



GitHub



git



git



Acceder a este
enlace para
descargar la versión
mas reciente

A screenshot of the Git website homepage. The page has a light beige background with a subtle grid pattern. At the top, the Git logo is on the left, and a search bar is on the right. Below the logo, there's a paragraph describing Git as a free and open source distributed version control system. To the right of this text is a diagram showing a branching model with multiple branches and commits. Below the main text, there are four circular icons with text: 'About', 'Documentation', 'Downloads', and 'Community'. To the right of these icons is a large monitor displaying the 'Latest source Release 2.33.0' and a 'Download for Windows' button. Below the monitor are links for 'Windows GUIs', 'Tarballs', 'Mac Build', and 'Source Code'. At the bottom, there's a section titled 'Companies & Projects Using Git' with logos for Google, Facebook, Microsoft, Twitter, LinkedIn, Netflix, PostgreSQL, Android, Ubuntu, Qt, GNOME, Eclipse, and others.

Para comenzar deberemos acceder a la pagina oficial de git y descargarnos la herramienta de comandos de git. Git es un sistema de control de versiones distribuido de código abierto y gratuito . En las siguientes diapositivas se muestran los pasos para instalar git (Dejar todo como se ve en las imágenes)

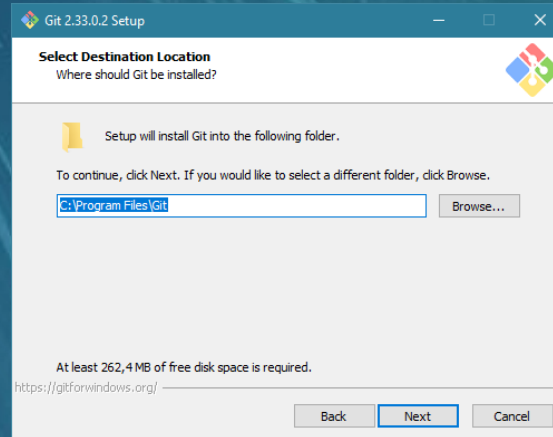


Pasos para instalar

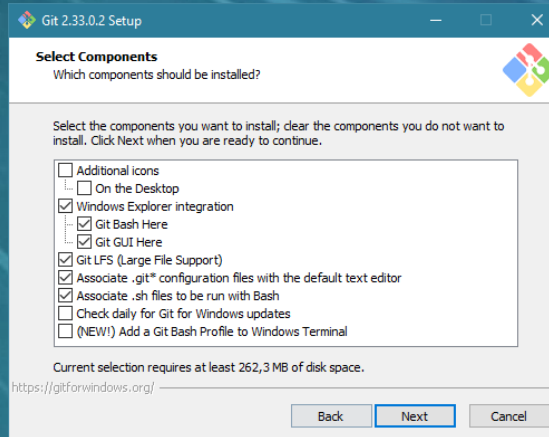
1



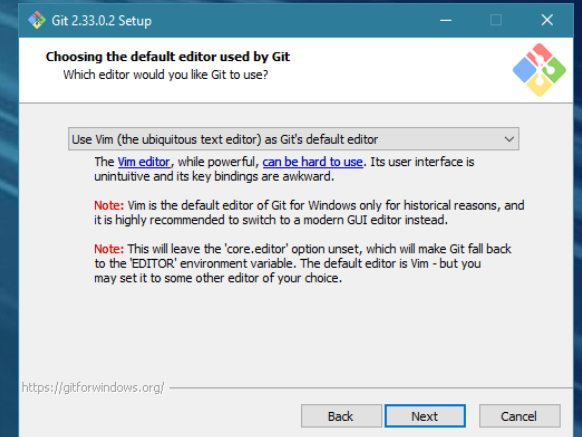
2



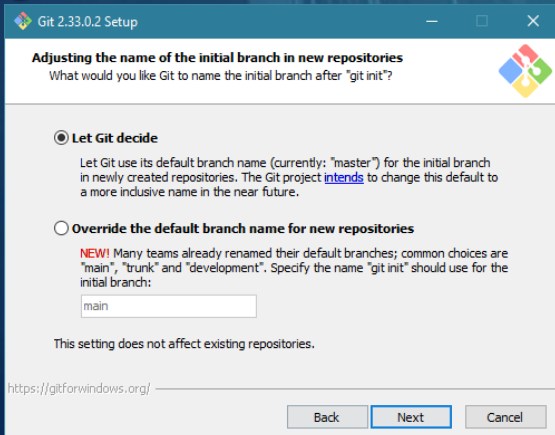
3



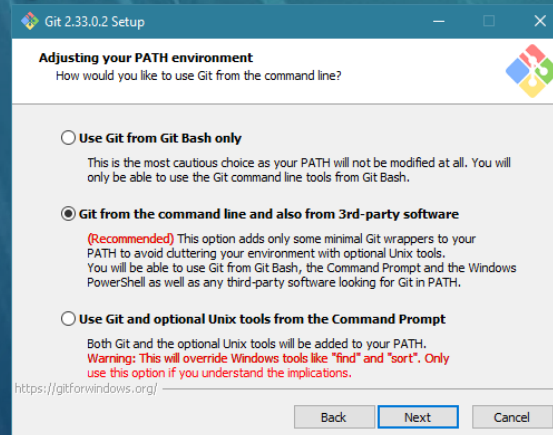
4



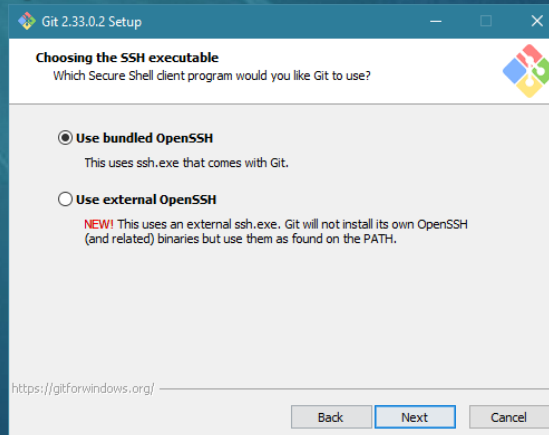
5



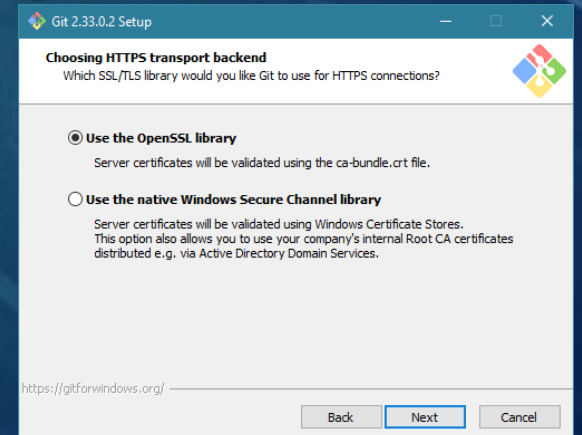
6



7



8



