemos visualizar que la leyenda cambió a "**This branch is 1 commit ahead of main**" y adicional a eso apareció otra leyenda con el botón "**Compare & pull request**"; esto quiere decir que nuestra rama se encuentra con cambios que podemos integrar a la rama "**main**" puesto que tenían el mismo historial y se puede integrar sin problema.



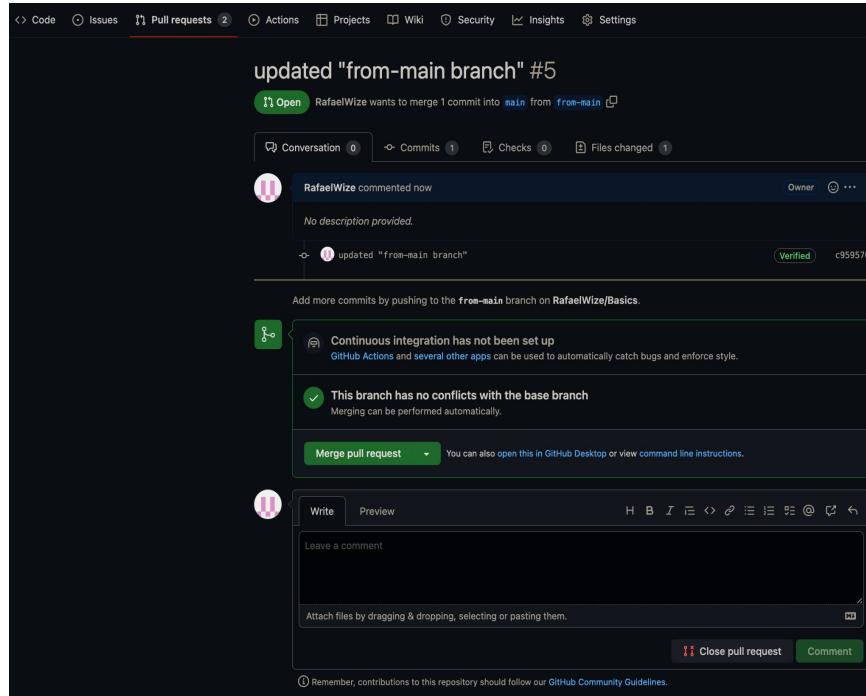
Pull Requests

Una vez que seleccionamos el botón de "**Compare y pull request**", podemos seleccionar la rama desde donde se van a integrar los cambios (from-main) y la rama donde se van a integrar (en este caso la rama "main"); además podemos escribir un nombre para identificar este commit.



Pull Requests

Una vez terminado el proceso anterior, se visualizará una pantalla similar a la de la imagen inferior, con cuatro pestañas (Conversation, Commits, Checks y Files changed). De igual manera, el título del Pull Request fue el que indicamos en el paso anterior.



Pull Requests

En la pestaña "Pull Requests" puedes verificar la cantidad de commits que se pretenden integrar, esto lo encuentras en la pestaña "Commits" de la interfaz. Esta contendrá el nombre de los commit, el autor de los mismos, la fecha de creación y el hash del commit (en este caso solo tenemos 1 commit para integrar).

The screenshot shows a GitHub repository named 'RafaelWize / Basics' (Public). The 'Pull requests' tab is selected, showing one open pull request titled 'push from main changes #6'. The pull request summary indicates that RafaelWize wants to merge 1 commit from the 'from-main' branch into the 'main' branch. The commit details show a single commit titled 'push from main changes' made by RafaelWize on October 5, 2022. The commit hash is d769ea3.

