

# Introducción a Git y GitHub como herramientas de control de versiones y gestión de proyectos

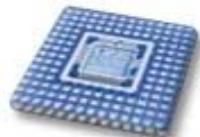
Oferta formativa del Instituto de Ciencias de la Educación  
Universidad de Sevilla

14 Octubre 2021  
Daniel Gutiérrez Galán



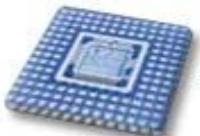
# Índice

- **Objetivos.**
- Introducción a Git.
- Introducción a GitHub.
- Metodología de creación y gestión de repositorios.
- Gestión de proyectos colaborativos.
- Integrando Git en herramientas de trabajo.



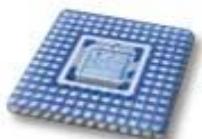
# Objetivos

- Conocer los conceptos principales sobre Git.
- Conocer qué es un repositorio y cómo funciona.
- Aprender a crear y utilizar una cuenta en la plataforma GitHub.
- Proporcionar una metodología organizada y sencilla para gestionar distintos repositorios privados y públicos.
- Gestionar proyectos colaborativos a través de la creación de ramas y peticiones pull-request.
- Integrar Git en herramientas de desarrollo.



# Evaluación

- “*EL ICE EXPEDIRÁ UN DOCUMENTO ACREDITATIVO A LOS PARTICIPANTES QUE HAYAN ASISTIDO AL 80% DE LAS HORAS PRESENCIALES Y HAYAN SUPERADO, EN SU CASO, OTROS CRITERIOS DE EVALUACIÓN.*”

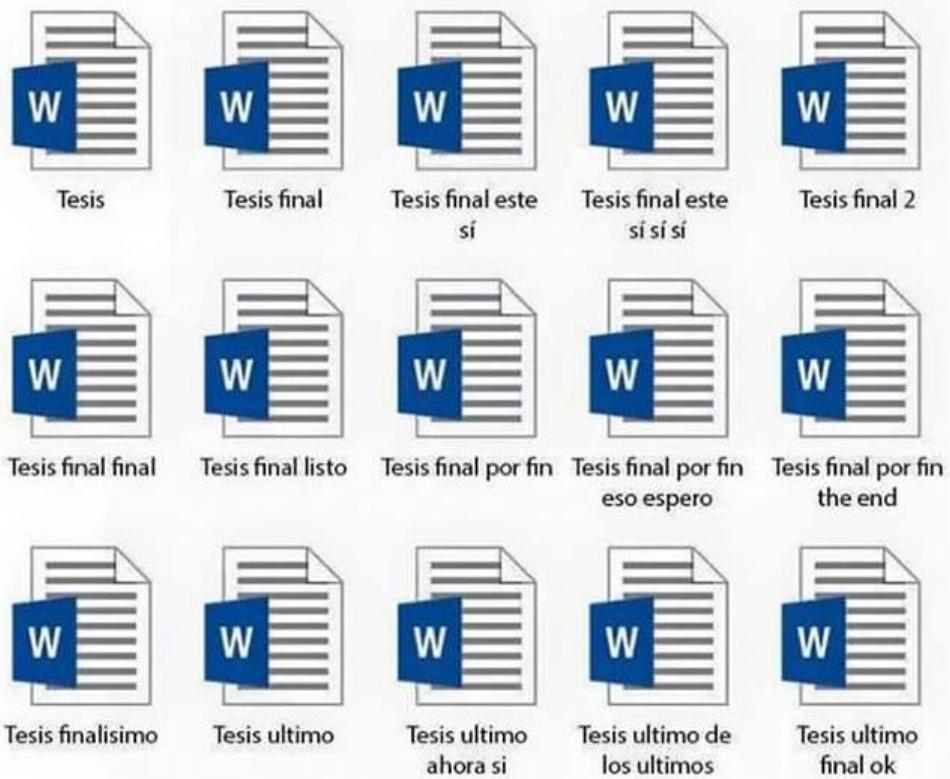


# Índice

- Objetivos.
- **Introducción a Git.**
- Introducción a GitHub.
- Metodología de creación y gestión de repositorios.
- Gestión de proyectos colaborativos.
- Integrando Git en herramientas de trabajo.

# ¿Qué es un control de versiones?

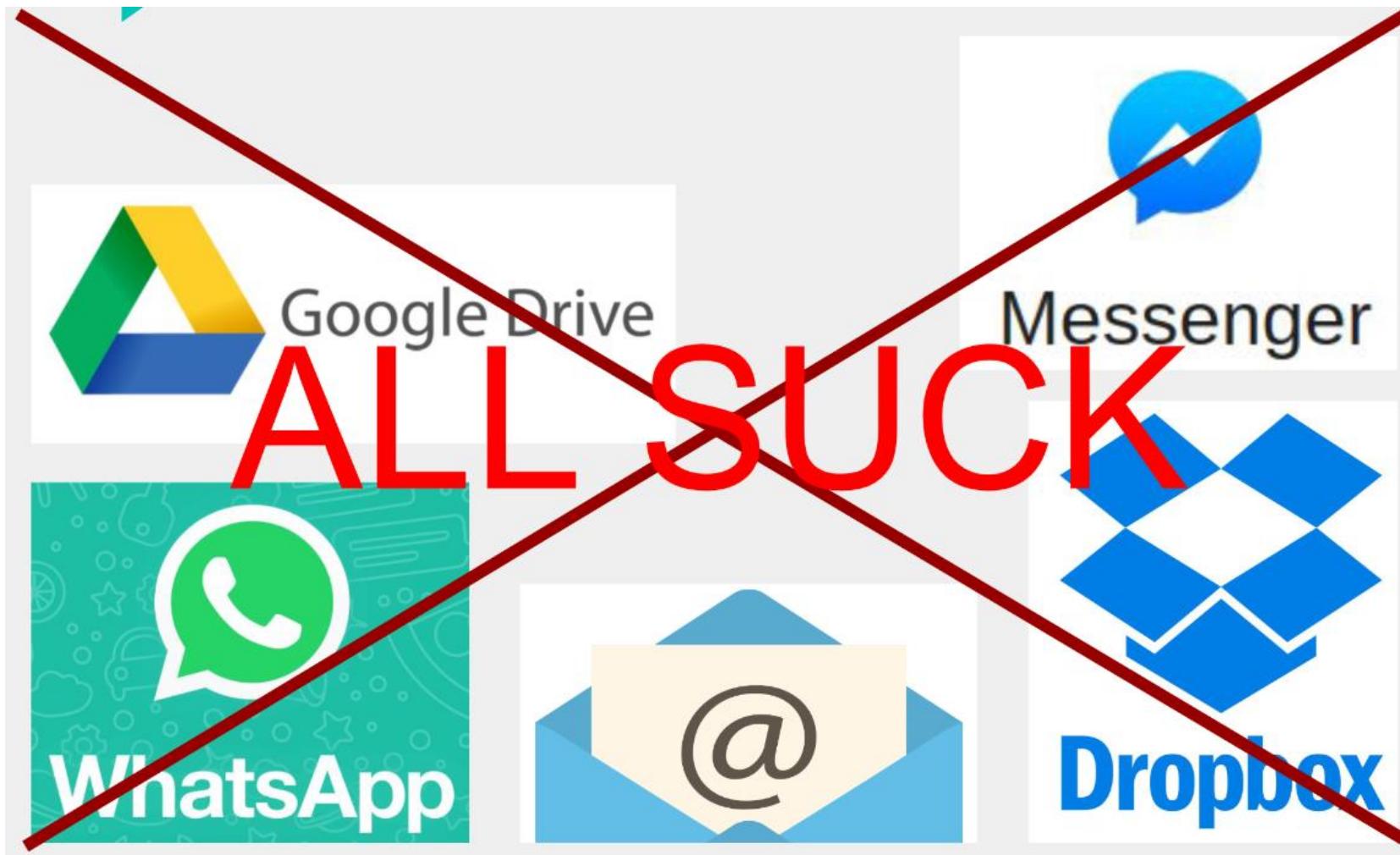
## EL PROCESO DE TU TESIS



# ¿Qué es un control de versiones?



# ¿Qué es un control de versiones?

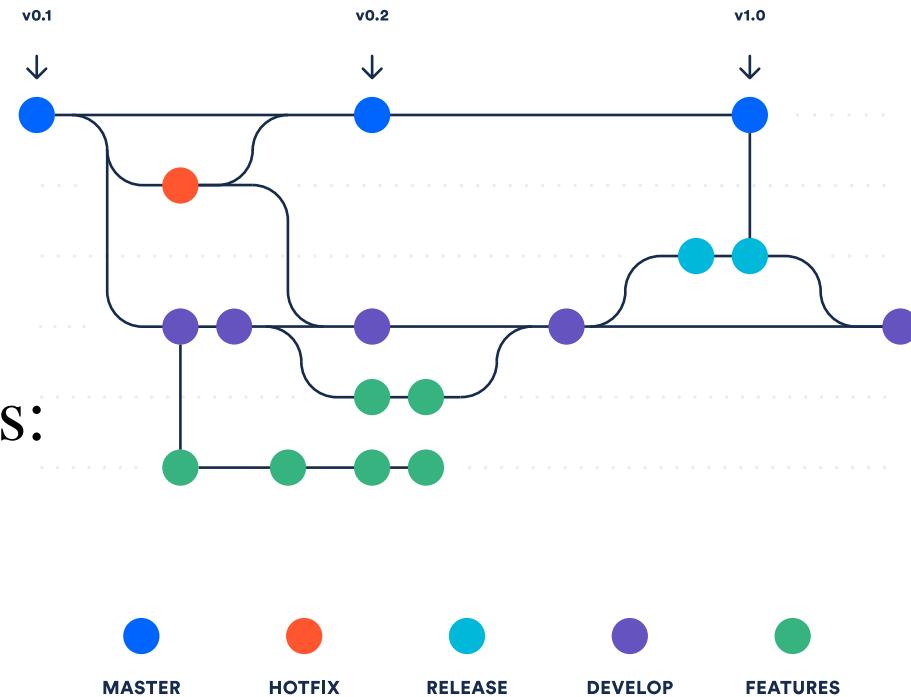


# ¿Qué es un control de versiones?

- Un control de versiones es:
  - La buena práctica de rastrear y gestionar los cambios en el código fuente de un software.
- Un sistema de control de versiones (VCS) es:
  - Una herramienta software que ayuda a gestionar los cambios en el código fuente a lo largo del tiempo.
  - También monitoriza y gestiona cambios en un sistema de archivos.

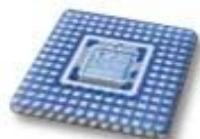
# ¿Qué es un control de versiones?

- Los VCS permiten:
  - Trabajar de forma colaborativa.
    - Compartir código.
    - Integrar código.
  - Monitorizar acciones a archivos o directorios:
    - Adición.
    - Eliminación.
    - Modificación.
  - Organizar el proyecto.



# ¿Qué es un control de versiones?

- Ventajas:
  - Trabajar más rápido y cómodo.
  - Reducen el tiempo de desarrollo y los bugs.
  - Permite volver a versiones anteriores funcionales.
  - Trabajar en paralelo en las versiones actuales y en las nuevas versiones.
  - Tener un historial del código para detectar el origen de un problema  
→ trazabilidad.



# ¿Qué es un control de versiones?

- ¿Cómo lo hace?
  - Se guardan “fotografías” del contenido de cada archivo/directorio.
  - Se almacenan en una base de datos especial.
  - El historial de cambios se obtiene comparando esas “fotografías”.

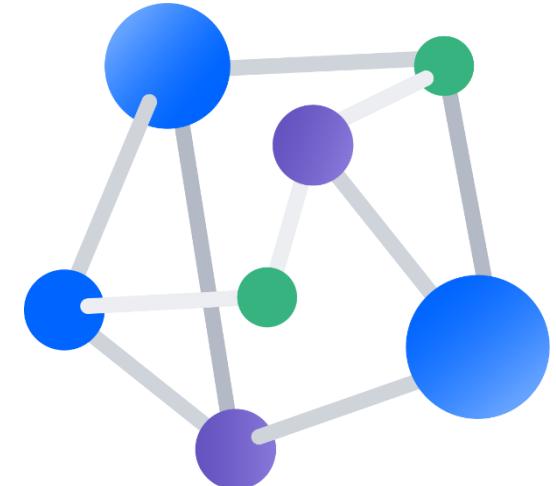


# ¿Qué es Git?

- Es un VCS (software de control de versiones) distribuido de código abierto y gratuito.
- Desarrollado por Linus Torvalds en 2005.
- Mantenimiento activo.
- Millones de proyectos usan Git; existen otras alternativas.

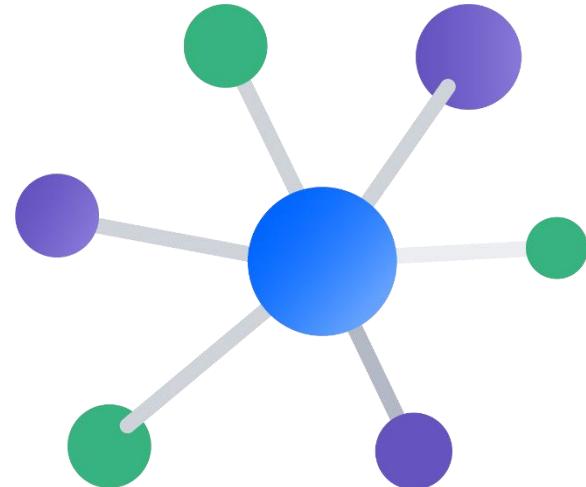
# ¿Qué es Git?

- Arquitectura distribuida (DVCS):
  - Existe una copia del historial completo de todos los cambios en cada nodo (copia de seguridad).
  - Además de la propia versión del trabajo.
  - Si un nodo falla, existe una en otro nodo para ser restaurada → Mejor tolerancia a fallos.
  - Funciona si estás offline.
  - Seguridad: SHA1.



# ¿Qué **NO** es Git?

- Arquitectura centralizada (CVCS):
  - Ejemplo: SVN (Apache Subversion).
  - Una sola copia del historial → Un solo punto de error.
  - Se puede producir pérdidas de datos.
  - No se pueden hacer commits mientras dure el fallo.
  - No funciona si estás offline.



# Conceptos básicos

- **Repositorio:**
  - Almacenamiento virtual de tu proyecto que te permite guardar las versiones del código.
  - Eje central del proyecto.
  - Puede ser un archivo, un conjunto de archivos, o un conjunto de carpetas que contienen más archivos y carpetas.

# Conceptos básicos

- **Blobs:**
  - Binary Large Object: archivo binario con los datos del archivo pero sin metadatos.
  - Cada versión de un archivo está representada por un blob.
  - En la base de datos de Git, están nombrados por un código SHA1 único. Es lo que se usa para el control de versiones.

# Conceptos básicos

- Ramas o “branches”:
  - Versión paralela del repositorio que está contenida dentro del propio repositorio.
  - Por defecto, rama main. Las otras ramas no afectan a la rama main.
  - Permite implementar nuevas funcionalidades sin modificar la funcionalidad original.
  - Las ramas se pueden fusionar, de forma que las nuevas implementaciones se añadan a la rama main.

# Conceptos básicos

- **Forks o bifurcaciones:**
  - Copia personal del repositorio de otro usuario en tu cuenta.
  - Permite hacer cambios sin afectar al repositorio original.
  - Se puede solicitar incluir los cambios al repositorio original.
  - Ambos repositorios permanecen sincronizados, por lo que los cambios en el repositorio original afectan al repositorio del fork.
  - Sirven para tomar un proyecto original como base para crear otro.

# Conceptos básicos

- **Clonar:**
  - Copiar un repositorio que está almacenado online en una carpeta local de tu ordenador.
  - Puedes navegar entre ramas en un repositorio clonado.
  - Se descarga todo el historial de cambios del repositorio.
  - Los cambios afectarían directamente al repositorio original.
  - Es un comando.

# Conceptos básicos

- **Commit:**
  - Guarda los cambios realizados en los archivos en el repositorio local.  
También se le conoce como “revisión”.
  - Crea un blob de cada archivo seleccionado para ser guardado con un identificador o hash único.
  - Suele hacerse un commit cuando se termina una implementación final o parcial →**No abusar de los commits.**

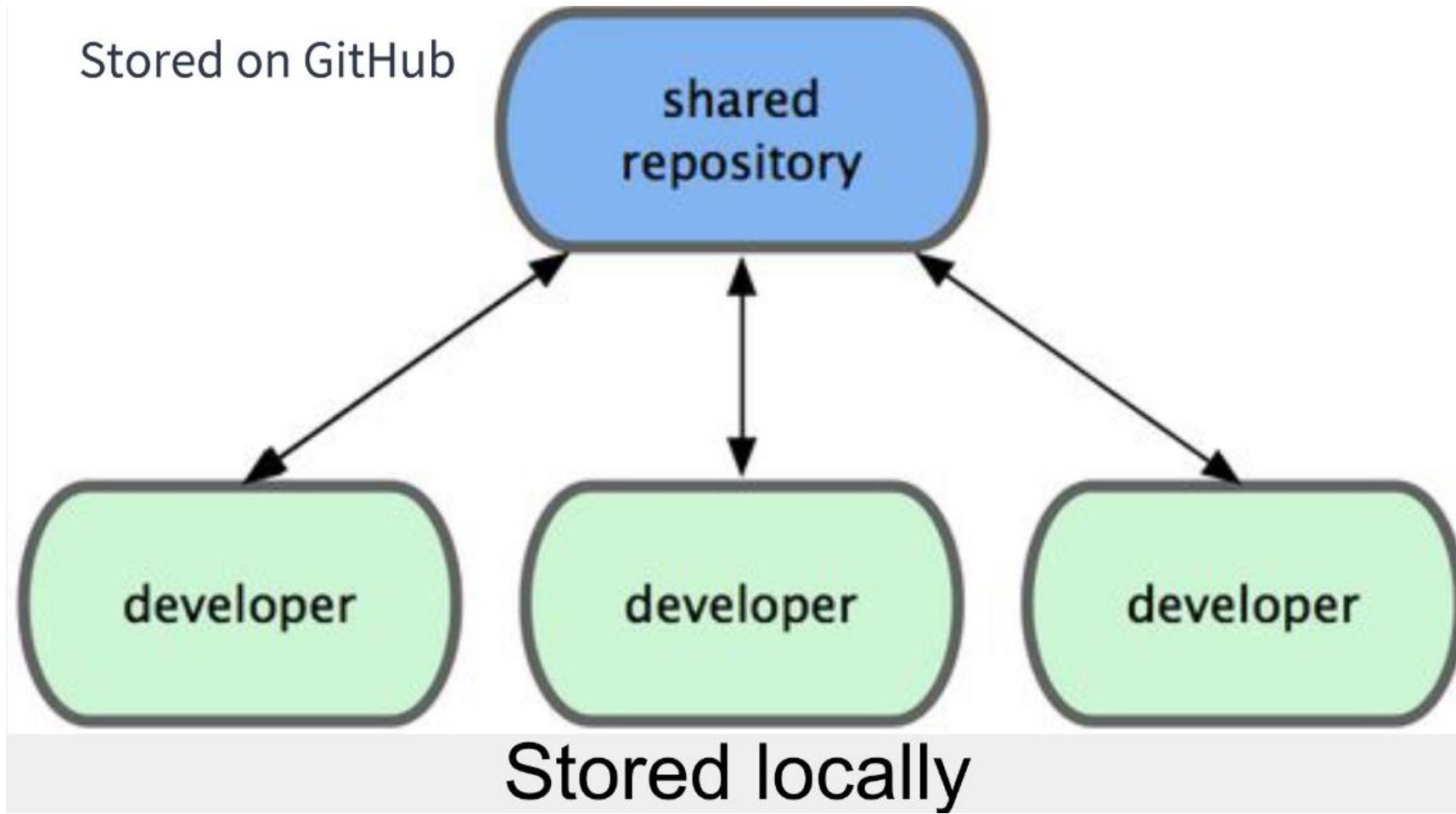
# Conceptos básicos

- **Push:**
  - Subir al repositorio online (copia remota) los commits de git locales.
  - Si no se hace push, los cambios son locales (offline), y el resto de usuarios no tiene la copia más actualizada.
  - Es como publicar en una red social tus últimos cambios.