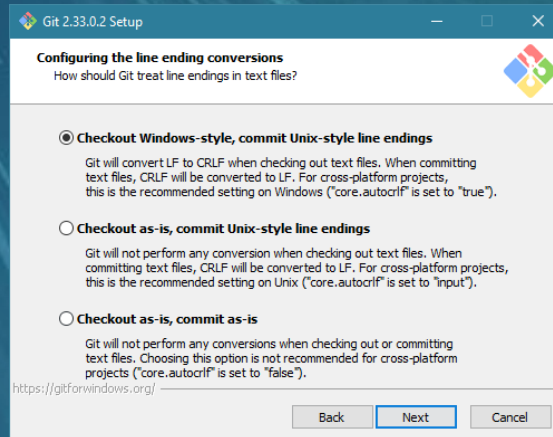


Pasos para instalar

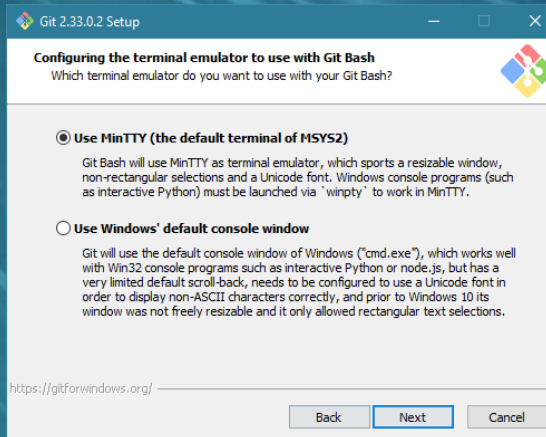
9



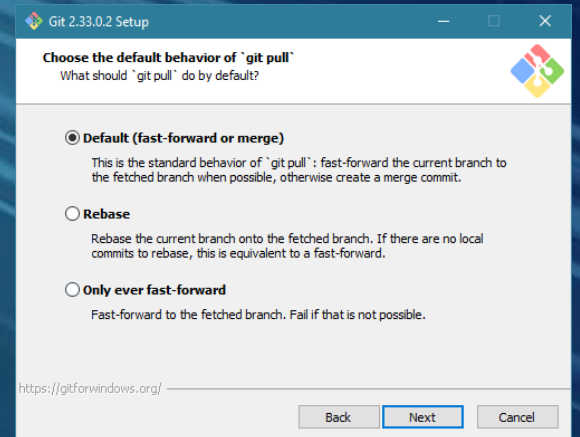
10



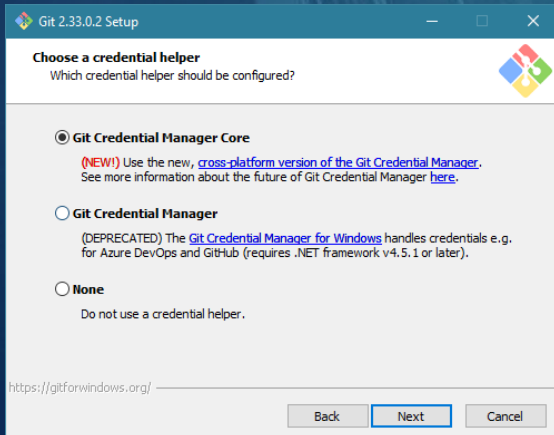
11



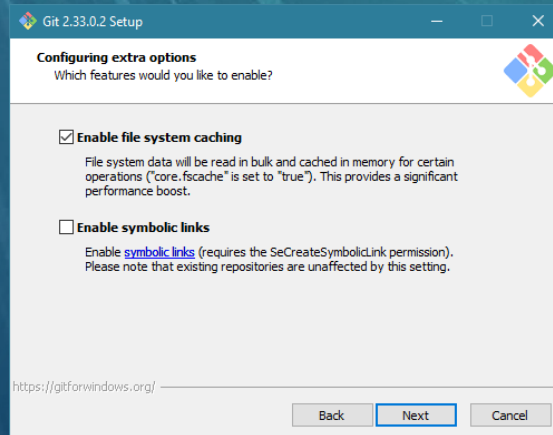
12



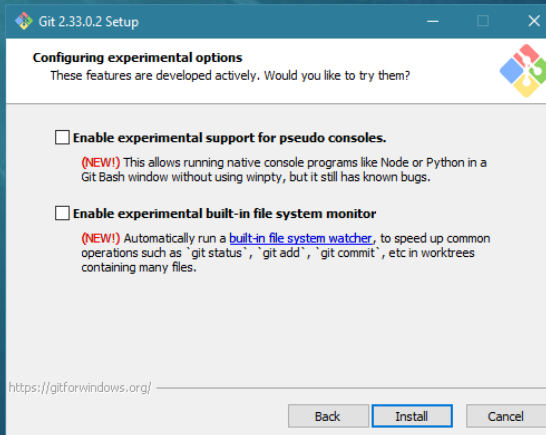
13



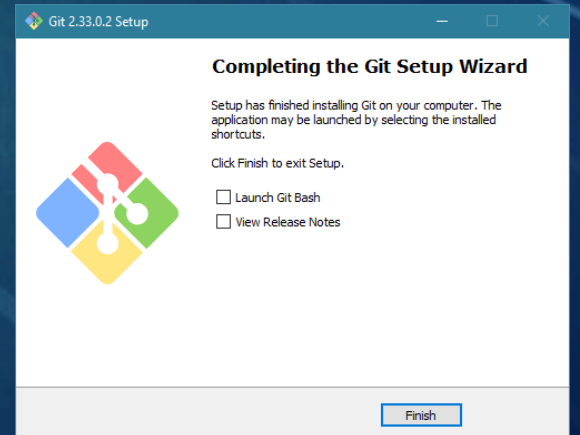
14



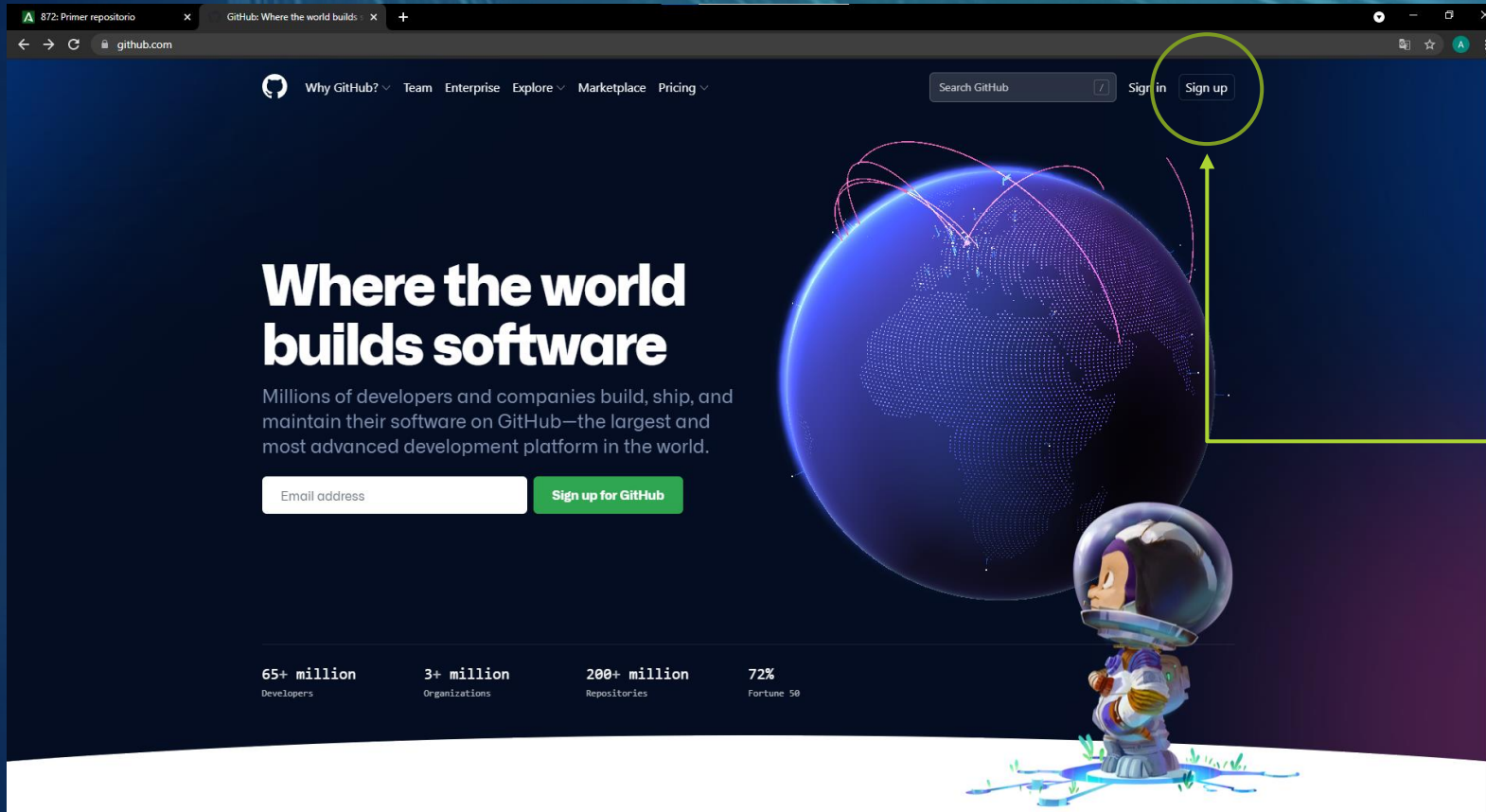
15



16







Esta es la pagina principal de GitHub.

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

En primer lugar accedemos al botón donde dice “Sign up” para crear nuestra cuenta.



Welcome to GitHub!
Let's begin the adventure

Enter your email

✓ arancol1101@g.educaand.es

Create a password

✓

Enter a username

✓ RamosColonAdrian2

Would you like to receive product updates and announcements via email?

Type "y" for yes or "n" for no

- n

Continue

Una vez empezamos a crear la cuenta, GitHub nos pedirá un correo electrónico, una contraseña y un nombre de usuario.

Abajo es recomendable escribir "n" para que no nos lleguen e-mails con publicidad



Almost done, @RamosColonAdrian !

To complete your GitHub sign up, we just need to verify your email address: aramcol1101@g.educaand.es.

Verify email address

Once verified, you can start using all of GitHub's features to explore, build, and share projects.

Button not working? Paste the following link into your browser:
https://github.com/users/RamosColonAdrian2/emails/175024932/confirm_verification/17d7a4e823e6d45beca4ffb0fbfa4e4ea95f93f

Una vez introducidos los datos tendremos dos opciones de verificar el correo

La Primera opción es que nos llegue un correo de verificación donde solo tenemos que pulsar el botón “Verify email address”

You're almost done!