RafaelWize / Basics (Public)

Code Issues Pull requests 2 Actions Projects Wiki Security Insights Settings

push from main changes #6

Open RafaelWize wants to merge 1 commit into main from from-main

Conversation 1 Commits 1 Checks 0 Files changed 1 +1 -0

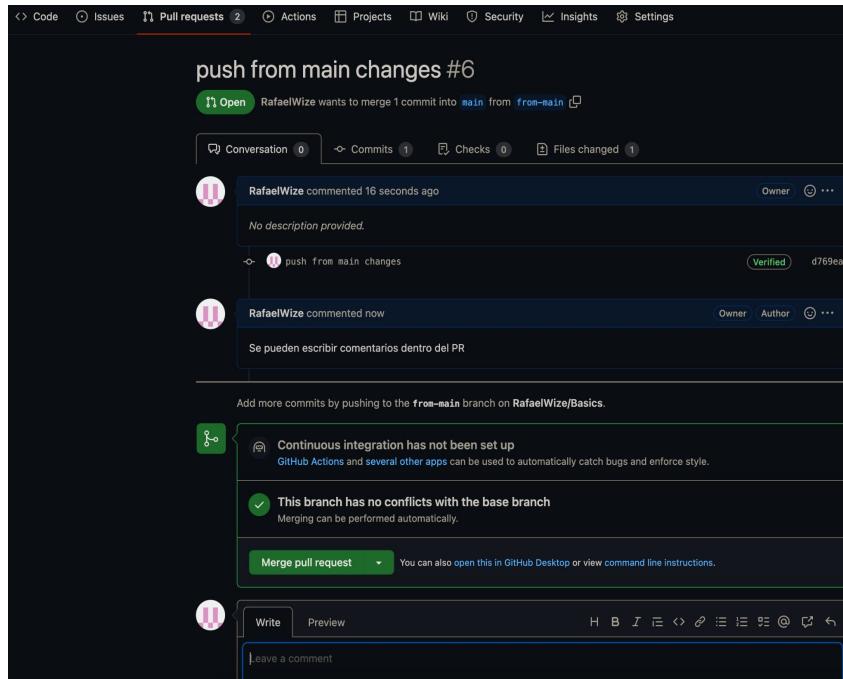
Commits on Oct 5, 2022

push from main changes
RafaelWize committed 1 minute ago

Verified d769ea3

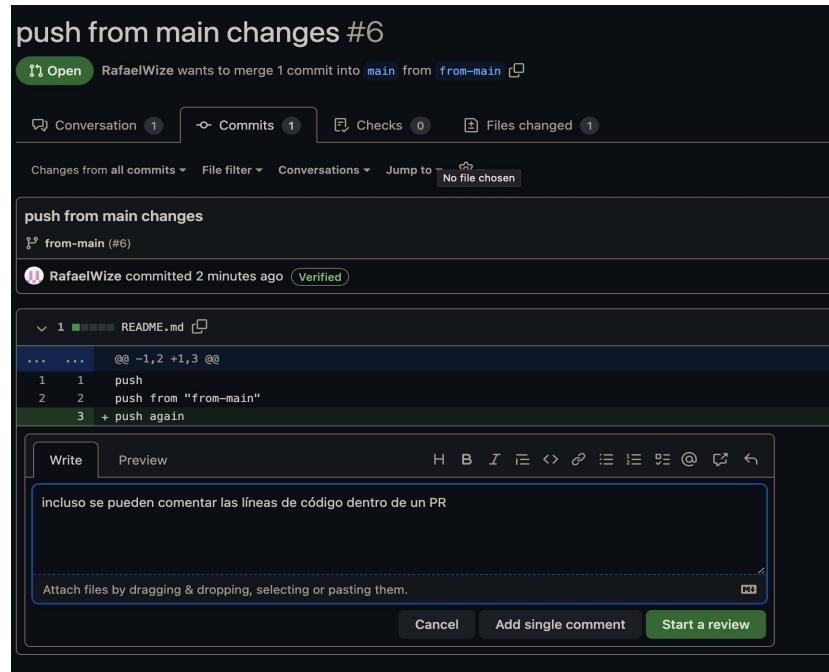
Pull Requests

En la sección "Conversation", podemos precisamente realizar comentarios -back and forth- para discutir acerca de la integración de los cambios; ya sea cambios requeridos para poder integrarlos, solicitar aprobaciones, entre otras cosas.



Pull Requests

Puedes realizar comentarios dentro de las Pull request, línea por línea, haciendo reseñas o requerimientos para los cambios que se van a integrar.



Pull Requests

También podemos encontrar los archivos que fueron modificados (las diferencias entre el original y el cambio), en la pestaña "Files changed".