# Conceptos básicos

- **Pull:**
  - Descarga los cambios del repositorio remoto al repositorio local.
  - Operación contraria al push.
  - Sirve para tener los cambios realizados por los otros colaboradores.
- **¡Siempre primero pull y luego push! → Si no, conflicto de versiones en el repositorio remoto. Mejor resolverlo en el repositorio local.**

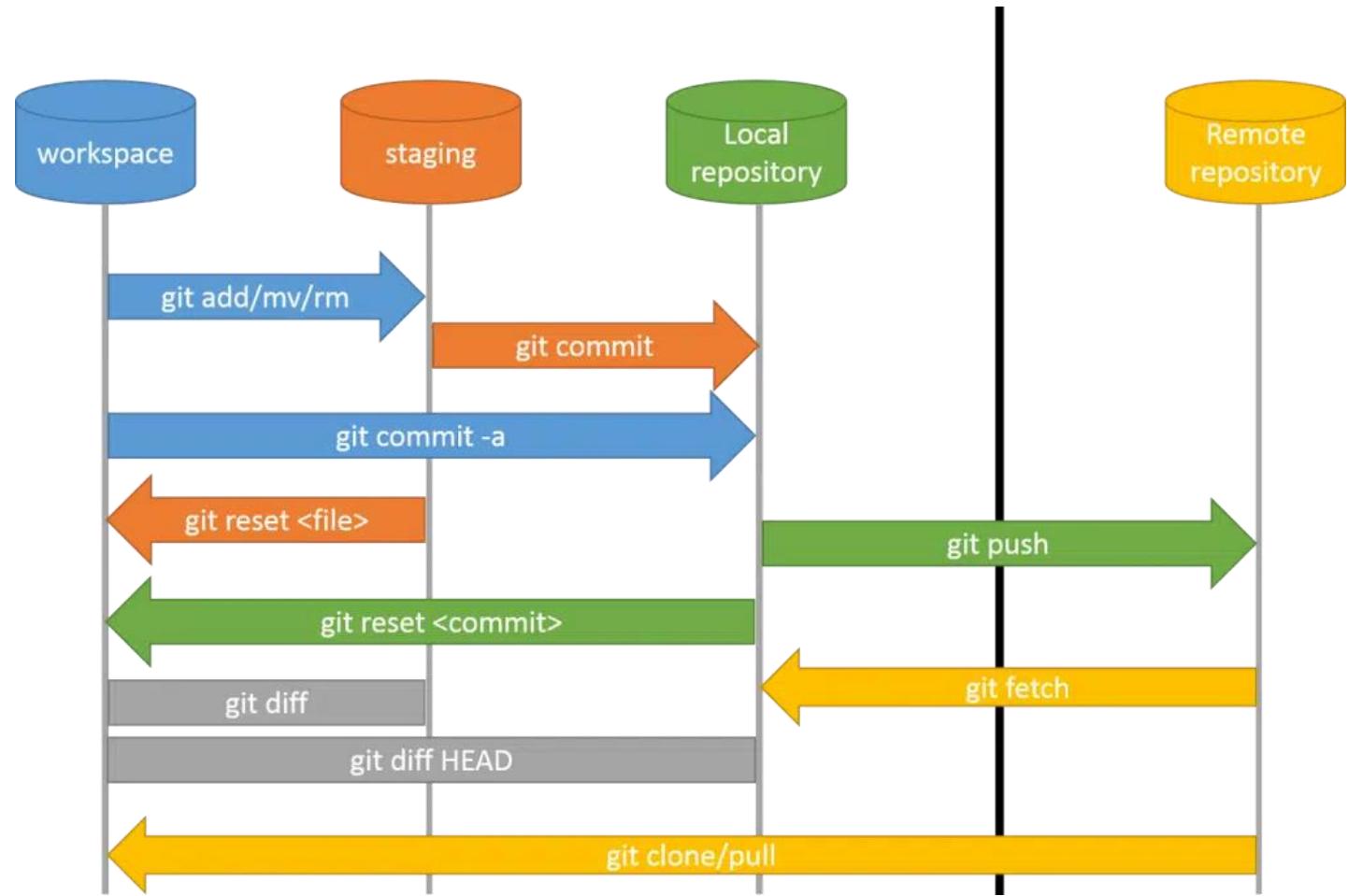
# ¿Cómo funciona?



# ¿Cómo funciona?

- Los archivos tienen tres estados:
  - No modificado: el archivo es idéntico al archivo almacenado en el repositorio.
  - Modificado: el archivo tiene alguna diferencia con el archivo almacenado en el repositorio
  - Staged: el archivo está listo para ser subido al repositorio (antes del commit). Todavía se puede modificar.

# ¿Cómo funciona?



# Documentación

- Documentación interactiva de comandos:
  - <http://ndpsoftware.com/git-cheatsheet.html#loc=index>;
- Aprender a usar Git branching mediante problemas:
  - [https://learngitbranching.js.org/?locale=es\\_ES](https://learngitbranching.js.org/?locale=es_ES)

# Índice

- Objetivos.
- Introducción a Git.
- **Introducción a GitHub.**
- Metodología de creación y gestión de repositorios.
- Gestión de proyectos colaborativos.
- Integrando Git en herramientas de trabajo.

# ¿Qué es?

- GitHub es:
  - Una plataforma de host para almacenar los repositorios online y trabajar de forma colaborativa que usa el software de control de versiones Git.
  - Permite gestionar proyectos de desarrollo tanto comerciales como de software libre.
  - También permite distribuir el software.
  - Es gratis, aunque con algunas limitaciones.

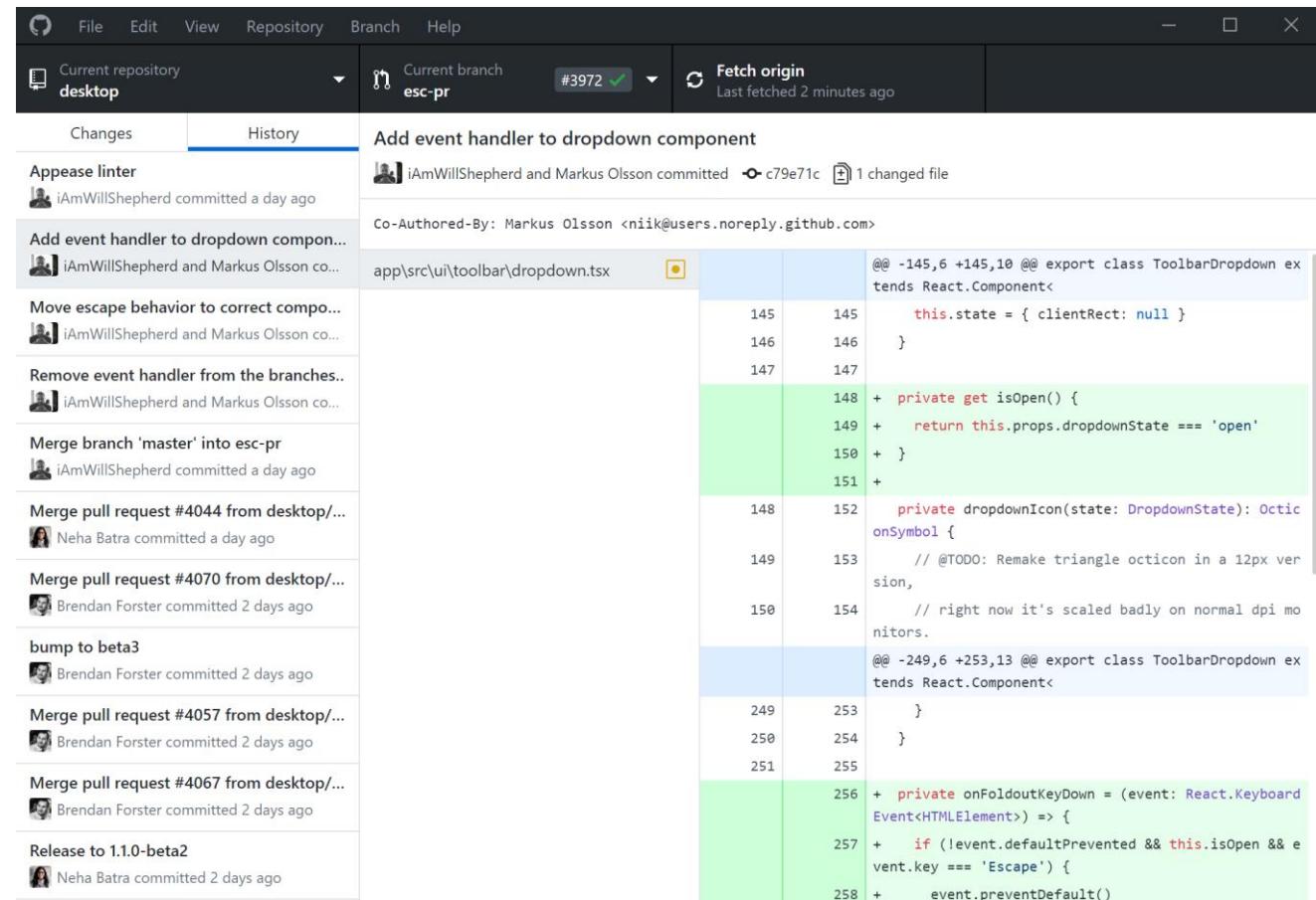


# ¿Cómo podemos trabajar con Git en GitHub?

- Dos más una formas de trabajar con GitHub:
  - A través de la línea de comandos:
    - Para usuarios avanzados o que necesiten un control absoluto del sistema de control de versiones.
  - A través de una interfaz gráfica:
    - Para usuarios iniciales/medios o que no necesiten tanto control. También, cuando se trabaja en un repositorio propio sin colaboradores.

# ¿Cómo podemos trabajar con Git en GitHub?

- GitHub Desktop:
  - Instala Git.
  - Permite gestionar proyectos (crear, clonar, etc).
  - Permite hacer commits, volver a versiones anteriores, cambiar de ramas, etc.
  - **Requiere cuenta.**



The screenshot shows the GitHub Desktop application interface. At the top, it displays the current repository as "desktop", the current branch as "esc-pr" with a pull request "#3972" checked, and the status of "Fetch origin" as "Last fetched 2 minutes ago". Below this, there are two tabs: "Changes" and "History", with "Changes" currently selected. The main pane lists several commit messages from different authors (iAmWillShepherd, Markus Olsson, Neha Batra, Brendan Forster) across various branches and pull requests. On the right side, a detailed view of a pull request's diff is shown for the file "app\src\ui\toolbar\dropdown.tsx". The diff highlights changes made between line 145 and 258, including new code for handling dropdown behavior and octicons.

```
@@ -145,6 +145,10 @@ export class ToolbarDropdown extends React.Component<
    this.state = { clientRect: null }
}

+ private get isOpen() {
+   return this.props.dropdownState === 'open'
+ }

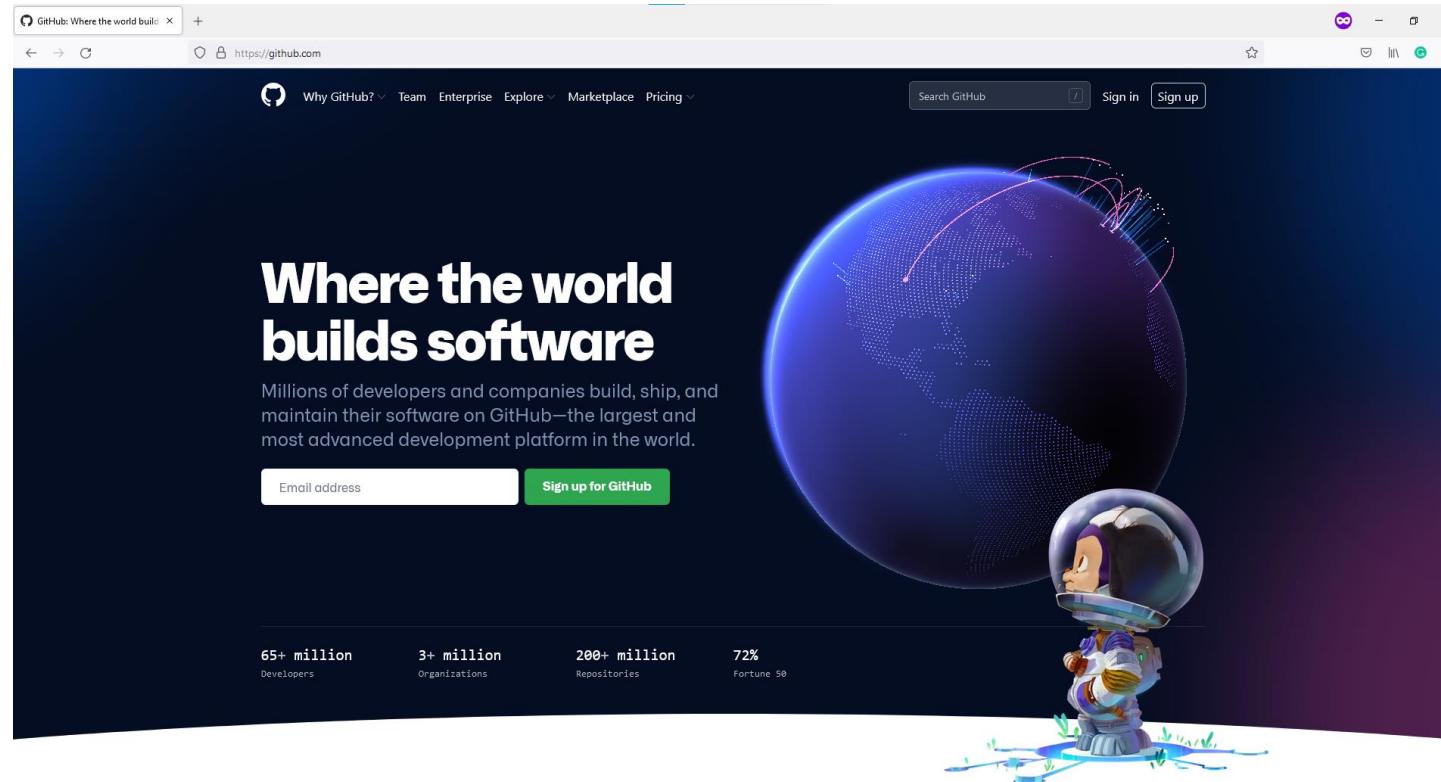
private dropdownIcon(state: DropdownState): OcticonSymbol {
  // @TODO: Remake triangle octicon in a 12px version,
  // right now it's scaled badly on normal dpi monitors.
@@ -249,6 +253,13 @@ export class ToolbarDropdown extends React.Component<
}

+ private onFoldoutKeyDown = (event: React.KeyboardEvent<HTMLElement>) => {
+   if (!event.defaultPrevented && this.isOpen && event.key === 'Escape') {
+     event.preventDefault()

```

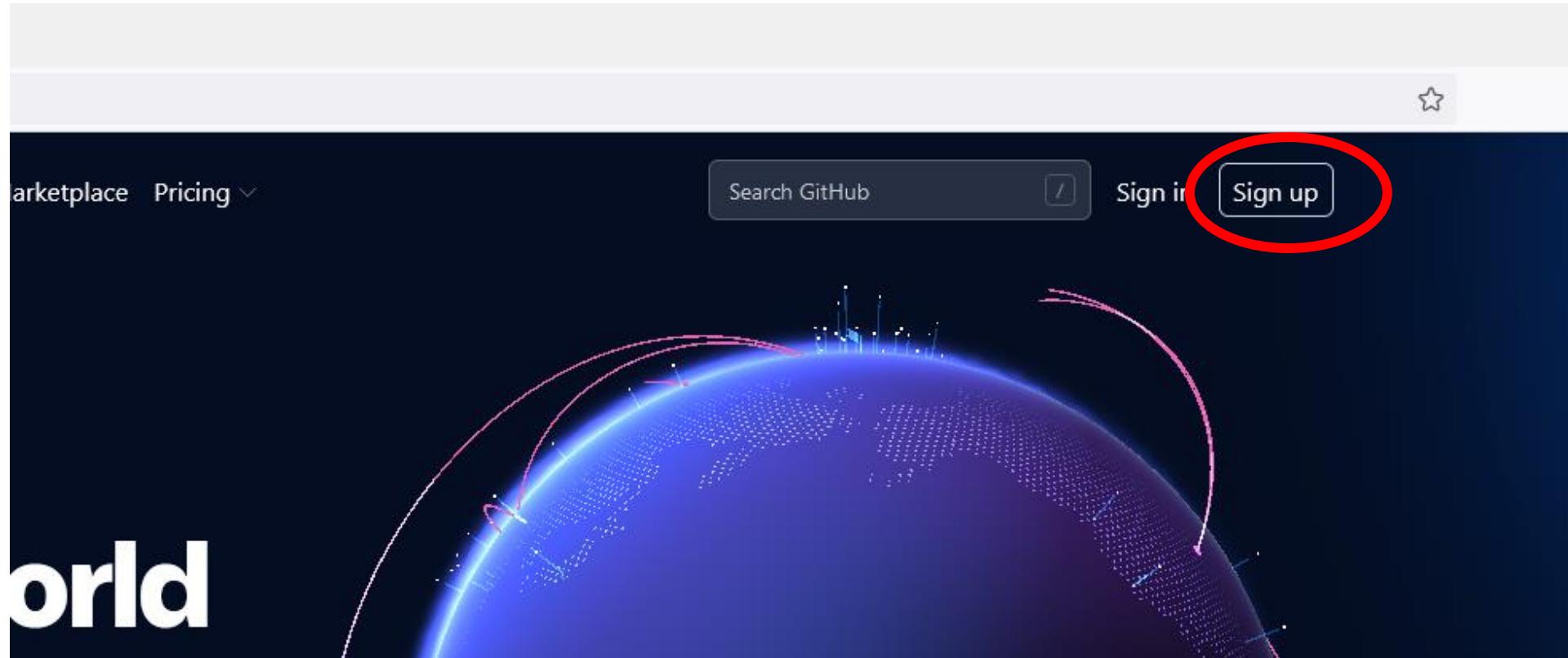
# ¿Cómo podemos crearnos una cuenta?