We sent a launch code to aramcol1101@g.educaand.es

→ Enter code

Didn't get your email? [Resend the code](#) or [update your email address](#).

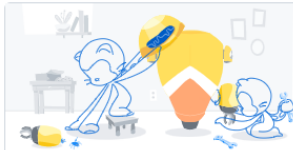
La Segunda opción que nos puede aparecer es introducir el código que nos ha sido enviado al correo electrónico de registro para verificar



Your email was verified.

What do you want to do first?

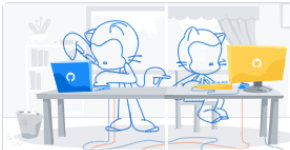
Every developer needs to configure their environment, so let's get your GitHub experience optimized for you.



Start a new project

Start a new repository or bring over an existing repository to keep contributing to it.

Create a repository



Collaborate with your team

Improve the way your team works together and get access to more features with an organization.

Create an organization



Learn how to use GitHub

Get started with an "Introduction to GitHub" course in our Learning Lab.

Start Learning

[Skip this for now >](#)



En mi caso, verifiqué el correo desde el botón “**Verify email address**”


Luego, GitHub nos pregunta que queremos hacer primero, a lo que nosotros pulsaremos en “**Create a new project**”



Create a new repository


A repository contains all project files, including the revision history. Already have a project repository elsewhere?
[Import a repository.](#)


Owner * Repository name *

 RamosColonAdrian2 / proyecto01 ✓

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

Description (optional)

☒  **Public**
Anyone on the internet can see this repository. You choose who can commit.

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

Initialize this repository with:

Skip this step if you're importing an existing repository.