The screenshot shows a GitHub repository named 'RafaelWize / Basics' (Public). The 'Pull requests' tab is selected, showing one open pull request titled 'push from main changes #6'. The 'Files changed' tab is also selected, showing a single file, 'README.md'. The diff view highlights the following changes:

```
@@ -1,2 +1,3 @@
 1   1 push
 2   2 push from "from-main"
 3 + push again
```

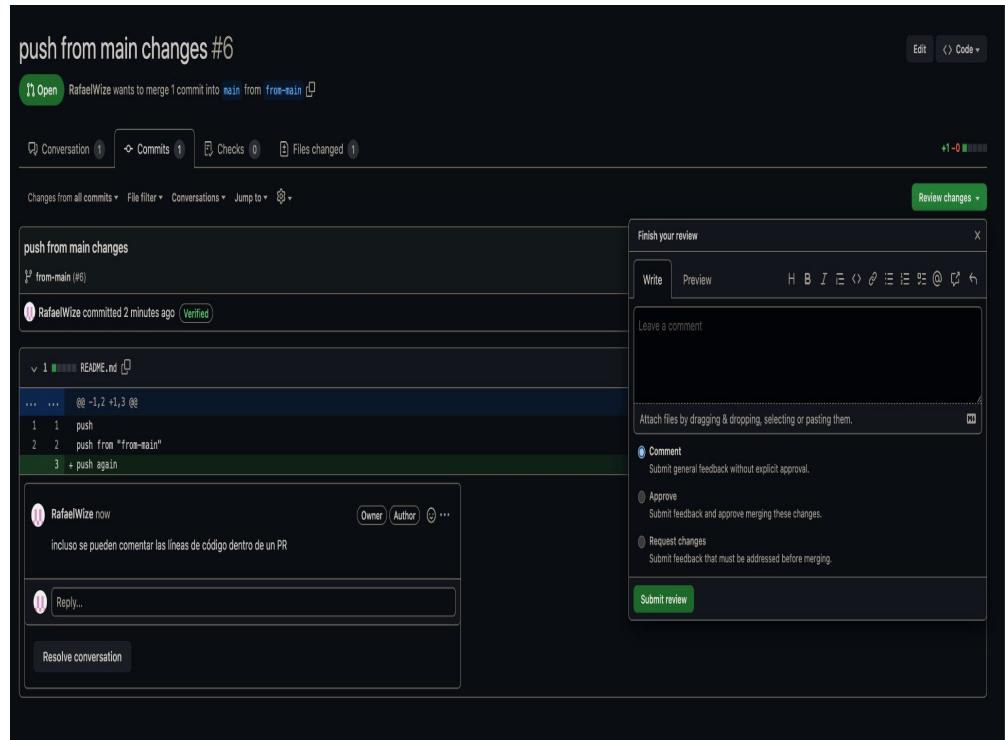
Pull Requests

En la interfaz del lado derecho, podemos apreciar el botón "Review changes"; al hacer clic este desplegará un menú con opciones diversas:

1) Comment: Puedes añadir un mensaje general sin explicitamente aprobar o rechazar los cambios.

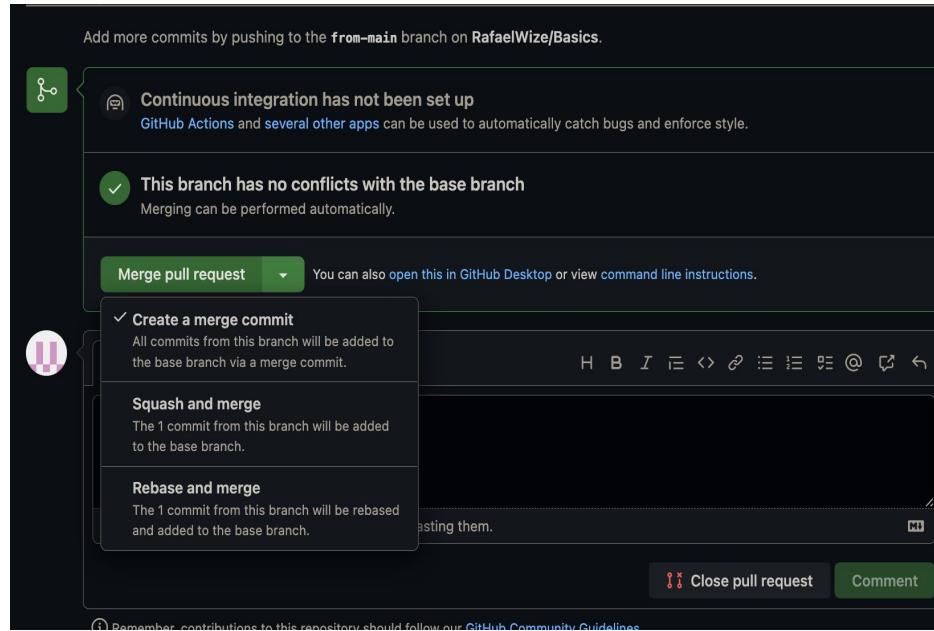
2) Approve: Puedes incluir un mensaje -o no- y posteriormente aprobar los cambios que se van a integrar (dar visto bueno a los cambios; seleccionamos esta opción).