- Accede a <https://github.com/>



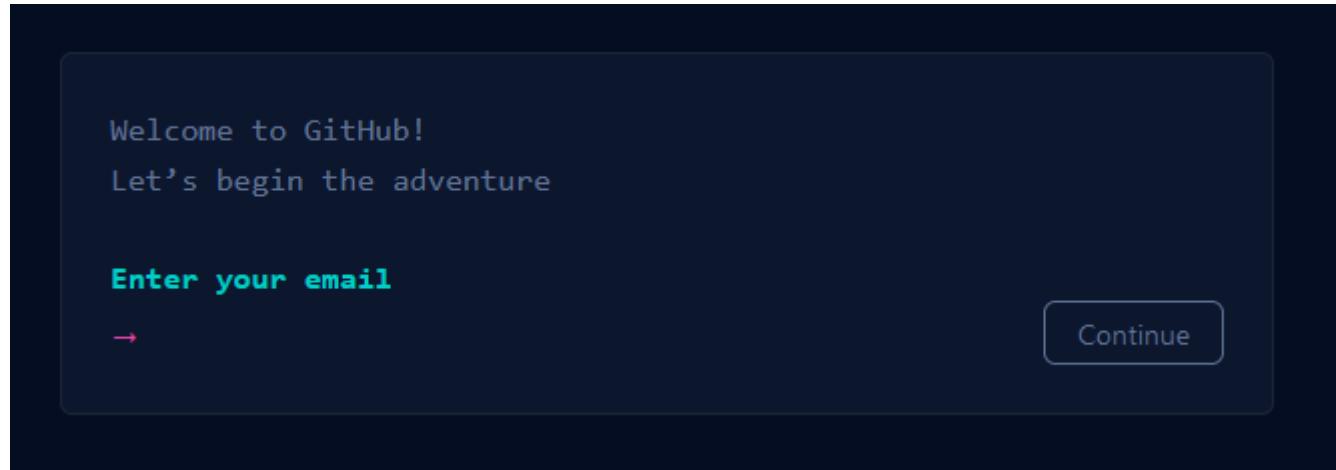
# ¿Cómo podemos crearnos una cuenta?

- Haz click en “Sign up” en la esquina superior derecha.



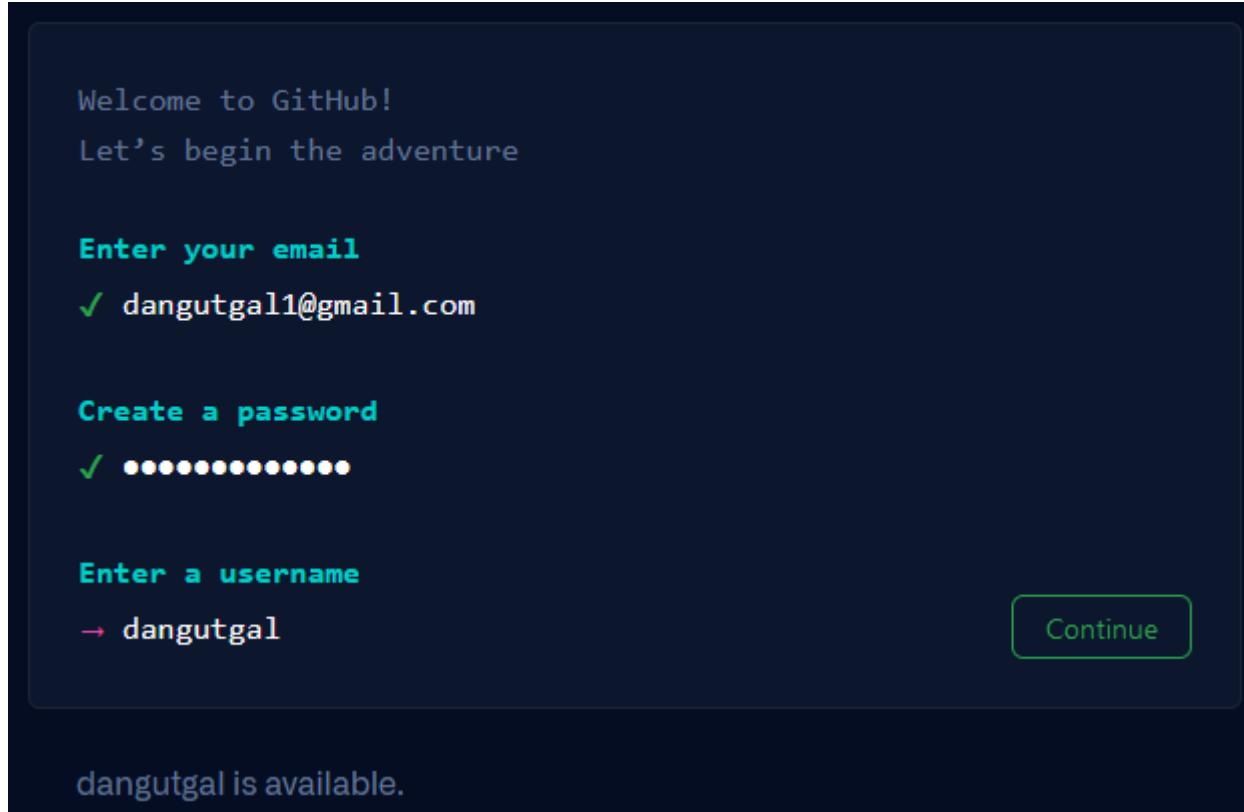
# ¿Cómo podemos crearnos una cuenta?

- Primer paso: introducir dirección de correo electrónico, contraseña y nombre de usuario
- **¡CUIDADO!**: Usa un correo que no caduque.



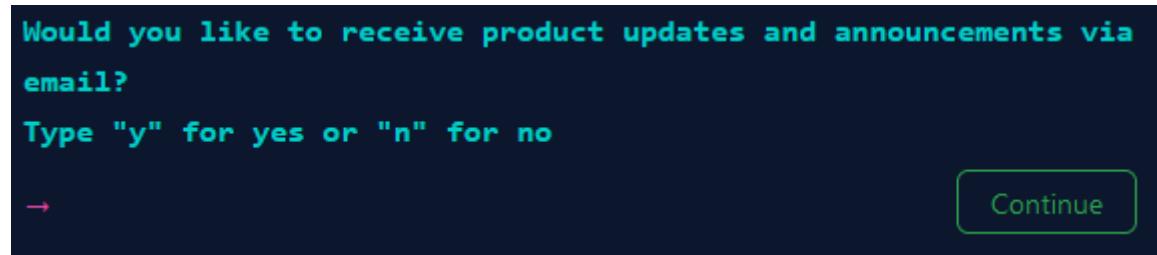
# ¿Cómo podemos crearnos una cuenta?

- Comprueba que el nombre de usuario está disponible.



# ¿Cómo podemos crearnos una cuenta?

- ¿Publicidad? Si estás interesado en probar nuevas funcionalidades, entonces sí.



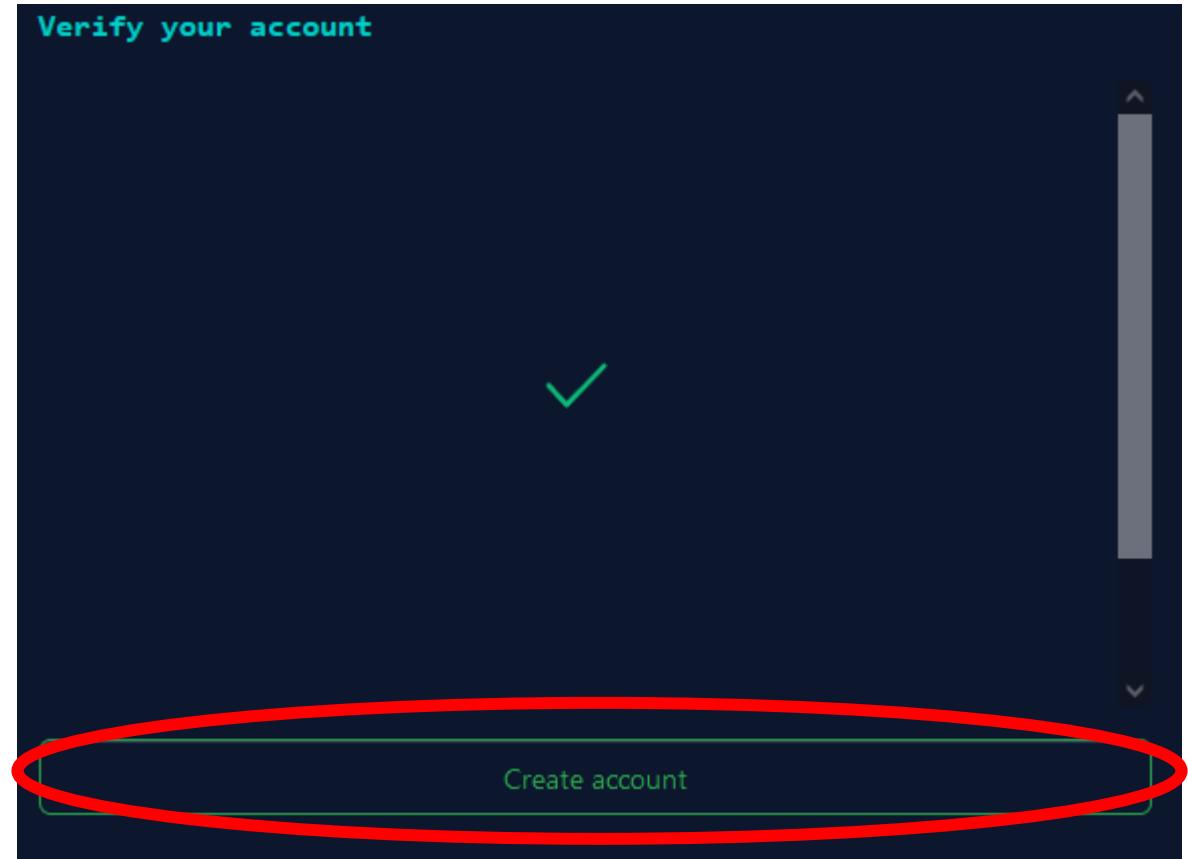
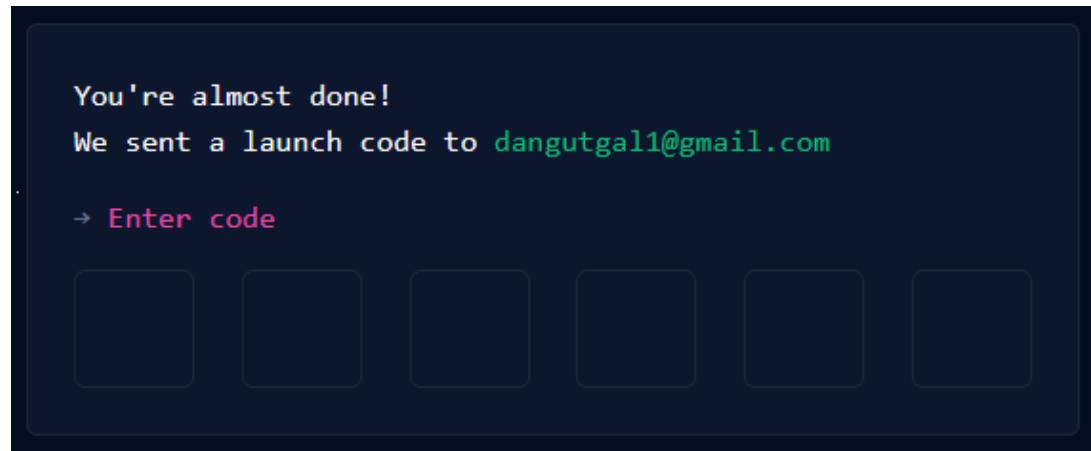
# ¿Cómo podemos crearnos una cuenta?

- Por último, hay que demostrar que no somos una máquina resolviendo un pequeño problemilla.



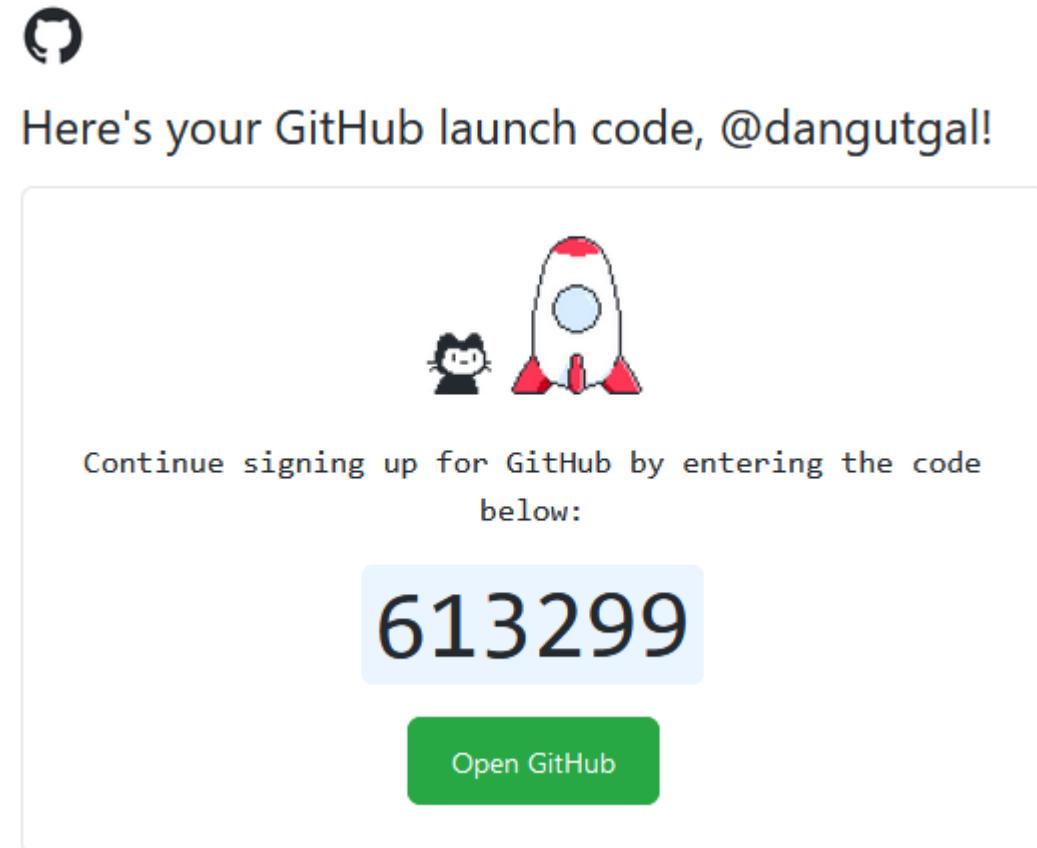
# ¿Cómo podemos crearnos una cuenta?

- Y ¡casi todo listo!
- Comprueba tu correo electrónico e introduce la clave.



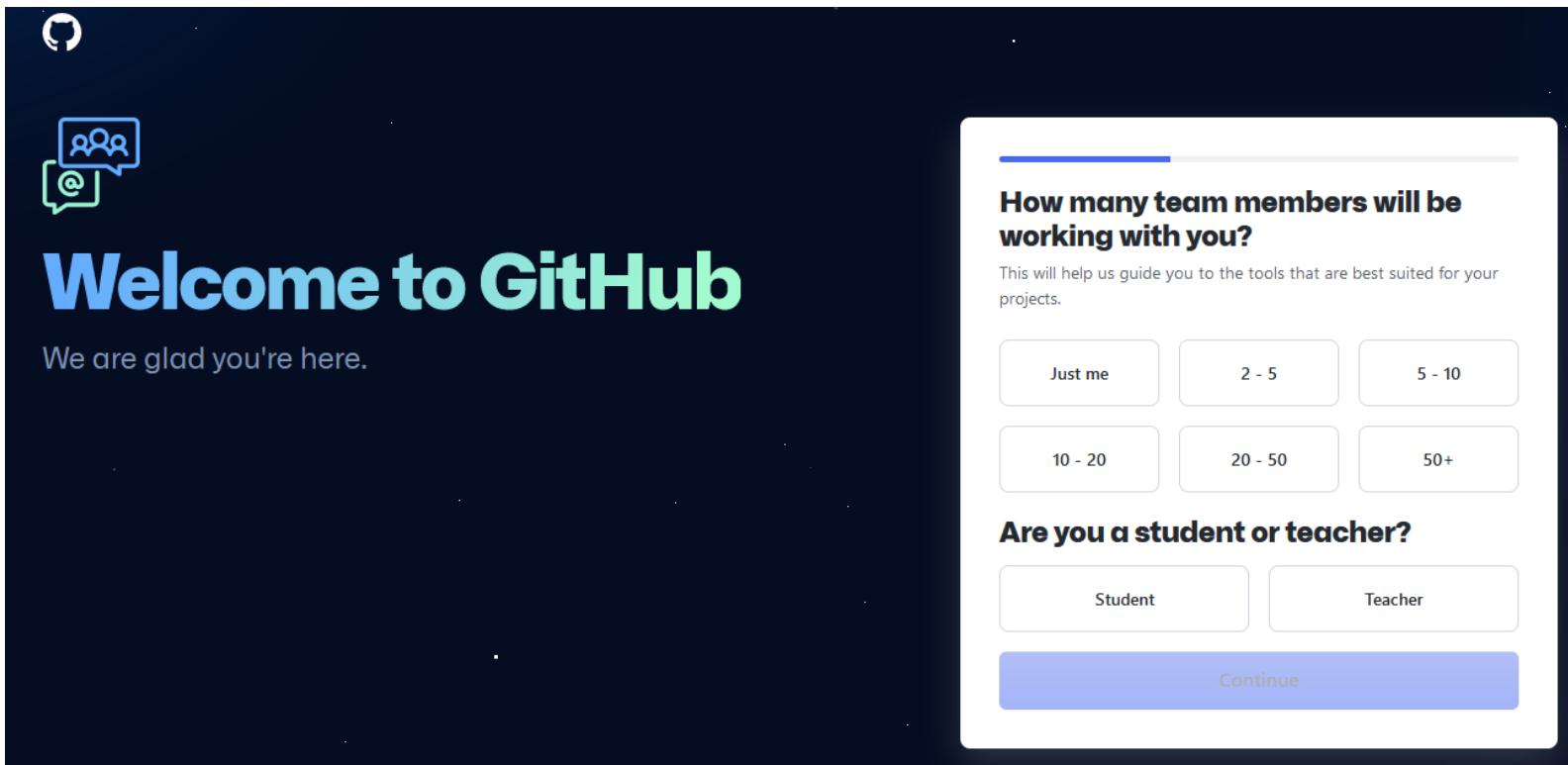
# ¿Cómo podemos crearnos una cuenta?