☐ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

☐ **Add .gitignore**
Choose which files not to track from a list of templates. [Learn more.](#)

☐ **Choose a license**
A license tells others what they can and can't do with your code. [Learn more.](#)

[Create repository](#)


Nombre del repositorio

Marcar publico para no pagar la licencia privada

Se nos abrirá la configuración del repositorio a crear (repositorio web). En él deberemos darle un nombre a nuestro repositorio y dejar las opciones tal y como se muestran en pantalla, ya que mas adelante haremos algunos cambios de otra manera (readme). Una vez configurado cliquharemos en “**Create repository**”



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

 Set up in Desktop or [HTTPS](#) [SSH](#) <https://github.com/RamosColonAdrian/proyecto01.git>

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

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

```
echo "# proyecto01" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/RamosColonAdrian/proyecto01.git
git push -u origin main
```

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

```
git remote add origin https://github.com/RamosColonAdrian/proyecto01.git
git branch -M main
git push -u origin main
```

...or import code from another repository

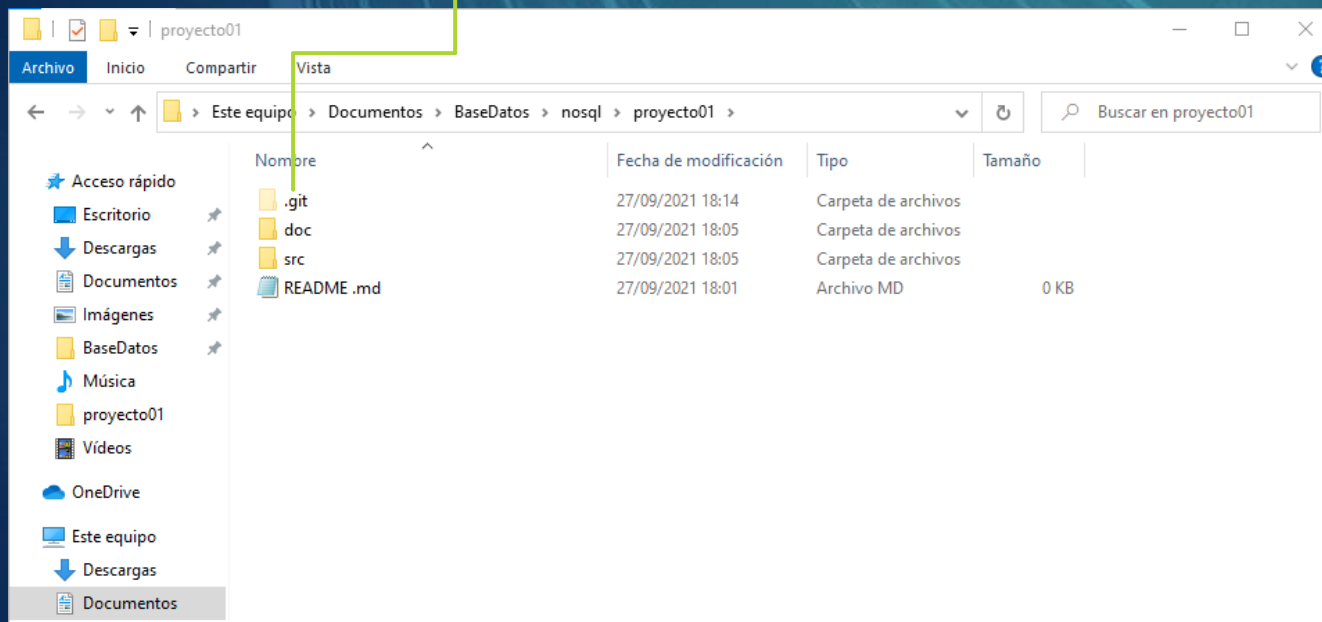
You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

[Import code](#)

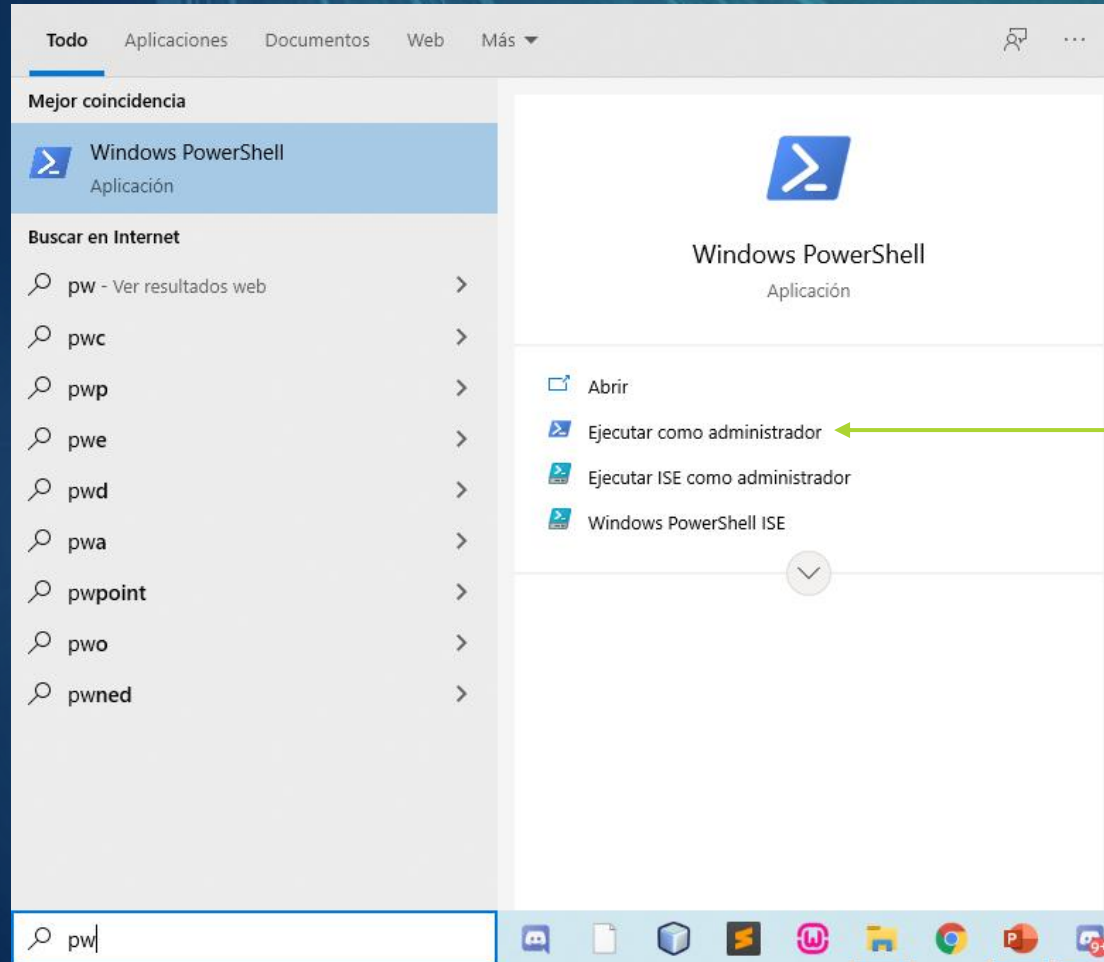
Una vez creado el repositorio nos saldrán unas líneas de comandos que se pueden tomar de referencia para realizar el repositorio local y vincularlo con el repositorio Web, yo como ayuda utilizo una [“CheatSheet”](#)