3) Request changes: Puedes añadir un mensaje para posteriormente requerir cambios antes de realizar la aprobación para la integración de los cambios.



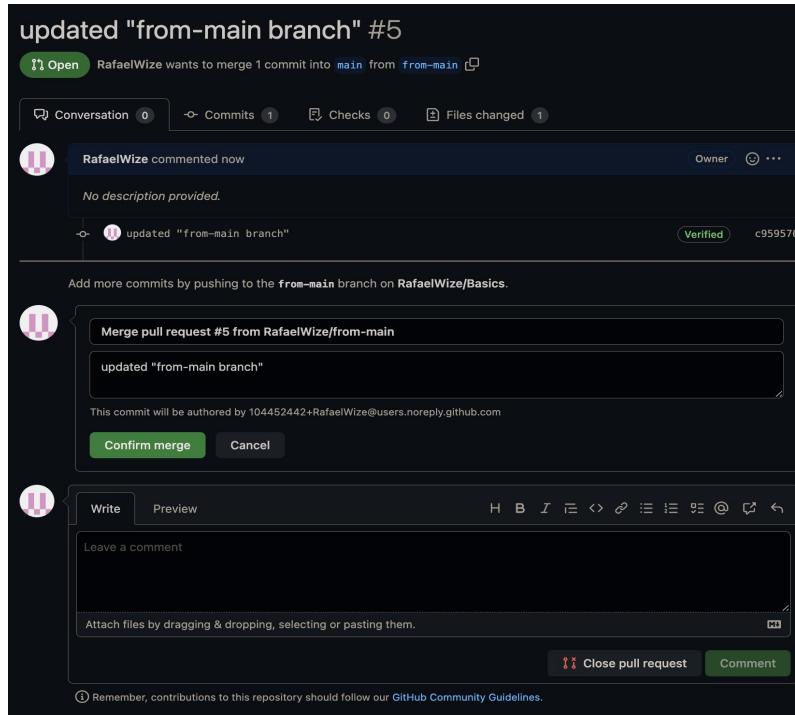
Pull Requests

Una vez que los cambios fueron aprobados, podemos seleccionar la manera en que se van a integrar a nuestra rama destino; existen tres estrategias, pero la más común es "Create a merge commit".