- Y ¡casi todo listo!
- Comprueba tu correo electrónico e introduce la clave.



# ¿Cómo podemos crearnos una cuenta?

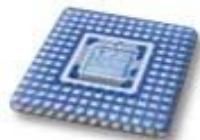
- Personalización de la cuenta:



# ¿Cómo podemos crearnos una cuenta?

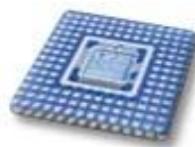
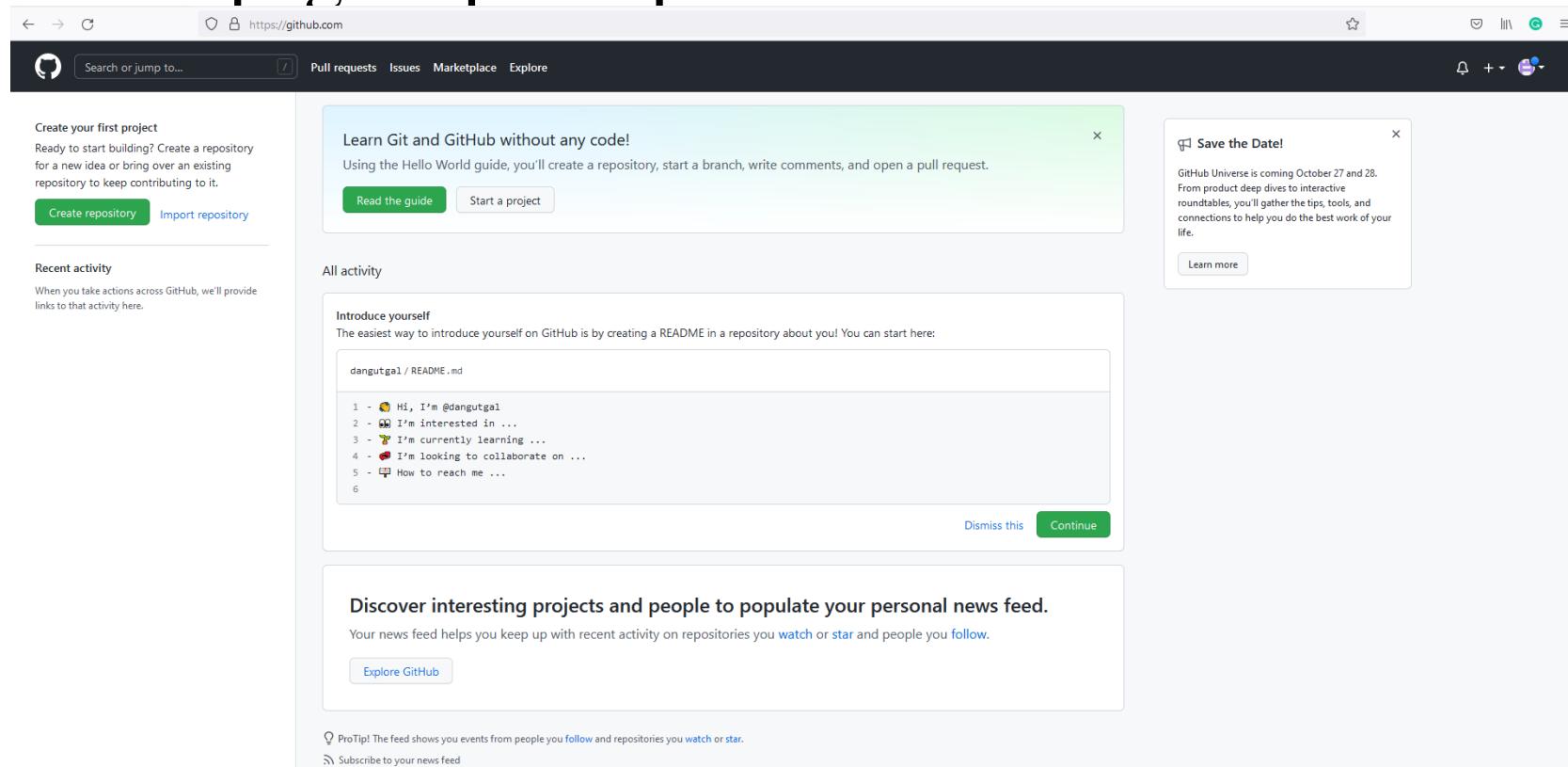
- Funcionalidades extra para docentes (necesario pedirlo):

The screenshot shows the GitHub teacher benefits page. At the top, there's a dark header with icons for a profile, a group, and a search bar. Below it is a blue banner with the text "Real-world tools, engaged students." and a subtext "GitHub gives teachers free access to industry-standard tools for training developers." On the left, a "Free" section lists: Unlimited public/private repositories, 2,000 Actions minutes/month (free for public repositories), 500MB of Packages storage (free for public repositories), and Community support. In the center, a "Get additional teacher benefits" section for "GitHub Team" includes: Protect your branches (ensuring collaborators can't make irreversible changes), Draft pull requests, Required reviewers, and 3,000 CI/CD minutes/month (free for public repositories). A large blue button at the bottom right says "Apply for your GitHub teacher benefits".



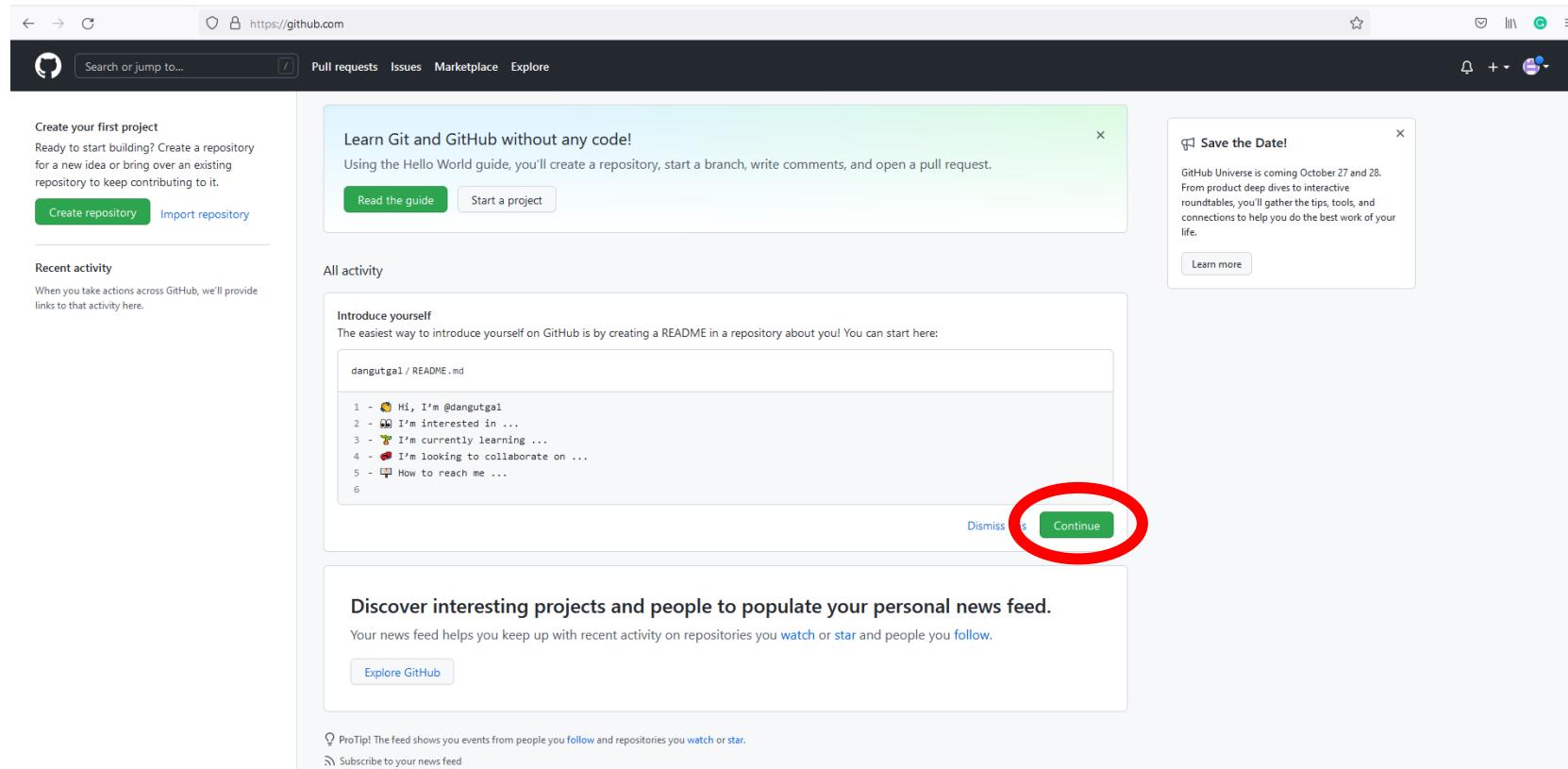
# ¿Cómo podemos crearnos una cuenta?

- Dashboard o página principal de tu cuenta:



# ¿Cómo podemos crearnos una cuenta?

- Empezamos configurando nuestra página de presentación.



# ¿Cómo podemos crearnos una cuenta?

- Es importante ser claro y conciso.

The screenshot shows a GitHub repository page for 'dangutgal/dangutgal'. The repository is public, as indicated by the 'Public' button. The main navigation bar includes 'Code', 'Pull requests', 'Actions', 'Projects', 'Security', 'Insights', and 'Settings'. The 'Code' tab is selected. In the center, there is a text editor for the 'README.md' file. The file content is as follows:

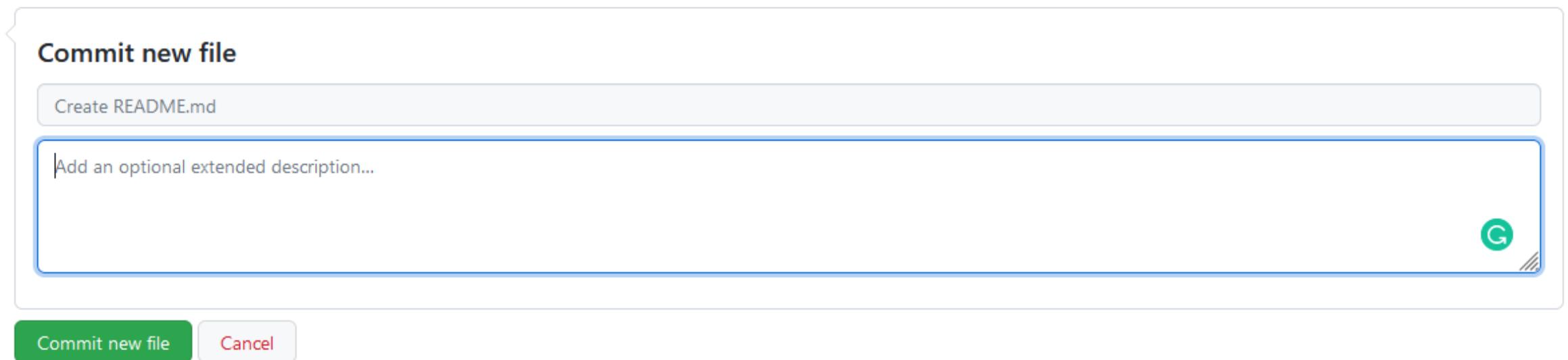
```
1 - 🎨 Hi, I'm @dangutgal
2 - 🧠 I'm interested in neuromorphic engineering and neurorobotic applications.
3 - 🎵 I'm currently learning about the auditory attention process for solving the cocktail party problem.
4 - 🍔 I'm looking to collaborate on digital design projects.
5 - 📩 How to reach me: dgutierrez5@us.es|
6
7 <!--
8 dangutgal/dangutgal is a 🌟 special 🌟 repository because its `README.md` (this file) appears on your GitHub profile.
9 You can click the Preview link to take a look at your changes.
10 -->
11
```

At the top of the editor, there are tabs for 'Edit new file' and 'Preview'. To the right, there are settings for 'Spaces' (set to 2), 'Wrap' (set to 'No wrap'), and a 'Cancel changes' button. A green GitHub icon is visible in the bottom right corner of the editor area.



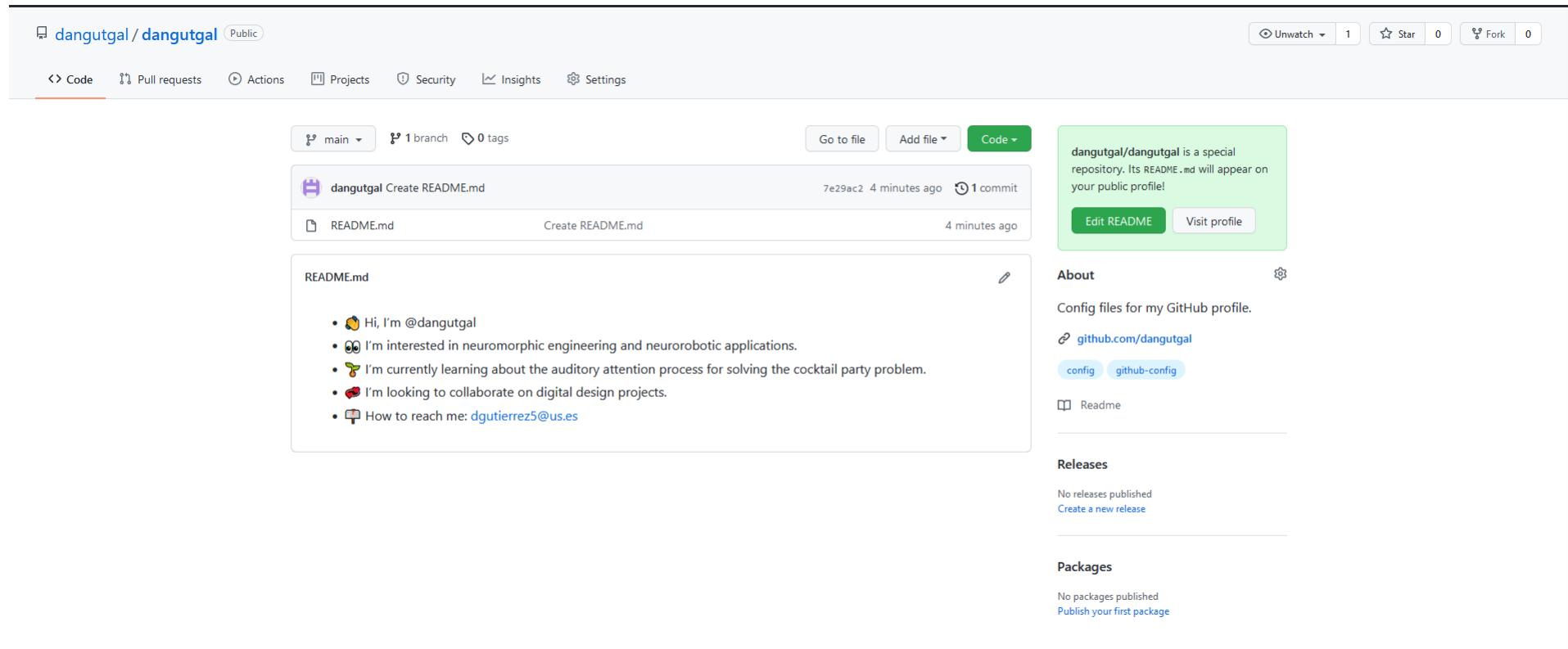
# ¿Cómo podemos crearnos una cuenta?

- ¡Hacemos nuestro primer commit!