El `.git` que aparece en la carpeta lo crea el comando "`git init`" que utilizaremos después



Antes de comenzar con la vinculación de los repositorios, debemos crear una carpeta con el nombre de nuestro proyecto ("proyecto01") donde crearemos una carpeta llamada `doc` con un archivo `.pdf` donde añadiremos la documentación de nuestro proyecto, una carpeta llamada `src` donde tendremos nuestro código fuente, en este caso un `.txt` y un archivo llamado `README.md` donde añadiremos la descripción del creador del proyecto y una breve descripción de en qué consiste el proyecto



Una vez revisado los comandos utilizaremos **Windows PowerShell** como interfaz de consola proporcionada por Windows. Es muy sencillo, para ello escribiremos en el buscador de Windows: "PowerShell" y lo ejecutaremos como **administrador**



El primer comando que utilizaremos es "`cd`" y el directorio de la carpeta donde se encuentre nuestro repositorio local para posicionarnos en la ubicación deseada.

```
Administrador: Windows PowerShell
PS C:\> cd C:\Users\DAM1\Documents\BaseDatos\nosql\proyecto01
PS C:\Users\DAM1\Documents\BaseDatos\nosql\proyecto01> git init
Initialized empty Git repository in C:/Users/DAM1/Documents/BaseDatos/nosql/proyecto01/.git/
PS C:\Users\DAM1\Documents\BaseDatos\nosql\proyecto01> git add .
PS C:\Users\DAM1\Documents\BaseDatos\nosql\proyecto01> git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   README.md
        new file:   doc/Documentacion.docx
        new file:   src/codigo.txt

PS C:\Users\DAM1\Documents\BaseDatos\nosql\proyecto01> git commit -m "Initial commit"
[master (root-commit) 07d3d0e] Initial commit
3 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 README.md
create mode 100644 doc/Documentacion.docx
create mode 100644 src/codigo.txt
```

Archivos
listos para
hacer "`git
commit`"

El comando "`git
status`" es un
comando
relativamente sencillo.
Simplemente, muestra
lo que ha ocurrido
con los comandos `git
add` y `git commit`.

El comando "`git init`" crea un nuevo repositorio de Git. Puede utilizarse para convertir un proyecto existente y sin versión en un repositorio de Git, o para inicializar un nuevo repositorio vacío.



El comando "`git add`" añade un cambio del directorio de trabajo en el entorno de ensayo. De este modo, indica a Git que quieres incluir actualizaciones en un archivo concreto en la próxima confirmación. Sin embargo, "`git add`" no afecta al repositorio de manera significativa: en realidad, los cambios no se registran hasta que ejecutas "`git commit`"

El comando "`git commit`" guardará todos los cambios hechos en la zona de montaje o área de preparación, junto con una breve descripción del usuario, en un "`commit`" al repositorio local. El comentario escrito entre dobles comillas nos permitirá una vez creado el repositorio diferenciar las diferentes versiones que subamos de nuestro programa

Seguimos en la siguiente
Diapositiva





El comando "**git remote**" te permite crear, ver y eliminar conexiones con otros repositorios.

En este caso estamos añadiendo una conexión remota vinculada con el repositorio (URL que nos ofrece la pagina del repositorio alojado en GitHub) con el alias "origin"

```
PS C:\Users\DAM1\Documents\BaseDatos\nosql\proyecto01> git remote add origin https://github.com/RamosColonAdrian/proyecto01.git
PS C:\Users\DAM1\Documents\BaseDatos\nosql\proyecto01> git push -u https://github.com/RamosColonAdrian/proyecto01.git
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (5/5), 375 bytes | 375.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/RamosColonAdrian/proyecto01.git
* [new branch] master -> master *
Branch 'master' set up to track remote branch 'master' from 'https://github.com/RamosColonAdrian/proyecto01.git'.
```

Descripción de los
archivos subidos

El comando "**git push**" te permite subir los "commits" desde tu rama (branch) local en tu repositorio "git" local al repositorio remoto.

* Una particularidad a destacar son las ramas o Branch, que aunque no hayamos hecho uso de ellas, a la hora de crear el repositorio podemos añadir ramas donde subir documentos determinados. De esta manera podremos tener un lugar (la rama) donde hacer pruebas sin que repercutir a la rama master o principal pudiendo hacer prueba-error de una manera mas fácil y segura. El código para crear una rama sería este "**git branch [nombre de la rama]**"



Si es la primera vez que usas GitHub deberás utilizar los comandos:

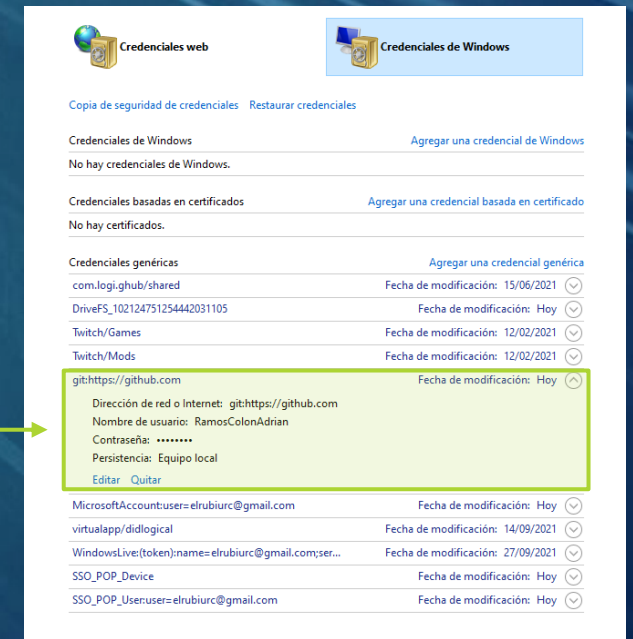
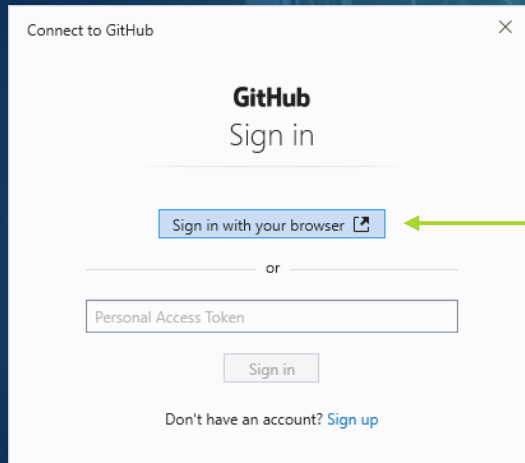
```
"git config --global user.email "Tu e-mail" "  
"git config --global user.name "Tu nombre" "
```

