

# **Cristhian Saez**

Git Evelyn

# Índice

Git Hub 3



# **GitHub Evelyn**

# 1. GitHub

Parte 1: Configuración Inicial

## 1. Configurar Git en tu equipo

```
git config --global user.name "Tu Nombre"
```

git config --global user.email <a href="mailto:tuemail@example.com">tuemail@example.com</a>

```
C:\Users\Usuario>git config --global user.name "cristhian"
C:\Users\Usuario>git config --global user.email "csa0012@alu.medac.es
C:\Users\Usuario>git config --global user.email "csa0012@alu.medac.es"
```

#### 2. Crear un nuevo repositorio en GitHub

o Accede a GitHub.

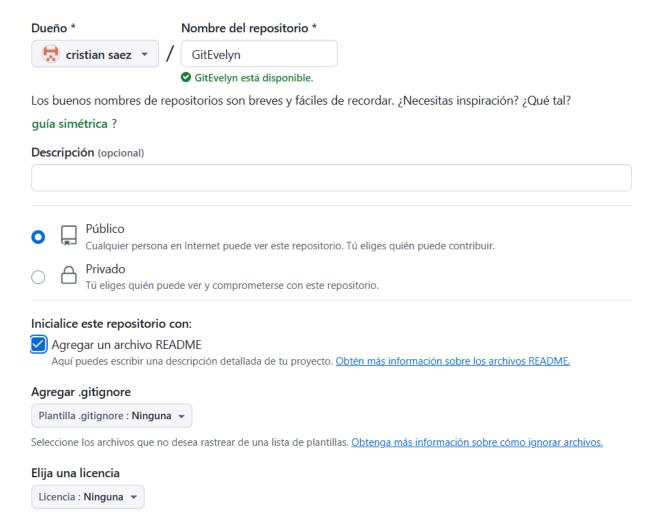
o Haz clic en "New Repository".

o Asigna un nombre al repositorio y selecciona la opción "Initialize this

repository with a README".











#### 3. Clonar el repositorio en tu equipo

```
git clone URL_DEL_REPOSITORIO cd NOMBRE_DEL_REPOSITORIO
```

```
C:\Users\Usuario\Desktop\DAW\Despliegue\Git>git clone https://github.com/cristhiansaez/GitEvelyn
Cloning into 'GitEvelyn'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.

C:\Users\Usuario\Desktop\DAW\Despliegue\Git>cd Git Evelyn
El sistema no puede encontrar la ruta especificada.

C:\Users\Usuario\Desktop\DAW\Despliegue\Git>cd GitEvelyn
C:\Users\Usuario\Desktop\DAW\Despliegue\Git>cd GitEvelyn
```

#### Parte 2: Trabajo Individual con Git

## 1. Crear un archivo de prueba y hacer un commit

echo "# Proyecto Web" > index.html

git add index.html

git commit -m "Añadir archivo index.html"



```
C:\Users\Usuario\Desktop\DAW\Despliegue\Git\GitEvelyn>echo "#ProyectoWeb">index.hmtl

C:\Users\Usuario\Desktop\DAW\Despliegue\Git\GitEvelyn>git add index.html
fatal: pathspec 'index.html' did not match any files

C:\Users\Usuario\Desktop\DAW\Despliegue\Git\GitEvelyn>echo "#ProyectoWeb">index.html

C:\Users\Usuario\Desktop\DAW\Despliegue\Git\GitEvelyn>git add index.html

C:\Users\Usuario\Desktop\DAW\Despliegue\Git\GitEvelyn>git commit -m "Añadir archivo index.html"
[main 8ba067d] Añadir archivo index.html

1 file changed, 1 insertion(+)
create mode 100644 index.html
```

# 2. Subir cambios al repositorio remoto

git push origin main

```
C:\Users\Usuario\Desktop\DAW\Despliegue\Git\GitEvelyn>git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 304 bytes | 152.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/cristhiansaez/GitEvelyn
fc25f37..8ba067d main -> main
```



3. Actualizar el repositorio local con los cambios remotos

git pull origin main

```
C:\Users\Usuario\Desktop\DAW\Despliegue\Git\GitEvelyn>git pull origin main
From https://github.com/cristhiansaez/GitEvelyn
* branch main -> FETCH_HEAD
Already up to date.
```