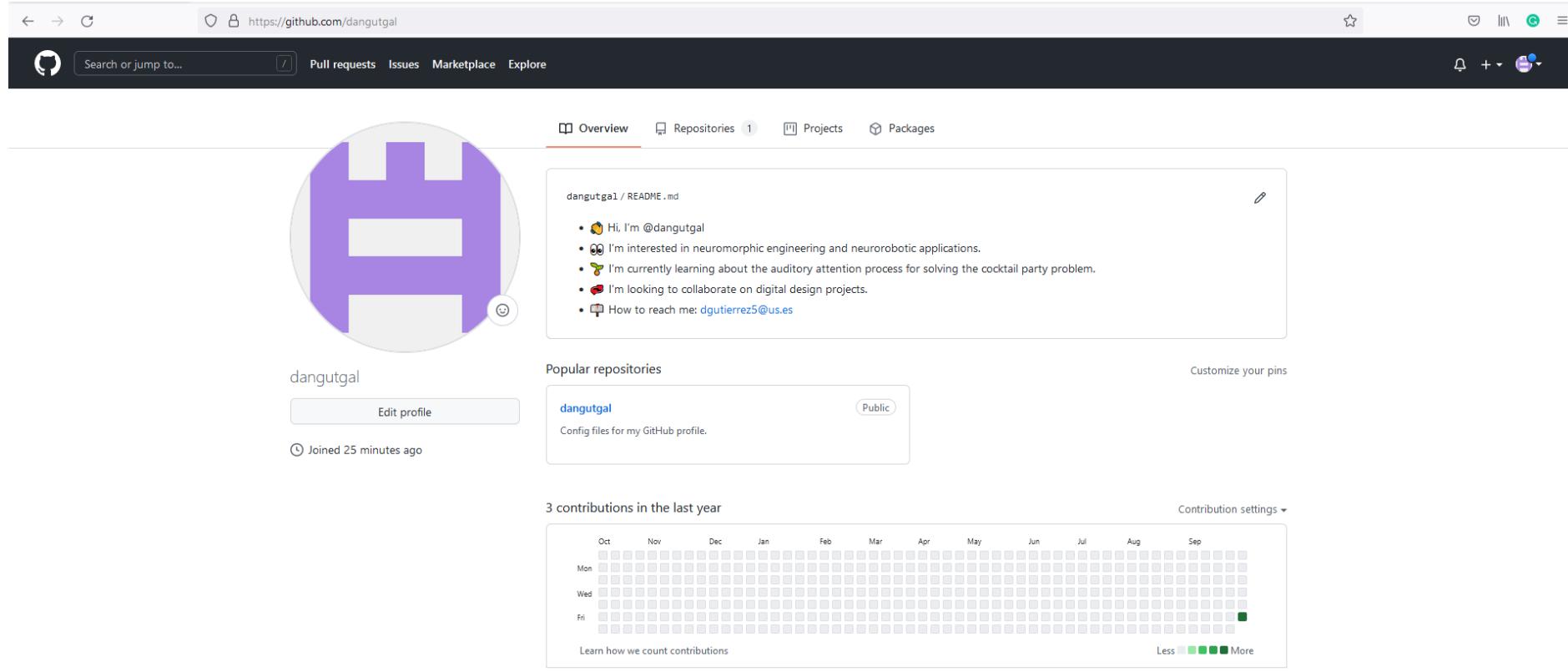
# ¿Cómo podemos crearnos una cuenta?

- Aspecto de nuestro repositorio personal (primer repositorio):



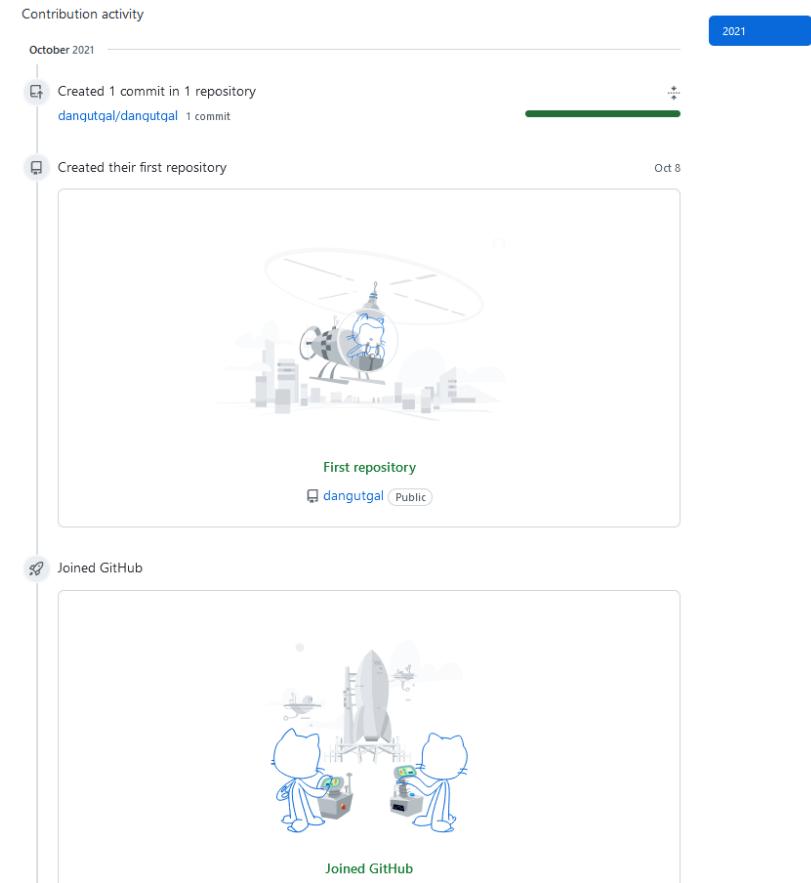
# ¿Cómo podemos crearnos una cuenta?

- Aspecto de nuestro perfil:



# ¿Cómo podemos crearnos una cuenta?

- Aspecto de nuestro perfil:
  - Resumen del historial de eventos en la página principal de nuestro perfil.
  - Ayuda a tener una visión global de los últimos cambios de forma rápida.



# ¿Qué me ofrece GitHub?

- Marketplace, Explore, notificaciones, gestión de proyectos...

**Extend GitHub**  
Find tools to improve your workflow

[Explore free apps](#)

**Types**

Apps  
Actions

**Categories**

API management  
Chat  
Code quality  
Code review  
Continuous integration  
Dependency management  
Deployment  
IDEs  
Learning  
Localization  
Mobile  
Monitoring

**Search** Search for apps and actions Sort: Best Match

CircleCI By circleci Recommended  
Automatically build, test, and deploy your project in minutes

CodeFactor By codefactor-io Recommended  
Automated code review for GitHub

codebeat By codequest-eu Recommended  
Code review expert on demand. Automated for mobile and web

gitpod.io By gitpod-io Always ready to code - spin up fresh, automated dev environments for each task, in the cloud, in seconds  
16.1k installs

Check Run Reporter By check-run-reporter See your test and style results without leaving GitHub. Works with any CI service. Supports JUnit, Checkstyle, and more  
540 installs

Django Doctor By higher-tier Improve the security, performance, and maintainability of your Django code  
1.3k installs

Layer CI By webappio A staging server replacement and CI/CD tool for full-stack apps  
949 installs

GitProtect.io Backup By xoperosoftware Fully manageable, most professional repository and metadata backup and recovery  
1k installs

Explore Topics Trending Collections Events GitHub Sponsors Get email updates

Here's what we found based on your interests...

Based on topics you've starred  
# astrochronology [Star](#)

There are 1 public repositories matching this topic

miguel-garces / Geol-Time-Scale  
Plot a Geological Time Scale (Time, Age, Stage, Geomagnetic Chrons, Polarity, and orbital parameters of eccentricity, obliquity and precession) for a given interval of the last 65Ma.  
[See more matching repositories](#)

Trending repositories today

raywen... / fita-materials 568 The projects and the materials that accompany the Flutter Apprentice book

facebook / flipper 9.3k A desktop debugging platform for mobile developers.

arkenfox / user.js 3.6k Firefox privacy, security and anti-fingerprinting: a comprehensive user.js template for configuration and hardening

Wo... / twentytwentytwo 91 Twenty Twenty-Two, the default WordPress theme that will launch with WordPress 5.9.

See more trending repositories >

Trending developers

Felix Angelov felangel felicitec  
Lee Robinson leerob leerob.io

Game Bytes

Beat Rocks

A stellar hybrid of Asteroids and a rhythm game made for this year's js13kGames

# ¿Qué me ofrece GitHub?

- Recomendación: en el Marketplace, seleccionad:
- Y realizad el curso de introducción:



A screenshot of the 'Introduction to GitHub' course. It shows the title 'Introduction to GitHub', the author 'The GitHub Training Team', a brief description about getting started with GitHub in less than an hour, and a list of topics: Git, GitHub Pages, Branches, Commits, and Pull Requests. At the bottom, it says 'Next: Create a branch', shows a progress bar at 3 of 8, and has 'Learn more' and 'Resume' buttons.

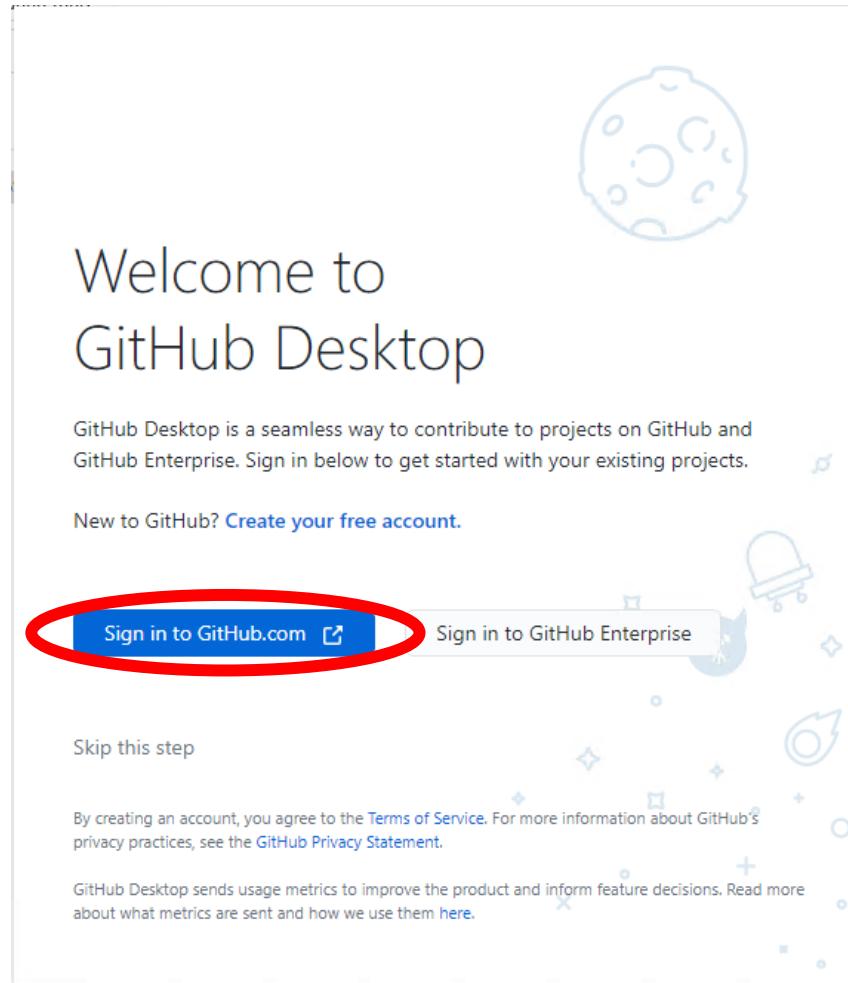
# Instalando GitHub Desktop

- Accede al siguiente link:
  - <https://desktop.github.com/>
- Por lo general, la web detecta el tipo de sistema operativo:



# Instalando GitHub Desktop

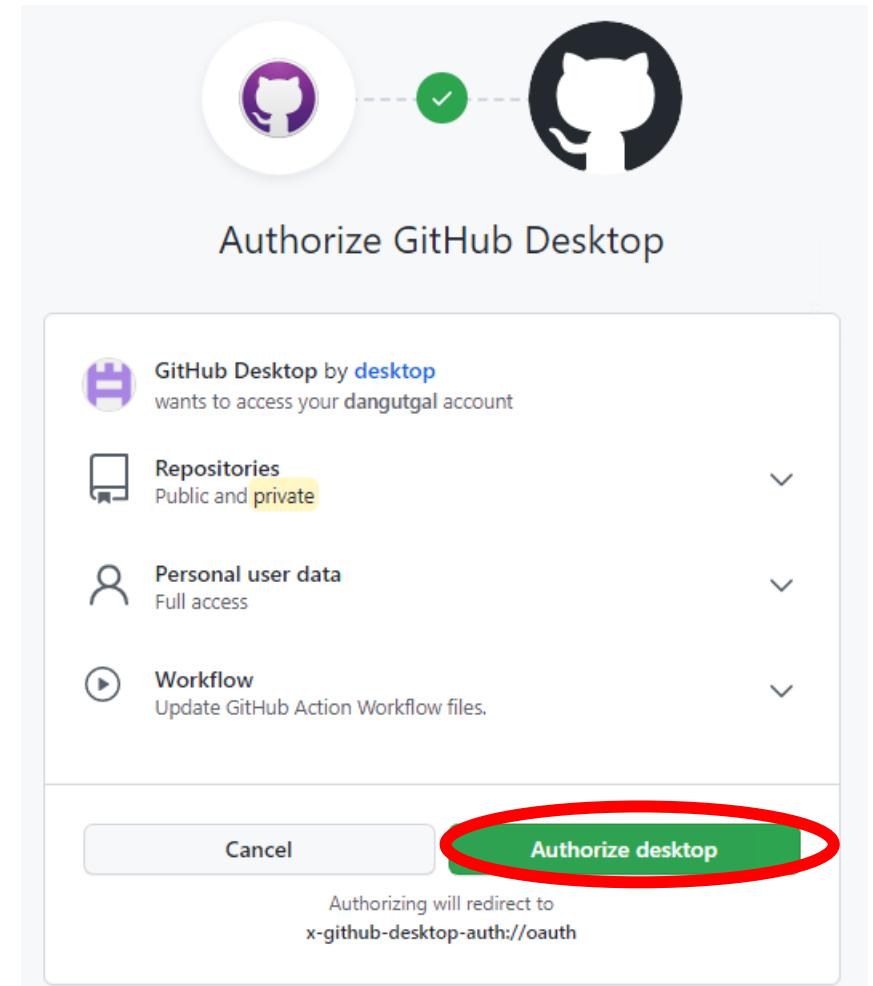
- Al lanzar el ejecutable:
  - Hacemos sign in con nuestra cuenta.
  - Te redirecciona a la web de GitHub, donde hay que usar nuestro correo y nuestra contraseña.



dgutierrez@atc.us.es

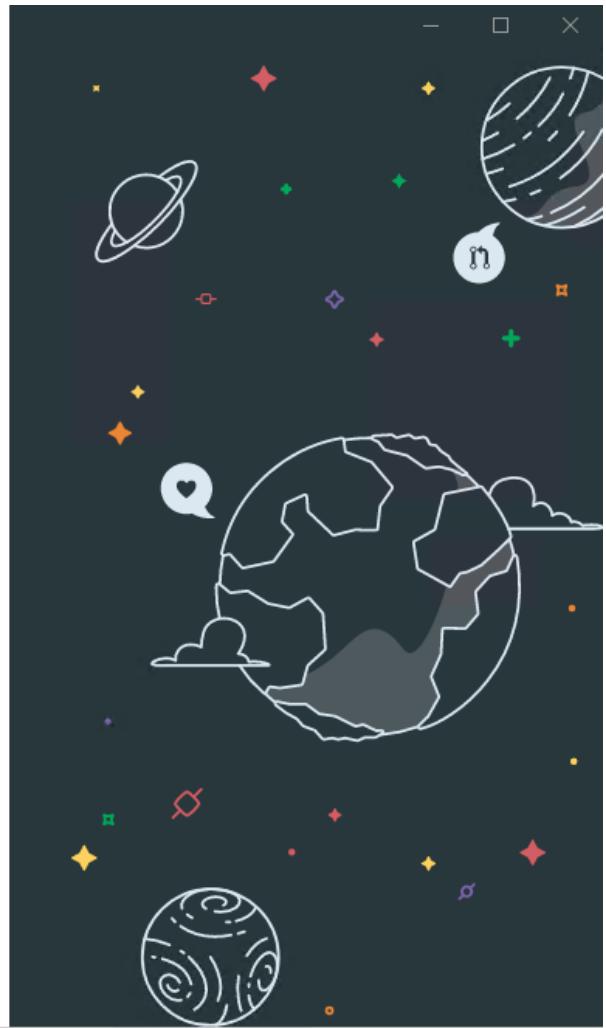
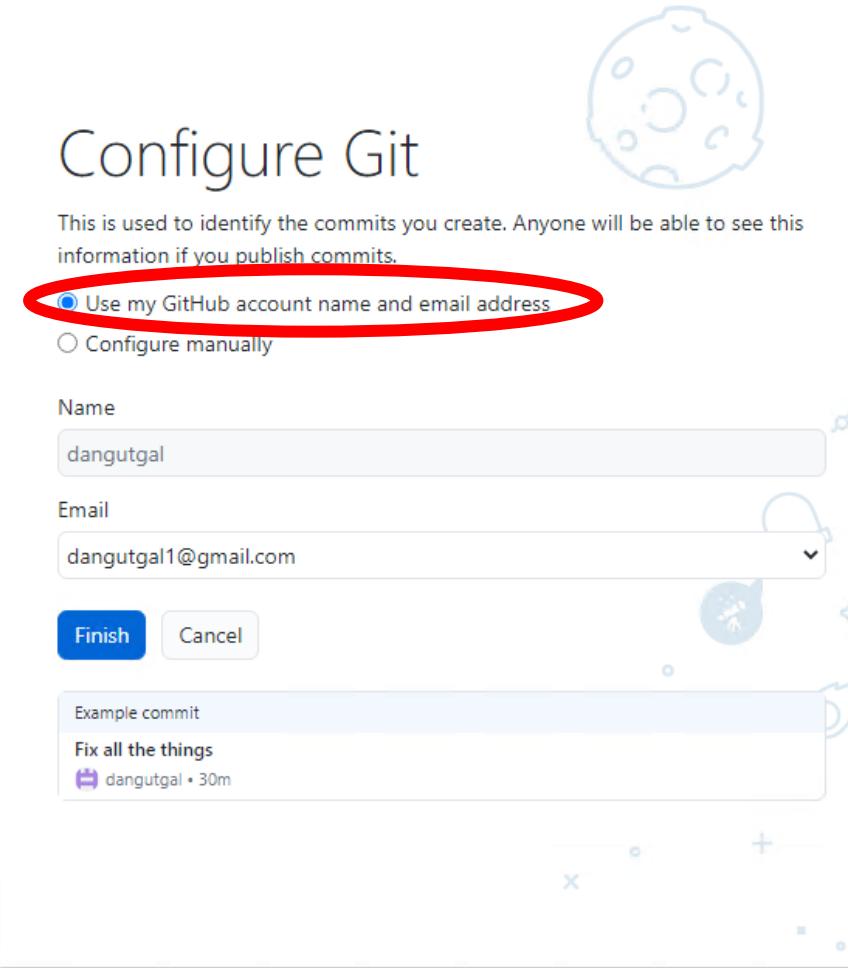
# Instalando GitHub Desktop

- Autorizamos a GitHub Desktop a tener acceso a nuestra cuenta de GitHub.