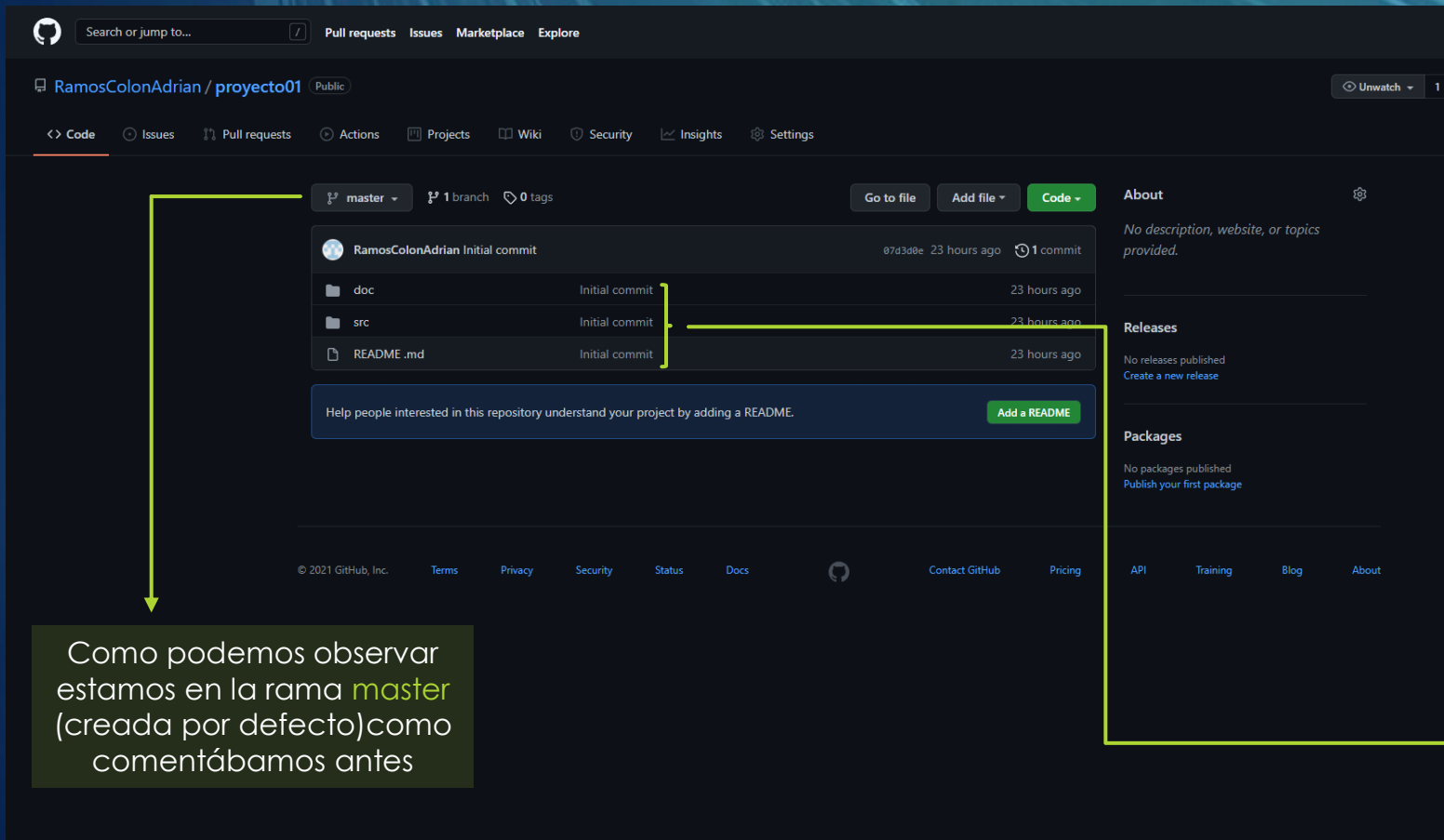
```
PS C:\Users\DAM1\Documents\BaseDatos\nosql\proyecto01> git config --global user.email "elrubiurc@gmail.com"  
PS C:\Users\DAM1\Documents\BaseDatos\nosql\proyecto01> git config --global user.name "Adrian Ramos Colon"  
PS C:\Users\DAM1\Documents\BaseDatos\nosql\proyecto01> git push  
info: please complete authentication in your browser...  
Everything up-to-date  
PS C:\Users\DAM1\Documents\BaseDatos\nosql\proyecto01>
```

Una vez configurados nuestros datos haremos un "git push" (aunque no tengamos nada comiteado) para que nos redirija a la autenticación desde el navegador.

Cuando hayamos terminado podremos ver desde *Panel de control\Todos los elementos de Panel de control\Administrador de credenciales* las credenciales de GitHub

Nota: Windows almacena la identificación del usuario GitHub, si entramos con otro usuario hay que borrar el anterior con Administrador de Credenciales