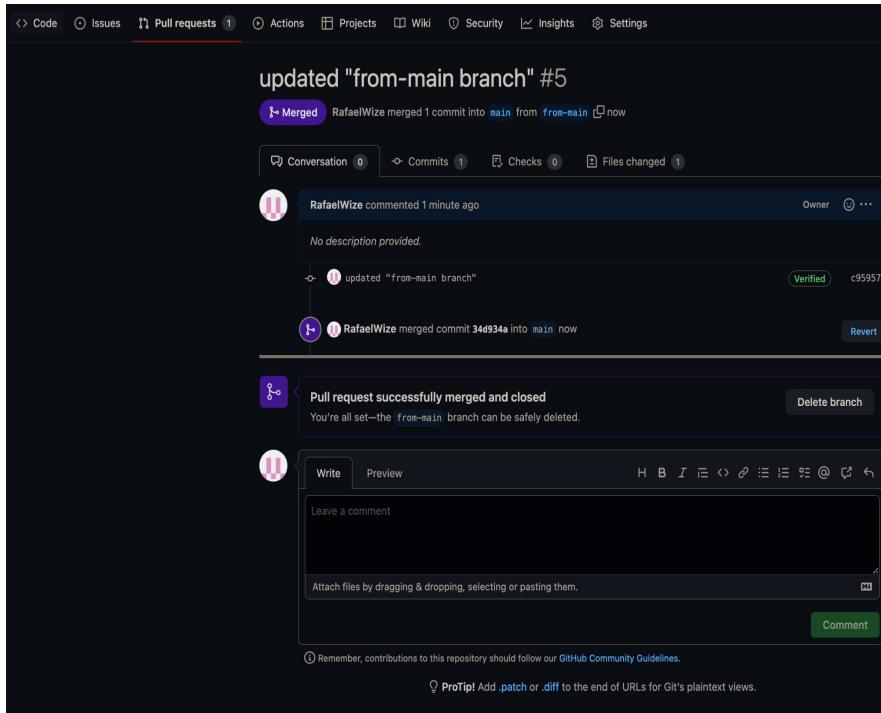
Pull Requests

Una vez seleccionada la estrategia para integrar nuestro código, se procede a crear un commit para identificar la integración de nuestros cambios en la rama destino.



Pull Requests

Al momento de integrar nuestros cambios la interfaz cambia y podemos apreciar como de color verde, cambia a color púrpura y con el status "merged". Esto indica que nuestros cambios ya fueron integrados a la rama destino (main).



Pull Requests

Podemos observar que posicionados en la rama "main" se encuentran integrados los cambios que teníamos desde la rama "from-main"; incluso podemos apreciar en la parte inferior que el contenido del archivo README.md está actualizado.

RafaelWize / Basics Public

<> Code ⚡ Issues Pull requests 1 Actions Projects Wiki Security Insights Settings

main ▾ 2 branches 2 tags Go to file Add file ▾ Code ▾

RafaelWize Merge pull request #5 from RafaelWize/from-main 34d934a 30 seconds ago 34 commits

File	Description	Time
html	Added Project Structure (No content on files)	5 months ago
js	Added Project Structure (No content on files)	5 months ago
ruby	Added Project Structure (No content on files)	5 months ago
.gitignore	Added Project Structure (No content on files)	5 months ago
README.md	updated "from-main branch"	2 minutes ago
languages.md	Fix: #2, agregué los lenguajes que me pidieron	5 days ago
teammates.md	closes #3 closes #4	5 days ago

README.md

push push from "from-main"

Contenido

¿Qué es Git?

Instalación

.gitignore

Comandos - Locales

Comandos - Remotos

GitHub

Pull Requests

Recursos

Actividad

- Generen un branch <Nombre_Apellido_Entregable1> (localmente y en el repositorio)
- Generar su directorio <Nombre_Aellido1_Aellido2>
- Generar directorio **Entregable 1 / Curso GIT**
- Dentro del directorio de "**Curso GIT**", agregar las **screenshots** de:
 - Pasos para hacer desde branch hasta push origin
 - Crear branch
 - Hacer cambios de archivos y directorios
 - Agregar los cambios (git add)
 - Hacer los commits
 - Subir los cambios al repositorio remoto (git push)
- Generar un README.md (contenido variado)
- Generar un Pull Request



Recursos



Recursos

- GIT: <https://git-scm.com/>
- GITHUB: <https://docs.github.com/es>
- CLAVE SSH: <https://docs.github.com/es/authentication/connecting-to-github-with-ssh>
- DOBLE AUTENTIFICACIÓN:
<https://docs.github.com/es/authentication/securing-your-account-with-two-factor-authentication-2fa>
- GIT - ATLISSIAN <https://www.atlassian.com/git>
- GITIGNORE:
<https://www.freecodecamp.org/espanol/news/gitignore-explicado-que-es-y-como-agregar-a-tu-repositorio>

■ Repositorio Contenido Bloque 2

<https://github.com/wizelineacademy/BAZJAVA2-2022>





THANK YOU