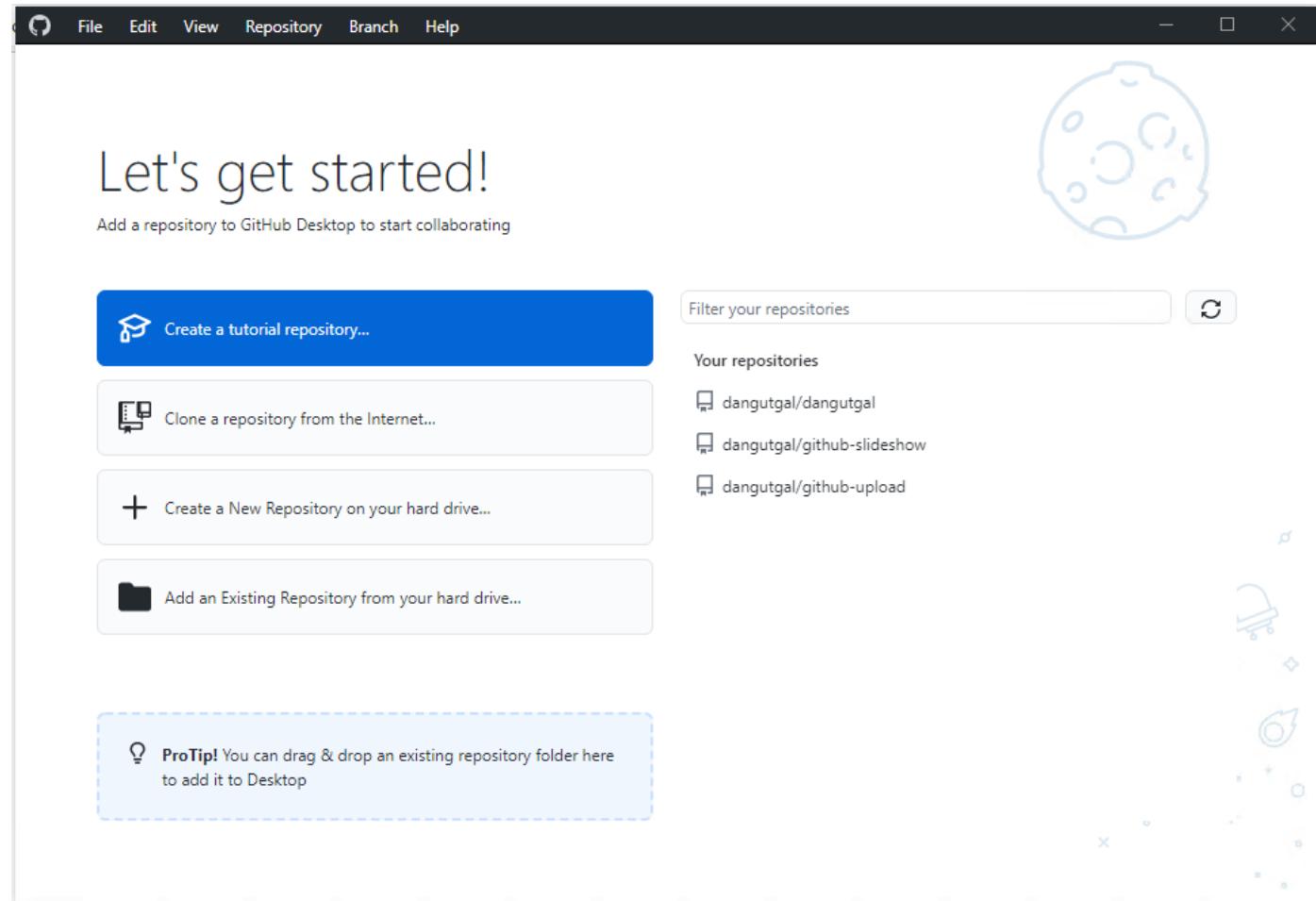
# Instalando GitHub Desktop

- Usamos nuestra información de la cuenta de GitHub para configurar Git en GitHub Desktop.



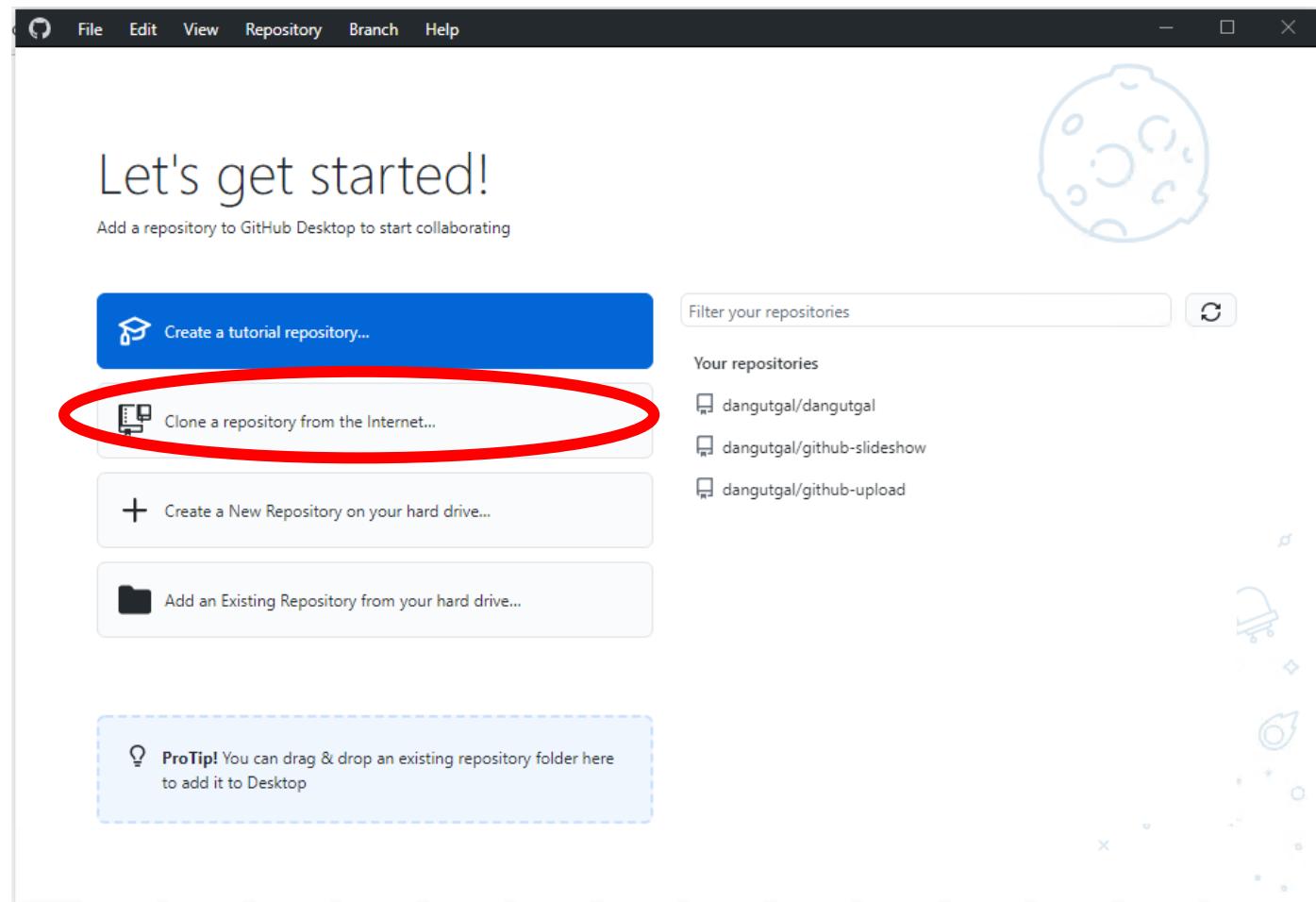
# Instalando GitHub Desktop

- ¡Instalación finalizada!  
Ahora puedes crear un nuevo repositorio, clonar uno existente de tu cuenta, o cargar un repositorio existente en tu disco duro.



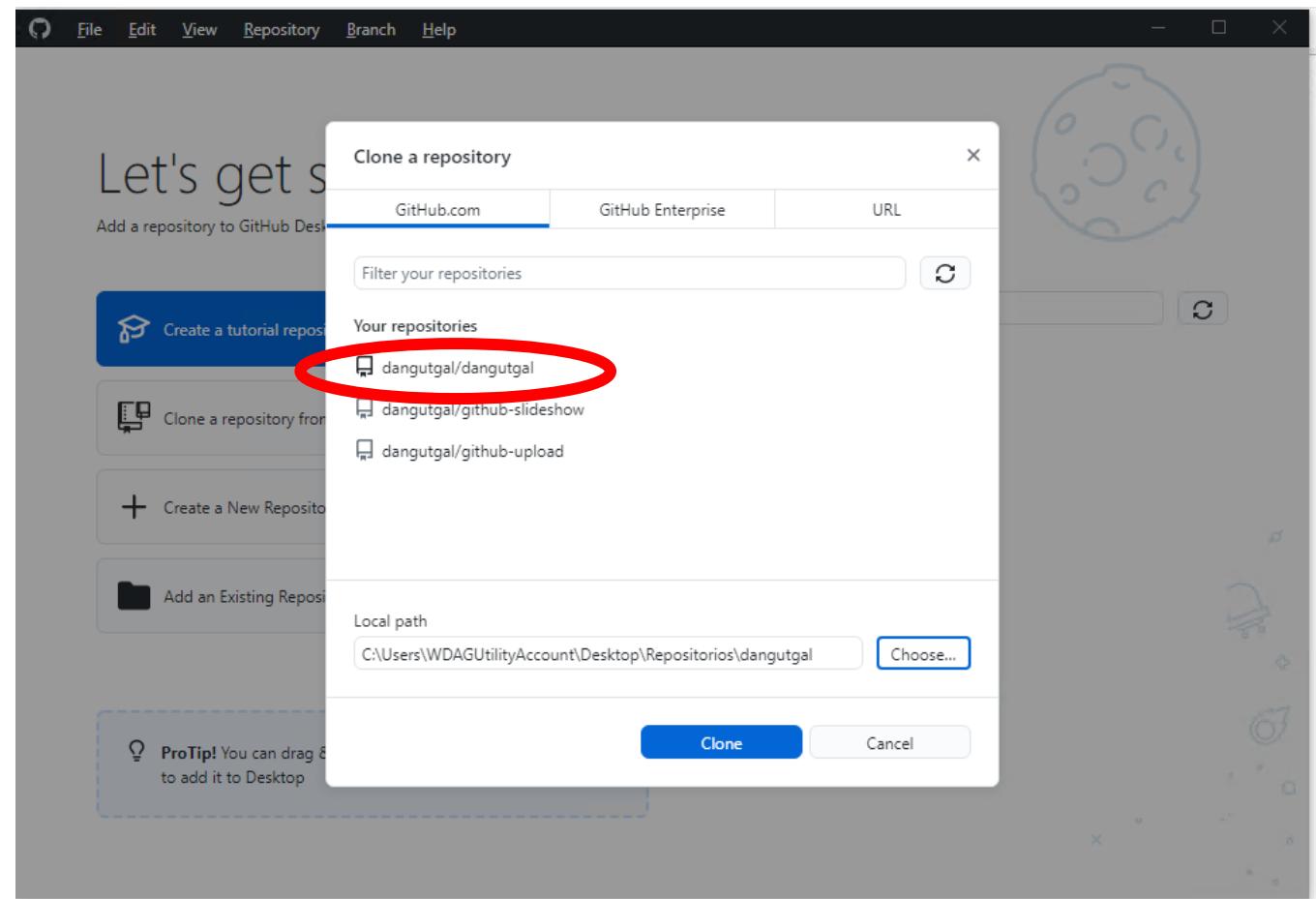
# Instalando GitHub Desktop

- Vamos a clonar nuestro primer repositorio.
- Seleccionamos “Clone a repository from the Internet”

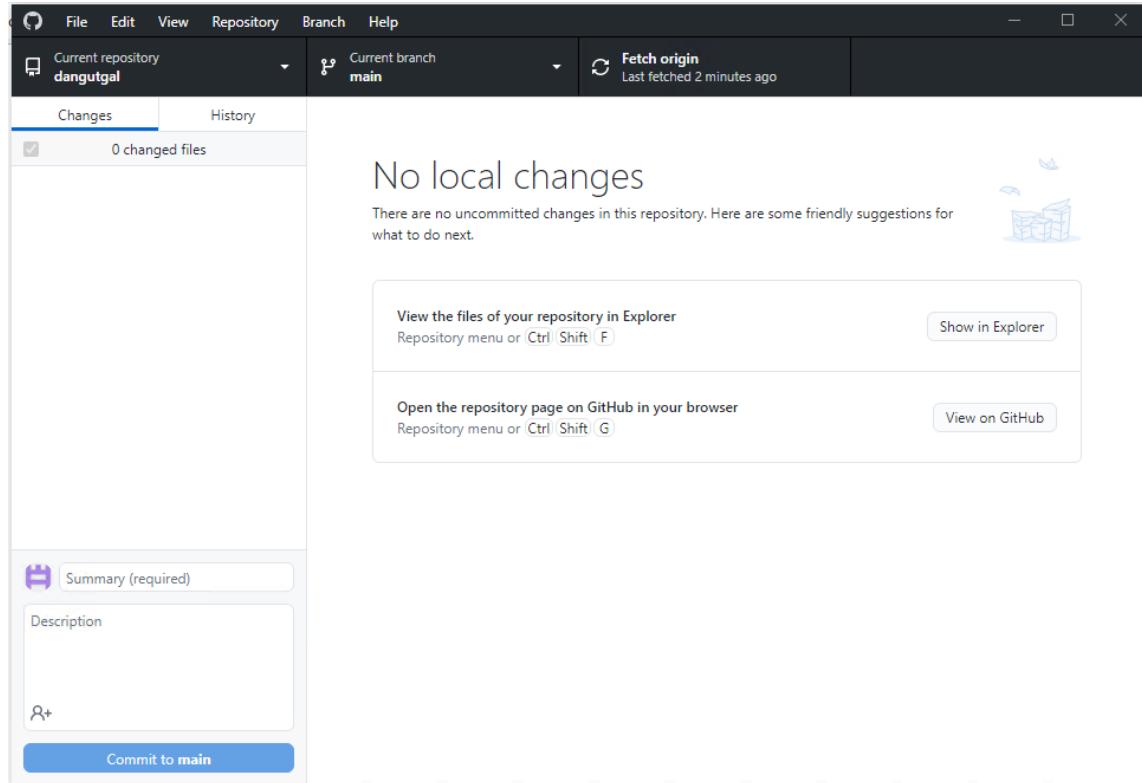


# Instalando GitHub Desktop

- Seleccionamos el repositorio deseado.
- Podemos seleccionar la carpeta de destino:
  - Consejo: agrupar todos los repositorios en una carpeta llamada “Repositorios”.



# Instalando GitHub Desktop



This screenshot shows the GitHub Desktop application interface with the same repository and branch setup as the first image. The 'History' tab is now selected, showing a commit for 'Create README.md' made by 'dangutgal' 5 days ago. The commit details show the content of the README.md file:

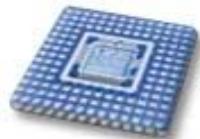
```
Hi, I'm @dangutgal
I'm interested in neuromorphic engineering and neurorobotic applications.
I'm currently learning about the auditory attention process for solving the cocktail party problem.
I'm looking to collaborate on digital design projects.
How to reach me: dgutierrez@us.es
+<!---
+dangergal/dangutgal is a 🌟 special 🌟 repository because its `README.md` (this file) appears on your GitHub profile.
+you can click the Preview link to take a look at your changes.
+--->
```

# Curioseando GitHub Desktop

- Se puede hacer prácticamente de todo desde la interfaz gráfica.
- También incluye la consola de Git integrada:
  - Para usarla, es necesario instalar Git:
    - [Set up Git - GitHub Docs](#)
- **Instalaremos Git más adelante.**

# Índice

- Objetivos.
- Introducción a Git.
- Introducción a GitHub.
- **Metodología de creación y gestión de repositorios.**
- Gestión de proyectos colaborativos.
- Integrando Git en herramientas de trabajo.

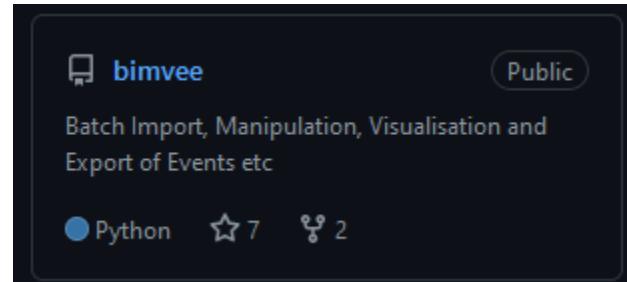


# Antes de empezar

- Hay que tener claro varios aspectos:
  - Organización del proyecto.
  - Lenguajes de programación a usar.
  - Número de colaboradores.
  - Plantillas.
  - Estilos de programación.
  - Documentación...
- **!ANTES-DE-EMPEZAR!**

# Nombre del repositorio

- Importancia del nombre:
  - Fácil de recordar.
  - Descriptivo.
  - No muy largo.
  - Sin letras raras.
  - Sin espacios.
  - Todo mayúsculas o todo minúsculas.