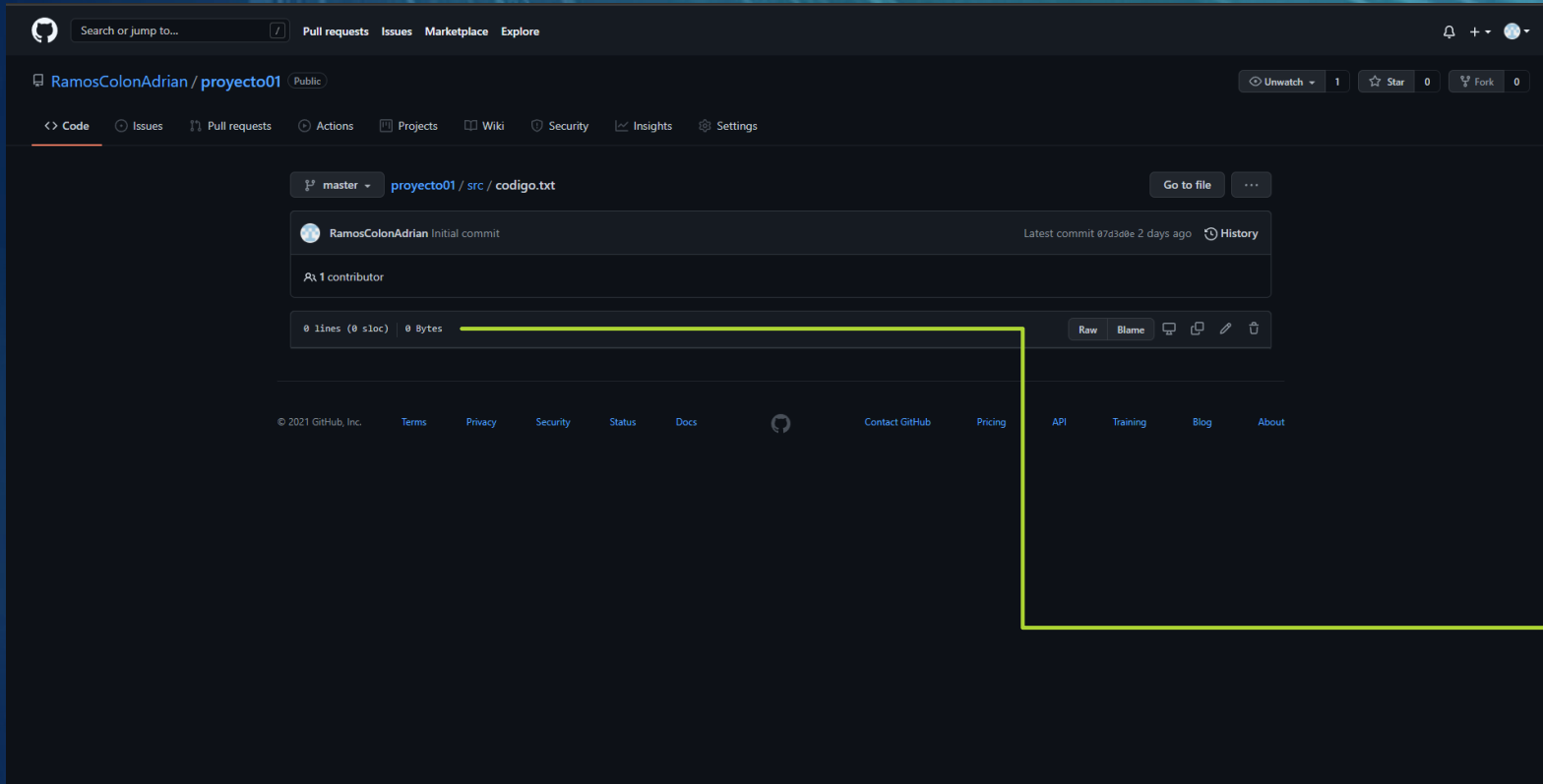




Como podemos observar estamos en la rama **master** (creada por defecto) como comentábamos antes

Después de seguir estos pasos habremos logrado crear nuestro repositorio sin ningún problema

Aquí nos indica el ultimo **commit** utilizado, gracias a esto podemos hacer un control exhaustivo de las versiones de nuestro programa según la subida del mismo al repositorio



Vamos a empezar viendo como podemos actualizar los archivos del repositorio web una vez editados desde el repositorio local

Como podemos observar nuestro `.txt` de la carpeta `src` se encuentra vacío. Vamos a ver como poder editarla y subir la `update` al repositorio web



```
code.txt: Bloc de notas
Archivo Edición Formato Ver Ayuda
Este documnto de prueba ha sido modificado despues de crear el reposirorio.
Vamos a ver como subir los archivos modificados en el repositorio local
al repositorio remoto
```

Primero editaremos añadiendo información nueva en el archivo

```
PS C:\Users\DAM1\Documents\BaseDatos\nosql\proyecto01> git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   src/code.txt
```

Vemos que si miramos el estado del fichero nos indica que un archivo ha sido modificado

```
no changes added to commit (use "git add" and/or "git commit -a")
PS C:\Users\DAM1\Documents\BaseDatos\nosql\proyecto01> git add src/code.txt
PS C:\Users\DAM1\Documents\BaseDatos\nosql\proyecto01> git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   src/code.txt
```

Añadimos el fichero modificado

```
PS C:\Users\DAM1\Documents\BaseDatos\nosql\proyecto01> git commit -m "Update 1 code.txt"
[master 9f5b1e1] Update 1 code.txt
 1 file changed, 1 insertion(+), 1 deletion(-)
PS C:\Users\DAM1\Documents\BaseDatos\nosql\proyecto01> git push
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 16 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (8/8), 695 bytes | 695.00 KiB/s, done.
Total 8 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 1 local object.
To https://github.com/RamosColonAdrian/proyecto01.git
   c8c2bef..9f5b1e1  master -> master
PS C:\Users\DAM1\Documents\BaseDatos\nosql\proyecto01>
```

Una vez añadido, ejecutamos un **commit** especificando en la etiqueta la información de la nueva update para luego saber con claridad que se ha modificado

Por ultimo ejecutamos **git push** y habremos subido la update al repositorio remoto



Search or jump to... Pull requests Issues Marketplace Explore

RamosColonAdrian / proyecto01 Public

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

master 1 branch 0 tags

Go to file Add file Code

RamosColonAdrian	Update 1 code.txt	9f5b1e1 1 hour ago	3 commits
doc	Initial Commit		1 hour ago
src	Update 1 code.txt		1 hour ago
README.md	Initial Commit		1 hour ago

Help people interested in this repository understand your project by adding a README. Add a README

About
No description, website, or topics provided.

Releases
No releases published
Create a new release

Packages
No packages published
Publish your first package

© 2021 GitHub, Inc. Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About

Si accedemos al repositorio remoto observaremos que hay un nuevo **commit** en la carpeta src



Search or jump to... Pull requests Issues Marketplace Explore

RamosColonAdrian / proyecto01 Public Unwatch 1

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

master proyecto01 / src / code.txt Go to file ...