### Parte 3: Trabajo Colaborativo

1. Crear una nueva rama para una funcionalidad

```
C:\Users\Usuario\Desktop\DAW\Despliegue\Git\GitEvelyn>git branch nueva-funcionalidad
C:\Users\Usuario\Desktop\DAW\Despliegue\Git\GitEvelyn>git checkout nueva-funcionalidad
Switched to branch 'nueva-funcionalidad'
```

git branch nueva-funcionalidad

git checkout nueva-funcionalidad

(O también puedes usar: git checkout -b nueva-funcionalidad)

2. Realizar cambios en la nueva rama y hacer commit

echo "<h1>Hola mundo</h1>" > index.html

git add index.html

git commit -m "Añadir título en index.html"





### 3. Subir la rama al repositorio remoto

git push origin nueva-funcionalidad

```
C:\Users\Usuario\Desktop\DAW\Despliegue\Git\GitEvelyn>git push origin nueva-funcionalidad
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 315 bytes | 157.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'nueva-funcionalidad' on GitHub by visiting:
remote: https://github.com/cristhiansaez/GitEvelyn/pull/new/nueva-funcionalidad
remote:
To https://github.com/cristhiansaez/GitEvelyn
* [new branch] nueva-funcionalidad -> nueva-funcionalidad
```



4. Realizar un Pull Request (PR) en GitHub

Un Pull Request (PR) es una solicitud para fusionar los cambios de una rama en otra, generalmente de una rama de desarrollo a main. Es una forma de revisar y discutir cambios antes de fusionarlos.

Pasos para crear un PR:

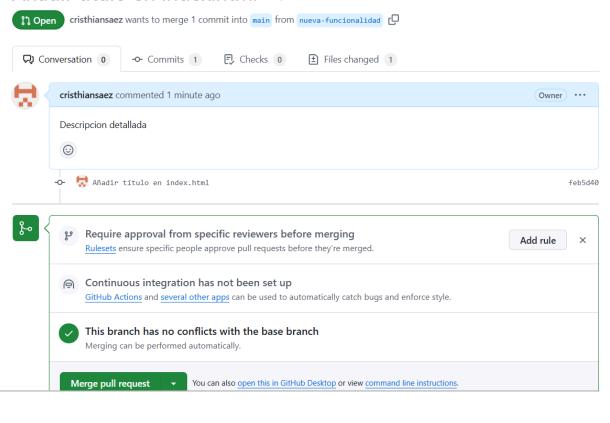
- o Accede al repositorio en GitHub.
- o Ir a la pestaña "Pull Requests" y hacer clic en "New Pull Request".
- o Seleccionar la rama que contiene los cambios y compararla con main.
- o Revisar los cambios antes de enviarlos.
- o Escribir una descripción detallada del PR explicando los cambios realizados.
- o Hacer clic en "Create Pull Request".
- o Esperar revisiones y feedback de los compañeros o del profesor.

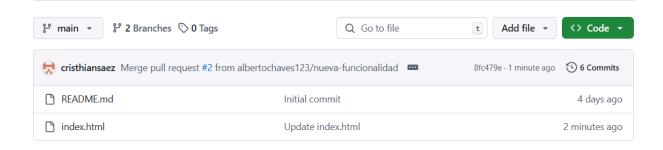
```
::\Users\Usuario\Desktop\DAW\Despliegue\Git\GitEvelyn>git clone https://github.com/albertochaves123/git-evelyn.git
Cloning into 'git-evelyn'...
remote: Enumerating objects: 10, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 10 (delta 0), reused 6 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (10/10), done.
```





# Añadir título en index.html #1









```
C:\Users\Usuario\Desktop\git-evelyn>git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
C:\Users\Usuario\Desktop\git-evelyn>git pull origin main
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (4/4), 1.80 KiB | 167.00 KiB/s, done.
From https://github.com/albertochaves123/git-evelyn
 * branch
                        main
                                     -> FETCH_HEAD
   0413ac3..8fc479e main
                                     -> origin/main
Updating 0413ac3..8fc479e
Fast-forward
 index.html | Bin 44 -> 25 bytes
 1 file changed, 0 insertions(+), 0 deletions(-)
```

- 5. Revisar y fusionar cambios
- o Otro compañero revisa el PR y lo aprueba.
- o Si todo está correcto, se fusiona con la rama main.
- o Si hay sugerencias o cambios requeridos, deben realizarse antes de la



aprobación.

o Una vez aprobado, hacer clic en "Merge Pull Request" y eliminar la rama si ya no es necesaria.