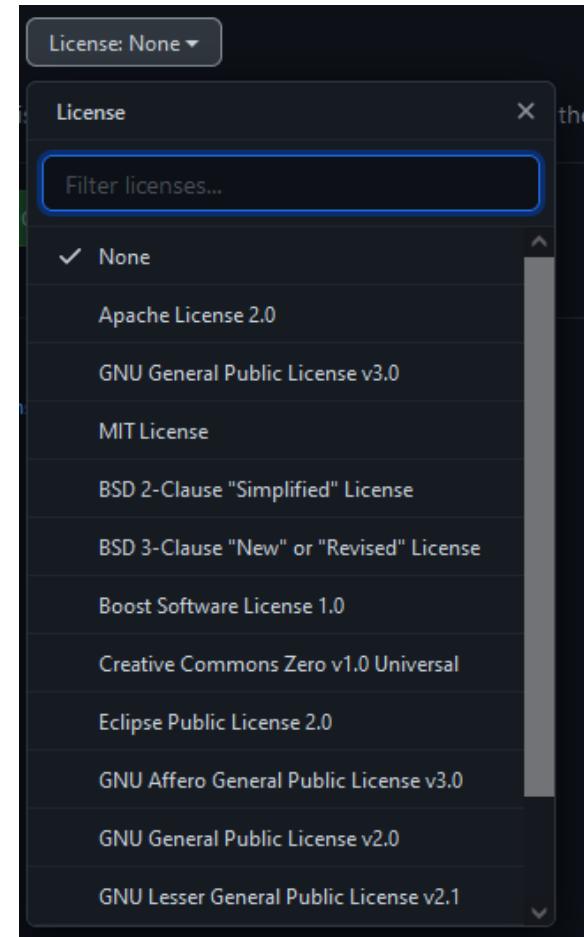


# ¿Privado o público?

- Privado:
  - Proyecto muy nuevo.
  - Todavía en desarrollo, y no funcional.
  - Relacionado con empresas (o con financiación externa).
- Público:
  - No importan los estados intermedios del proyecto.
  - Compartir tu código con la comunidad para ser usado.

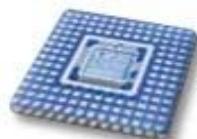
# ¿Qué tipo de licencia?

- Existen varios tipos tipos de licencia.
- Dependerá del proyecto:
  - Se permite su uso únicamente para fines no comerciales.
  - Se permite su uso también para fines comerciales.
  - Otros...
- Por lo general: **GNU GPL v3.0**



# ¿Es necesario el README.md?

- Depende:
  - Para usar Git sin interfaz gráfica → No tanto.
  - Para usar Git junto con GitHub → 100% necesario.
    - Un README principal para explicar el contenido y la organización del repositorio, y alguna indicación importante.
    - Un README en cada carpeta del proyecto explicando qué contiene (incluyendo fotos, vídeos, código, etc.)



# ¿Es necesario el README.md?

README.md

## pyNAVIS: an open-source cross-platform Neuromorphic Auditory VISealizer

License [GPL v3](#) pypi [v1.1.0](#) docs [passing](#)

pyNAVIS is an open-source cross-platform Python module for analyzing and processing spiking information obtained from neuromorphic auditory sensors. It is primarily focused to be used with a NAS, but can work with any other cochlea sensor.

### Installing

pyNAVIS is distributed on [PyPi](#) and can be installed with pip:

```
pip install pyNAVIS
```

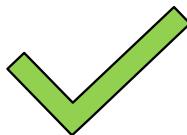
For more information, head over to the [ReadTheDocs Documentation](#).

### Getting started

Now that you have everything set up, you can create a new Python file and import the package:

```
from pyNAVIS import *
```

With this, you can now use all the functionalities that pyNAVIS provides. A more detailed guide with examples can be read [here](#).



README.md

# openNAS

### Description

OpenNAS is an open source VHDL-based Neuromorphic Auditory Sensor (NAS) code generator capable of automatically generating the necessary files to create a VHDL project for FPGA. OpenNAS guides designers with a friendly interface and allows NAS specification using a five-steps wizard for later code generation. It includes several audio input interfaces (AC'97 audio codec, I2S ADC and PDM microphones), different processing architectures (cascade and parallel), and a set of neuromorphic output interfaces (parallel AER, Spinnaker). After NAS generation, designers have everything ready for building and synthesizing the VHDL project for a target FPGA using manufacturer's tools.

### Table of contents

- Description
- Getting started
- Usage
- Post-processing
- Contributing
- Credits
- License
- Cite this work

# ¿Es necesario el README.md?

AERNode

No commit message

3 years ago

LICENSE

Initial commit

4 years ago

README.md

Create README.md

4 years ago

README.md

AERtools

User manuals and resources to use AERtools developed by RTC-Sevilla

**X**

# Creación de Wikis

- Parte fundamental para documentar el proyecto de forma interactiva.
- Muy recomendado si se quiere compartir el proyecto con otros centros de investigación.
- Debe contener toda la información que no se ha añadido al README.
- También puede incluir enlaces hacia otras plataformas de generación automática de documentación.

# Creación de Wikis

Home

Angel Jimenez edited this page on 11 Feb · 11 revisions

Edit New Page

## Welcome to the OpenNAS wiki!



### What is OpenNAS?

OpenNAS is an open-source VHDL-based Neuromorphic Auditory Sensor (NAS) code generator capable of automatically generating the necessary files to create a VHDL project for FPGA. OpenNAS guides designers with a friendly interface and allows NAS specification using a five-step wizard for later code generation. It includes several audio input interfaces (AC'97 audio codec, I2S ADC and PDM microphones), different processing architectures (cascade and parallel filter banks), and a set of neuromorphic output interfaces (parallel AER and Spinnaker). After NAS generation, designers have everything prepared for building and synthesizing the VHDL project for a target FPGA using the manufacturer's tools.

Pages 13

Wiki

- Home
- Glossary

Getting Started on Windows

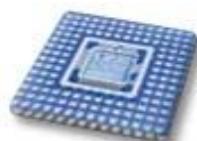
- Installing software dependencies
- Download the latest version
- FAQs

Running OpenNAS

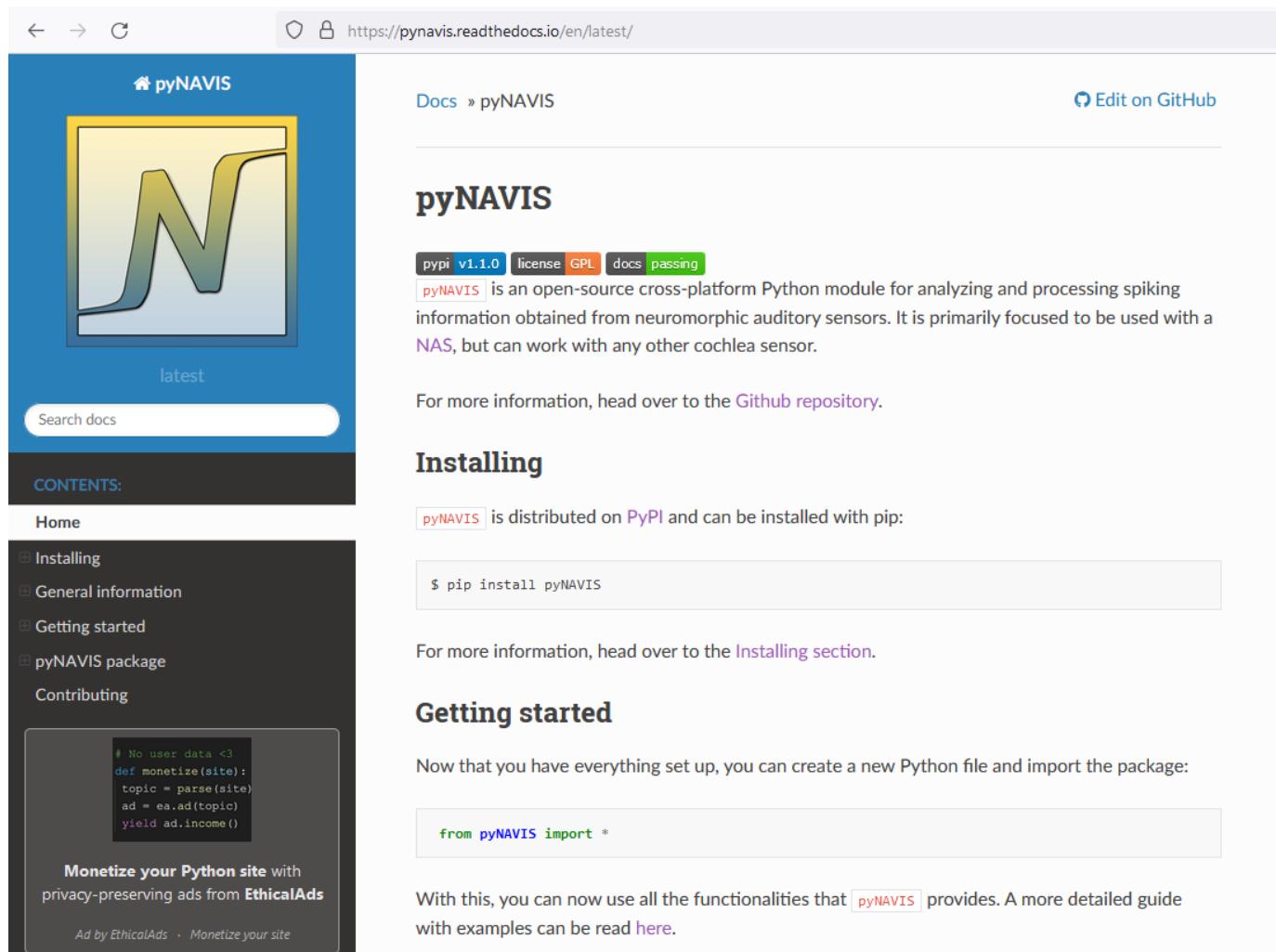
- OpenNAS wizard quick view
- Step 1: NAS commons settings
- Step 2: NAS audio input source
- Step 3: NAS processing architecture
- Step 4: NAS neuromorphic output interface
- Step 5: NAS destination folder

What's next?

- Files generated by OpenNAS
- Using NAS in an FPGA



# Creación de Wikis



The screenshot shows a web browser displaying the documentation for the `pyNAVIS` Python module. The URL in the address bar is `https://pynavis.readthedocs.io/en/latest/`. The page features a header with the `pyNAVIS` logo and navigation links for 'Docs' and 'Edit on GitHub'. Below the header, there's a large section titled 'pyNAVIS' with a brief description, a 'pypi v1.1.0' badge, and a link to the 'Github repository'. A 'CONTENTS:' sidebar on the left lists sections like 'Home', 'Installing', 'General information', 'Getting started', 'pyNAVIS package', and 'Contributing'. A snippet of Python code for monetizing a site is shown in a box. At the bottom, there's an advertisement for EthicalAds.

`pyNAVIS` is an open-source cross-platform Python module for analyzing and processing spiking information obtained from neuromorphic auditory sensors. It is primarily focused to be used with a `NAS`, but can work with any other cochlea sensor.

For more information, head over to the [Github repository](#).

## Installing

`pyNAVIS` is distributed on [PyPI](#) and can be installed with pip:

```
$ pip install pyNAVIS
```

For more information, head over to the [Installing](#) section.

## Getting started

Now that you have everything set up, you can create a new Python file and import the package:

```
from pyNAVIS import *
```

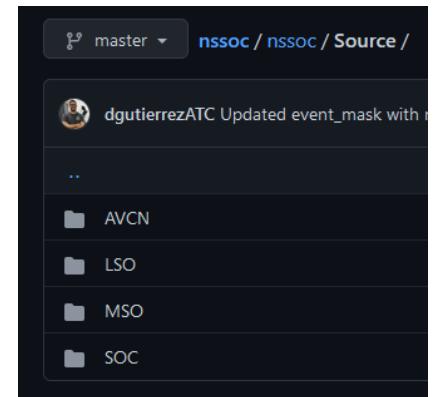
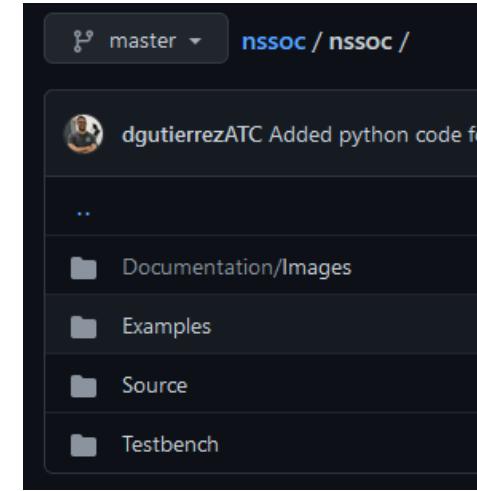
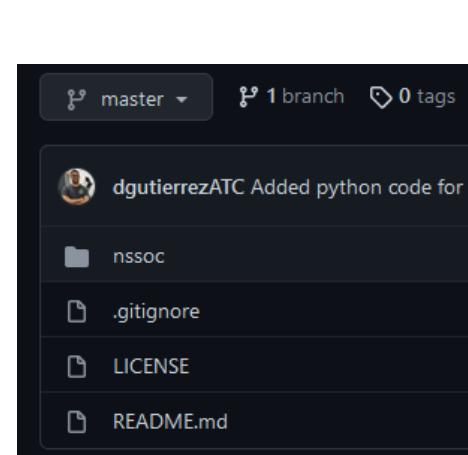
With this, you can now use all the functionalities that `pyNAVIS` provides. A more detailed guide with examples can be read [here](#).

# Organización de carpetas dentro del repositorio

- Depende de la naturaleza del proyecto.
- Distinguir siempre entre código fuente, código de test, documentación, etc.
- Pensar desde el principio en la organización de los directorios del proyecto:
  - Un cambio a posteriori podría afectar a rutas relativas o a confusión por parte de los usuarios.

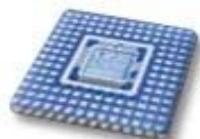
# Organización de carpetas dentro del repositorio

- Recomendación:
  - Repositorio en GitHub:
    - Nombre\_proyecto
      - Source
      - Test
      - Examples
      - Docs
      - Manual
    - .gitignore
    - LICENSE
    - README.md



# Descripción de los commits: específicos vs. genéricos

- El aspecto más importante de Git.
- Vital para llevar una buena trazabilidad del código.
- Descripción sencilla pero concreta.

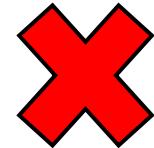


# Descripción de los commits: específicos vs. genéricos

- Si el cambio es grande (restructuración o cambio de plataforma)  
→ commit genérico.
- Si el cambio es pequeño pero crítico (cambio de una función)  
→ commit específico.
- Más detalles: en la descripción del commit, no en el título.

# Descripción de los commits: específicos vs. genéricos

📄 _init_.py	initial commit from archive
📄 allconv_decolle.py	updated to dcll_private:public HEAD
📄 base_model.py	updated to dcll_private:public HEAD
📄 lenet_decolle_model.py	updated to dcll_private:public HEAD
📄 lenet_delle.py	updated to dcll_private:public HEAD
📄 utils.py	updated to dcll_private:public HEAD



# Descripción de los commits: específicos vs. genéricos

■ HPU_lib	Interception and sensor selection
■ IP/Vivado	Adds port sensor Type
■ common_lib	Major change V3.0 adds SpiNNlink capabilities
■ datapath_lib	Adds port sensor Type
■ neuelab_lib	Rearm Command
■ neuserial.lib	Updated to 3.6 Rev1 anw working in ZCB
■ spinn_neu_if.lib	REmove old START/STOP Forcing method
□ ddr_sampler.v	Upgraded HPUCore module to the 2.2 version
□ fifo_lfsr.v	Upgraded HPUCore module to the 2.2 version
□ hssaer_paer_rx_wrapper.v	Bug fixed following the coding coming from AERsensorsMap.xlsx (svn ve...



# Descripción de los commits: específicos vs. genéricos

The image shows two screenshots of a GitHub commit history. The left screenshot displays a list of commits, while the right screenshot shows a detailed view of a specific commit.

**Left Screenshot (Commit History):**

- Update eventcodecs.md
- Revert "Update eventcodecs.md"
- Created codec\_CochleaEvent
- Updated vCodec.h with CochleaCodec
- Added CochleaEvent
- Update eventcodecs.md
- Bug in class CochleaEvent fixed** (This commit is circled in red)
- added cochlea event
- Update decode\_events with decode\_earEvents

**Right Screenshot (Detailed View of the Bug Fix Commit):**

**Bug in class CochleaEvent fixed**

dgtierrezATC authored and arrenglover committed on 15 Mar

```
@@ -198,7 +198,7 @@ class SkinSample : public SkinEvent
198     };
199
200 - class CochleaEvent : public CochleaEvent
201 {
202     public:
203         static const std::string tag;
204
@@ -235,7 +235,7 @@ class CochleaEvent : public CochleaEvent
235     virtual std::string getType() const;
236     virtual int getChannel() const;
237     virtual void setChannel(const int channel);
238 + }
239
240     /// \brief an AddressEvent with a velocity in visual space
241     class FlowEvent : public AddressEvent
```

# Gestión de issues

- Depende del proyecto:
  - Con plantilla:
    - Más organizado.
    - Asegura repetibilidad.
    - Pero más complicado de llenar.
  - Sin plantilla:
    - Menos profesional.
    - Suficiente para desarrollos pequeños.