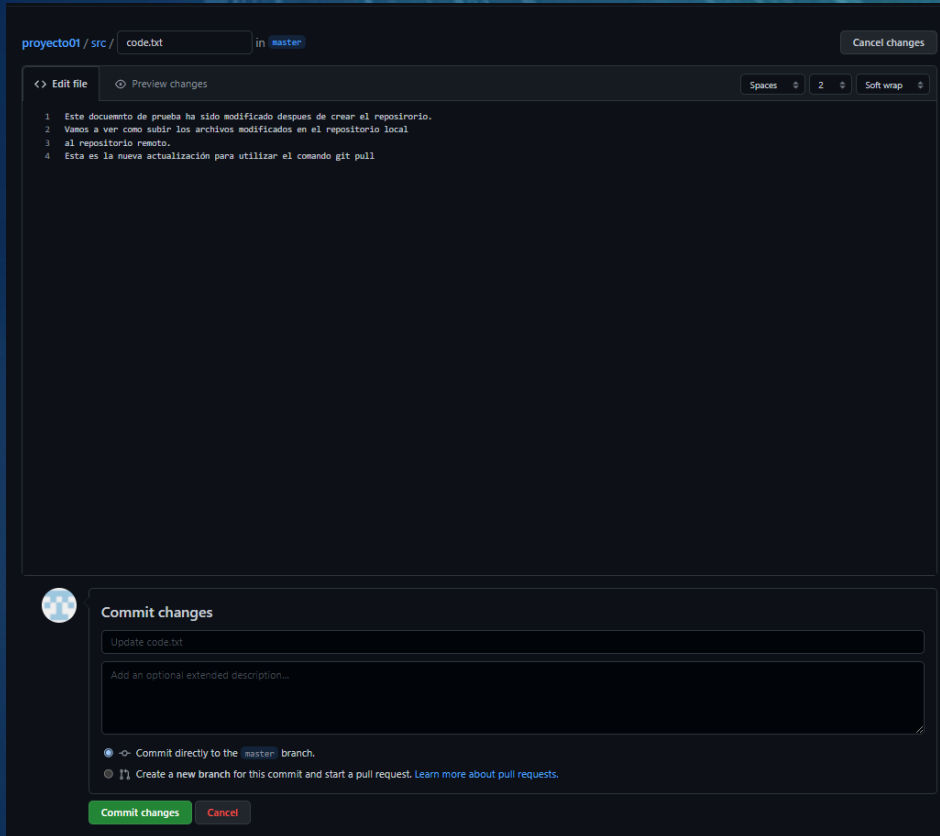
RamosColonAdrian Update 1 code.txt Latest commit 9f5b1e1 1 hour ago History

1 contributor

3 lines (3 sloc) 171 Bytes Raw Blame

```
1 Este documento de prueba ha sido modificado despues de crear el repositorio.  
2 Vamos a ver como subir los archivos modificados en el repositorio local  
3 al repositorio remoto.
```

Podemos observar que el documento `code.txt` se actualizó con la edición que hicimos en local



IMPORTANTE
Guardar el commit

Si queremos hacer lo contrario que en la diapositiva anterior deberemos conocer el comando `git pull`. En este caso la edición la realizaremos desde el repositorio remoto. El comando `git pull` nos cargará la información del repositorio remoto en nuestro repositorio local. La ventaja de esto es que podemos trabajar desde la nube sin necesidad de estar en nuestro repositorio local y con la posibilidad de guardarlo en todo momento.



```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\Rubi_01> cd C:\Users\DAM1\Documents\BaseDatos\nosql\proyecto01
PS C:\Users\DAM1\Documents\BaseDatos\nosql\proyecto01> git pull
remote: Enumerating objects: 19, done.
remote: Counting objects: 100% (19/19), done.
remote: Compressing objects: 100% (9/9), done.
remote: Total 16 (delta 7), reused 7 (delta 3), pack-reused 0
Unpacking objects: 100% (16/16), 2.03 KiB | 83.00 KiB/s, done.
From https://github.com/RamosColonAdrian/proyecto01
* branch      master      -> FETCH_HEAD
Updating c8c2bef..0b8a93c
Fast-forward
 src/code.txt | 4 ++++
 1 file changed, 4 insertions(+)
PS C:\Users\DAM1\Documents\BaseDatos\nosql\proyecto01>
```

code.txt: Bloc de notas

Archivo Edición Formato Ver Ayuda

Este documnto de prueba ha sido modificado despues de crear el reposirorio.
Vamos a ver como subir los archivos modificados en el repositorio local
al repositorio remoto.
Esta es la nueva actualización para utilizar el comando git pull .

Veremos como
automáticamente detecta el
cambio y lo carga en nuestro
repositorio local

IMPORTANTE

No realizar cambios en los dos
repositorios sin actualizar la
información antes por
separado ya que si lo
hacemos, al hacer *git pull* o *git push*
lo hará pero no será una
copia idéntica ya que dará
ciertos errores que pueden
hacernos perder parte del
documento



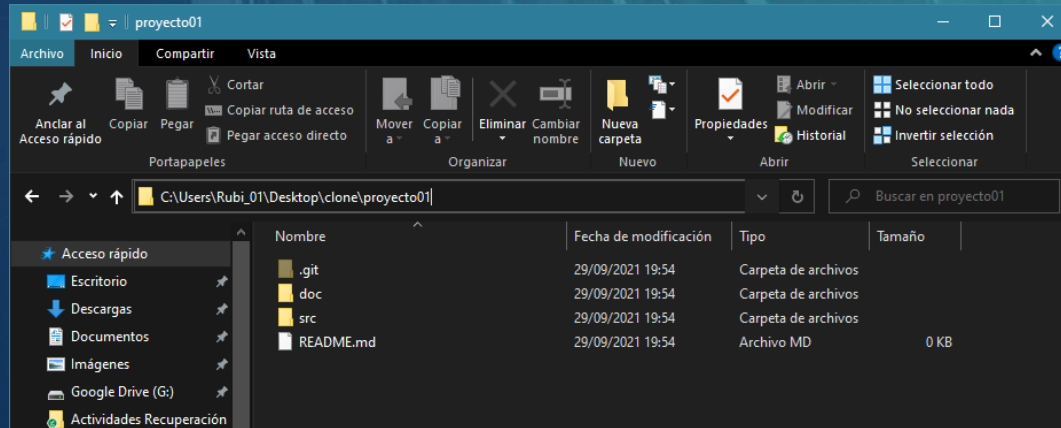
```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\Rubi_01> cd C:\Users\Rubi_01\Desktop\clone
PS C:\Users\Rubi_01\Desktop\clone> git clone https://github.com/RamosColonAdrian/proyecto01.git
Cloning into 'proyecto01'...
remote: Enumerating objects: 22, done.
remote: Counting objects: 100% (22/22), done.
remote: Compressing objects: 100% (12/12), done.
remote: Total 22 (delta 7), reused 12 (delta 3), pack-reused 0
Receiving objects: 100% (22/22), 7.73 KiB | 7.73 MiB/s, done.
Resolving deltas: 100% (7/7), done.
PS C:\Users\Rubi_01\Desktop\clone> git clone
```

Otro ejemplo que podemos realizar es el de clonar el repositorio remoto en un directorio de nuestro pc para convertirlo en el repositorio local.

El comando que se utiliza en este caso es *git clone*



Primero utilizaremos el comando *git clone* con el url del repositorio remoto y luego posicionado en la carpeta deseada, ejecutamos git clone para clonar el repositorio en el directorio



Adrián Ramos Colón 1ºDAM