6. Actualizar la rama local con los cambios aprobados

git checkout main

git pull origin main

Parte 4: Resolución de Conflictos

- 1. Generar un conflicto (de forma intencionada)
- o Dos alumnos editan la misma línea de index.html en diferentes ramas y

hacen commits.

- o Ambos intentan fusionar sus ramas con main.
- 2. Resolver el conflicto
- o Git avisará del conflicto.
- o Edita manualmente el archivo afectado, manteniendo los cambios

correctos.

o Una vez resuelto, hacer:

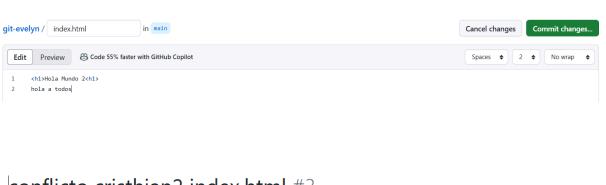
git add index.html

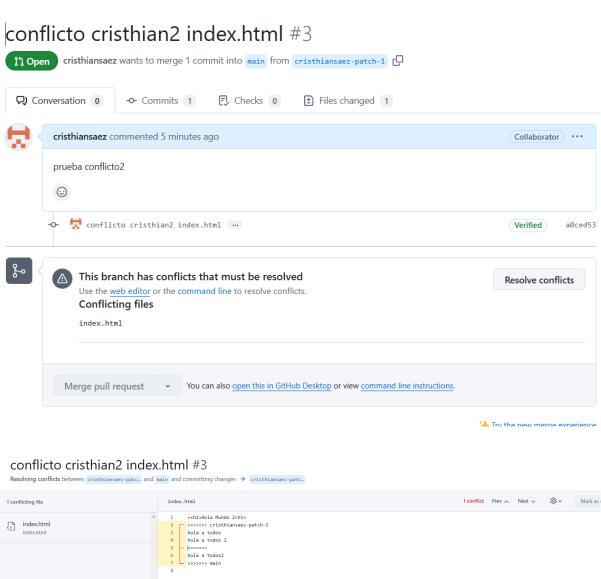
git commit -m "Resolver conflicto en index.html"

git push origin main











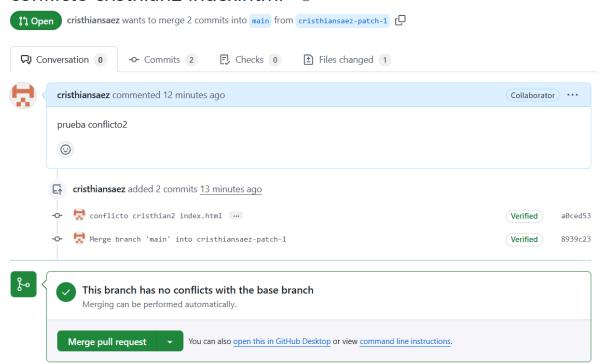








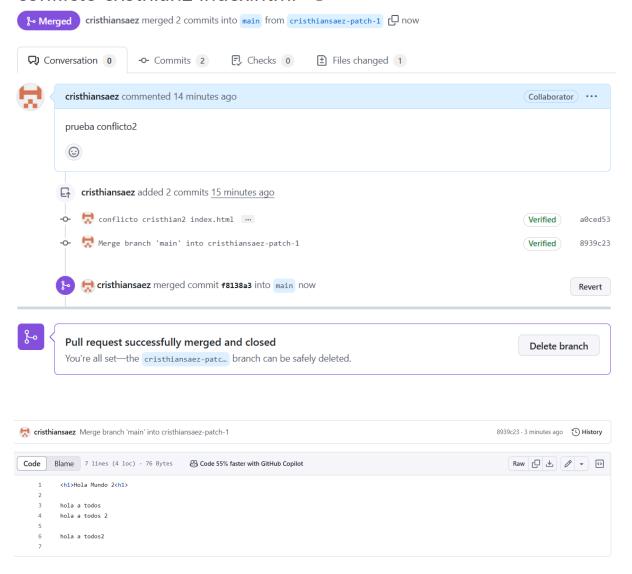
#### conflicto cristnian2 index.ntml #3







#### conflicto cristhian2 index.html #3





# Parte 5: Entrega Final

- 1. Cada equipo debe tener un repositorio con al menos:
- o Un README.md documentado.
- o Un historial de commits bien estructurado.
- o Uso de ramas y Pull Requests.
- o Al menos un conflicto resuelto correctamente.
- 2. Enviar el enlace del repositorio a la plataforma de entrega