# Gestión de issues

Filters ▾  Labels 9 Milestones 0 New issue

Clear current search query, filters, and sorts

	Author ▾	Label ▾	Projects ▾	Milestones ▾	Assignee ▾	Sort ▾
■ ① 3 Open ✓ 9 Closed						
■ ⓘ Function "Functions.PDF_report" does not close any Figure <small>enhancement</small>	dgutierrezATC					1
#11 by dgutierrezATC was closed on 16 Jun						
■ ⓘ Averga activity plot does not return any value <small>enhancement</small>	dgutierrezATC					1
#10 by dgutierrezATC was closed on 16 Jun						
■ ⓘ Time unit not indicated in Functions.mono_to_stereo(..., delay, ...)	dgutierrezATC	documentation				2
#7 by dgutierrezATC was closed on 3 Feb						
■ ⓘ Add another loader function for supporting events from a NAS with the SOC model <small>enhancement</small>	dgutierrezATC					1
#6 by dgutierrezATC was closed on 23 Feb						
■ ⓘ Implementation of a function to calculate the Interspike Interval <small>enhancement</small>	dgutierrezATC					1
#5 by dgutierrezATC was closed on 16 Jun						
■ ⓘ Include a parameter in loadCSV() to specify the separator <small>enhancement</small>	jdominguez					1
#4 by jdominguez was closed on 25 Jan						
■ ⓘ Add more information to the loadCSV function <small>documentation</small>	jdominguez					1
#3 by jdominguez was closed on 25 Jan						
■ ⓘ Use data.log file format as allowed input file <small>enhancement</small>	dgutierrezATC					1
#2 by dgutierrezATC was closed on 9 Nov 2020						
■ ⓘ AEDAT splitter's parameters <small>enhancement good first issue</small>	dgutierrezATC					1
#1 by dgutierrezATC was closed on 9 Nov 2020						



# Gestión de issues

38 lines (30 sloc) | 834 Bytes

name	about	title	labels	assignees
Bug report	Create a report to help us improve			

**Describe the bug** A clear and concise description of what the bug is.

**To Reproduce** Steps to reproduce the behavior:

1. Go to '...'
2. Click on '....'
3. Scroll down to '....'
4. See error

**Expected behavior** A clear and concise description of what you expected to happen.

**Screenshots** If applicable, add screenshots to help explain your problem.

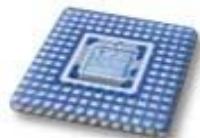
**Desktop (please complete the following information):**

- OS: [e.g. iOS]
- Browser [e.g. chrome, safari]
- Version [e.g. 22]

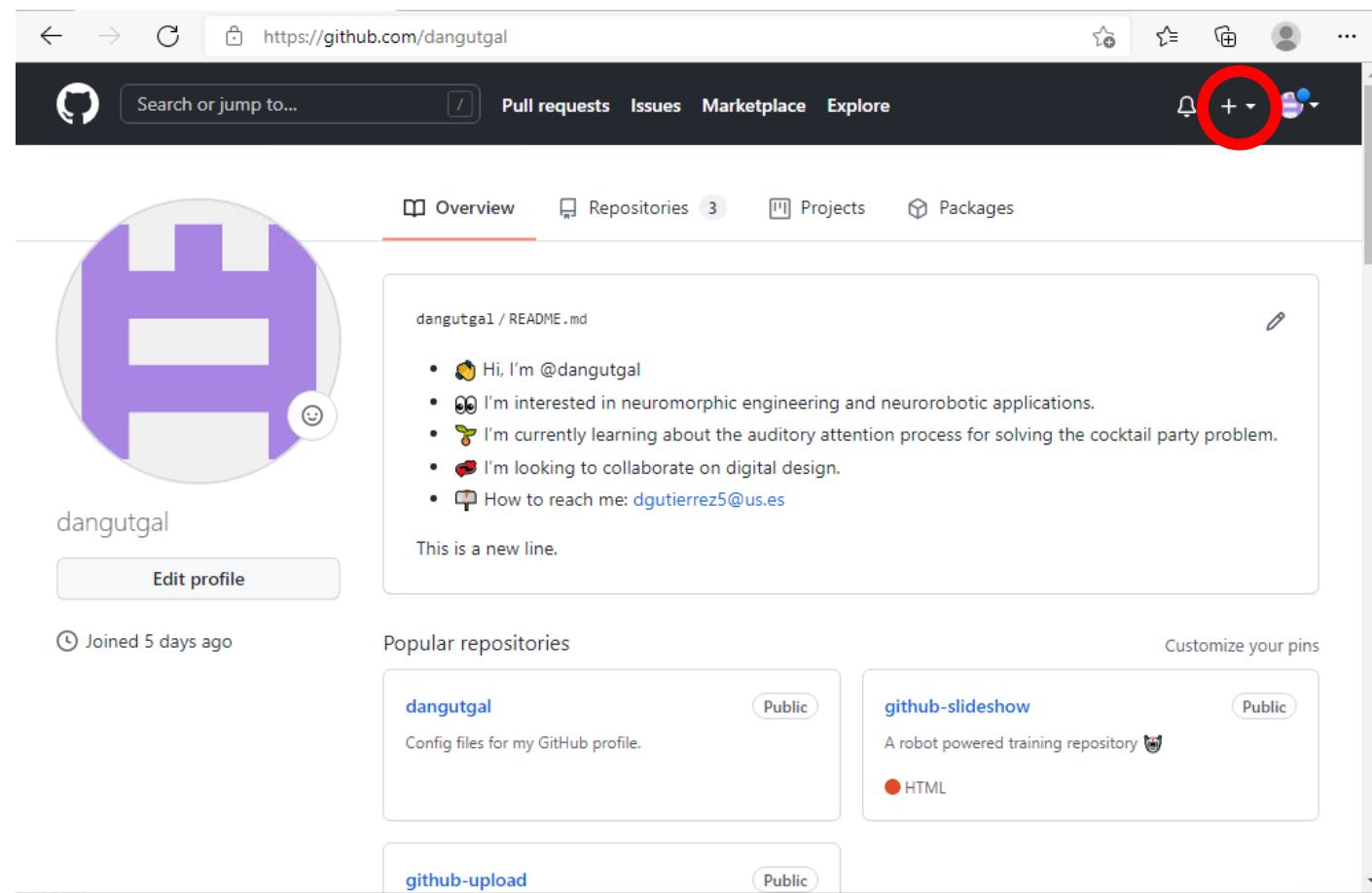
**Smartphone (please complete the following information):**

- Device: [e.g. iPhone6]
- OS: [e.g. iOS8.1]
- Browser [e.g. stock browser, safari]
- Version [e.g. 22]

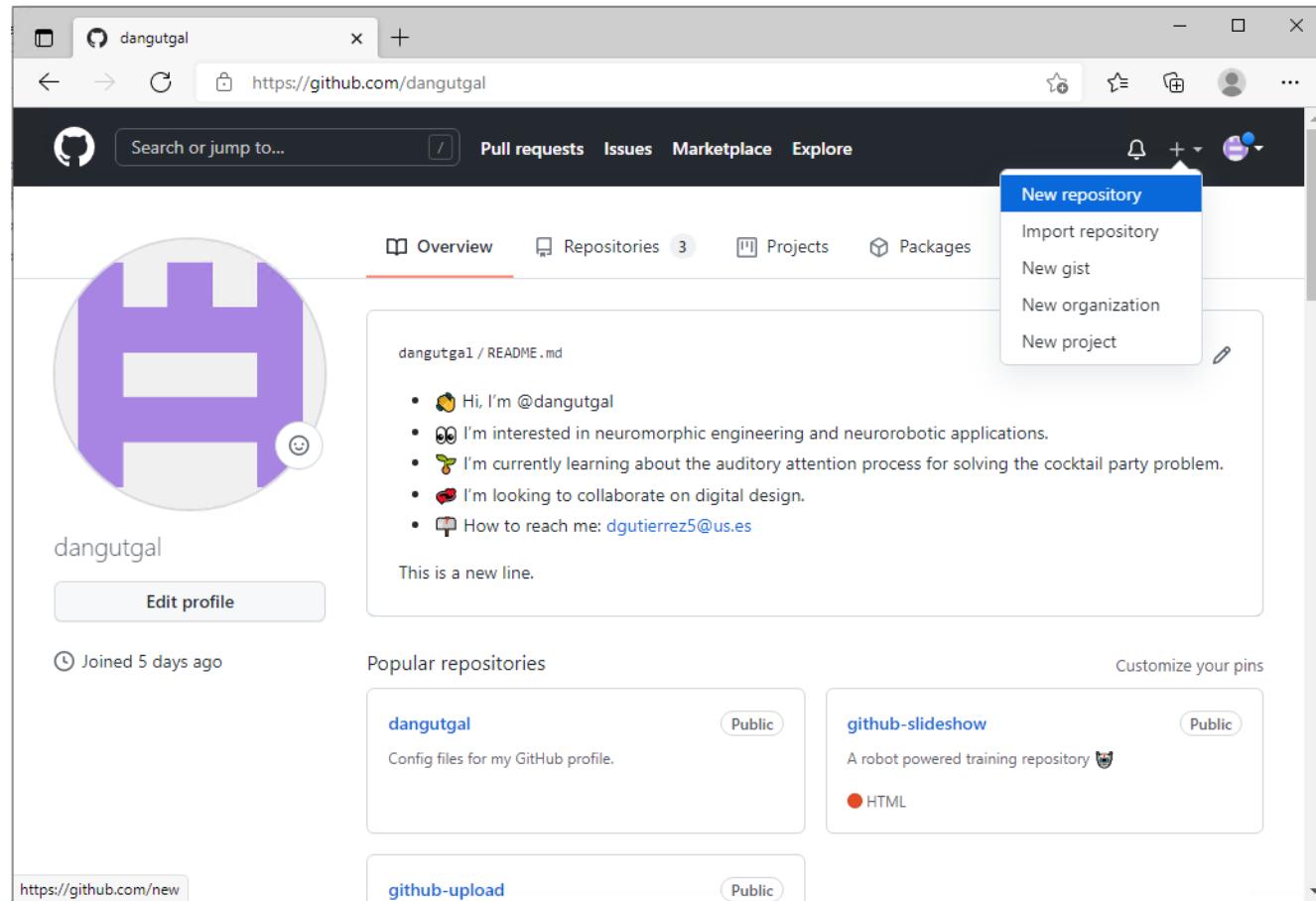
**Additional context** Add any other context about the problem here.



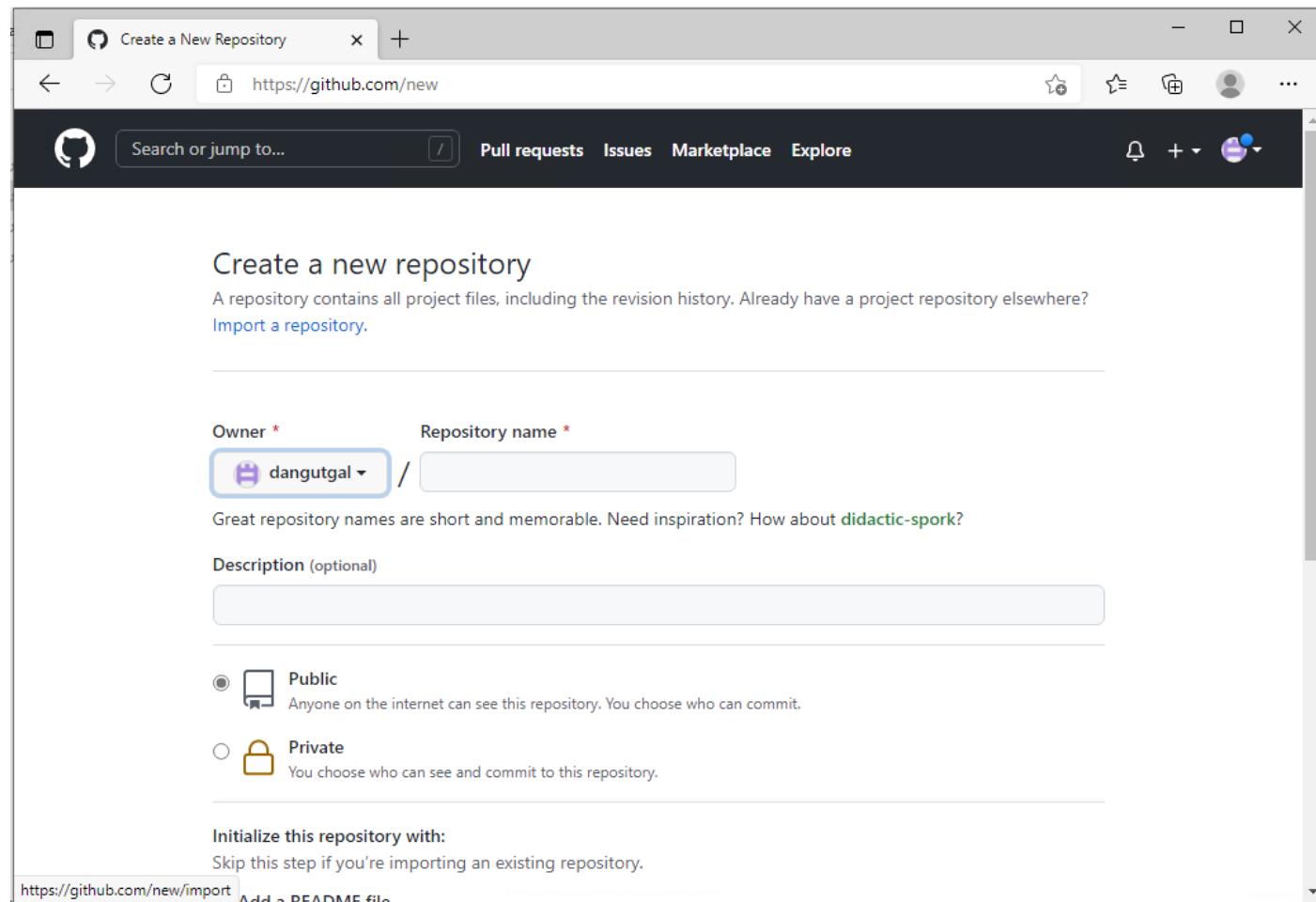
# Ejemplo de creación de repositorio



# Ejemplo de creación de repositorio



# Ejemplo de creación de repositorio



# Ejemplo de creación de repositorio

https://github.com/new

dangutgal / test\_ICE ✓

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

Description (optional)  
Repositorio de prueba para el curso del ICE

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.](#)  
.gitignore template: [VisualStudio](#) ▾

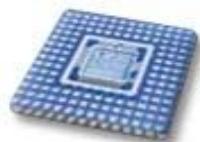
Choose a license  
A license tells others what they can and can't do with your code. [Learn more.](#)  
License: [GNU General Public ...](#) ▾

This will set [main](#) as the default branch. Change the default name in your [settings](#).

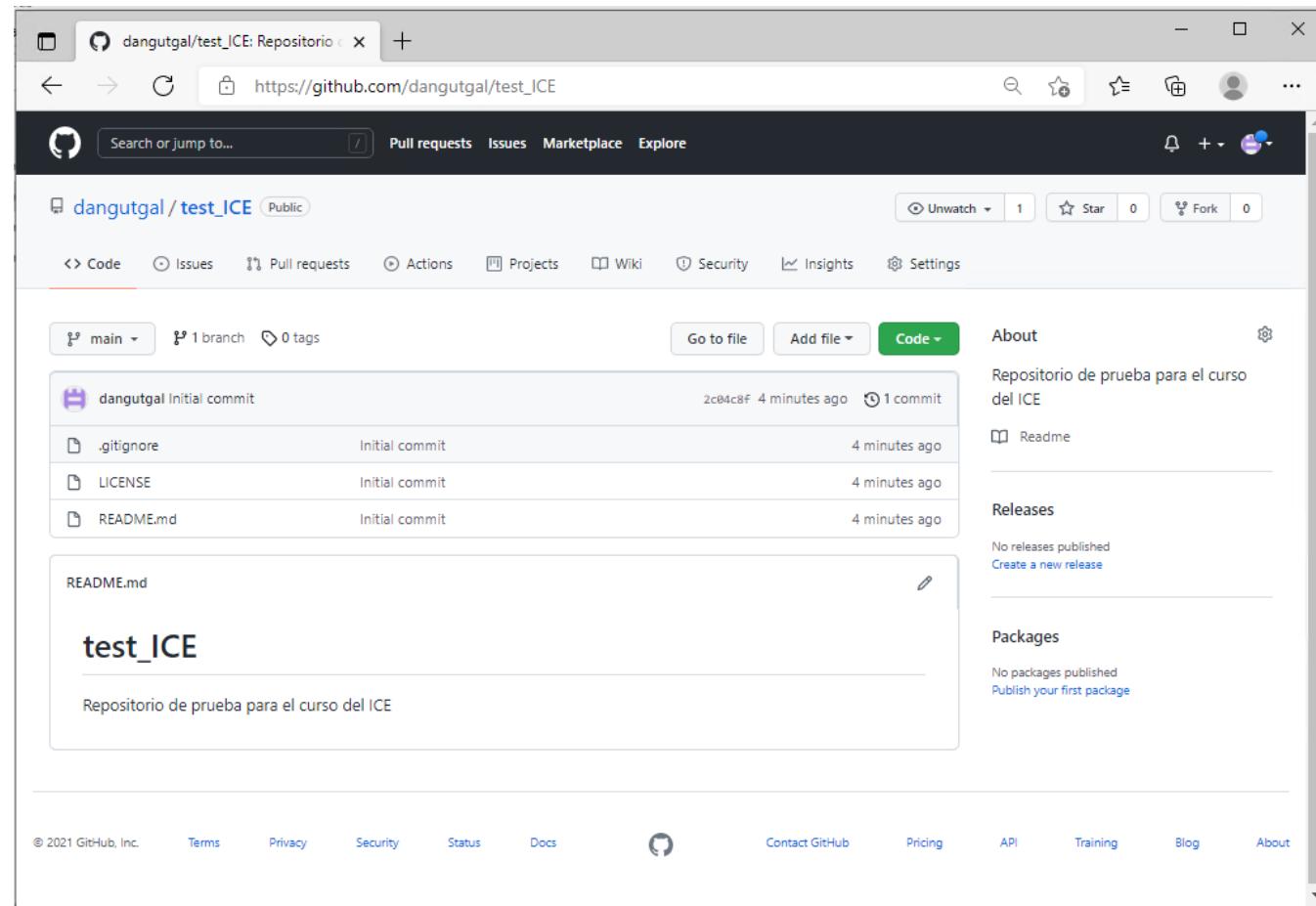
Grant your Marketplace apps access to this repository  
You are subscribed to 1 Marketplace app

GitHub Learning Lab  
Your interactive guide to learning the skills you need without leaving GitHub

Create repository



# Ejemplo de creación de repositorio



# Ejemplo de creación de repositorio

- Tanto para el README como para la Wiki, puede usarse:
  - Markdown:
  - HTML:

```
# Welcome to the OpenNAS wiki!

![OpenNAS logo](https://github.com/RTC-research-group/OpenNAS/blob/master/OpenNA

## What is OpenNAS?
OpenNAS is an open-source VHDL-based Neuromorphic Auditory Sensor (NAS) code generat
create a VHDL project for FPGA. OpenNAS guides designers with a friendly interface
code generation. It includes several audio input interfaces (AC'97 audio codec,
(cascade and parallel filter banks), and a set of neuromorphic output interfaces
everything prepared for building and synthesizing the VHDL project for a target

## Meet the team!
* [Angel Jimenez-Fernandez](https://scholar.google.es/citations?user=LshMmjIAAAJ)
* [Juan Pedro Dominguez-Morales](https://scholar.google.es/citations?user=Z0gb
* [Daniel Gutierrez-Galan](https://scholar.google.es/citations?user=cCn1rQUAAA
* [Alejandro Linares-Barranco](https://scholar.google.es/citations?user=oihgDkoA
* [Gabriel Jimenez-Moreno](https://scholar.google.com/citations?user=TaixTSIAAA
```

<> Edit file Preview

```
1 <p align="center">
2 
4 </p>
5 <h2 name="Description">Description</h2>
6 <!--What OpenNAS is; Its main features-->
7 <p align="justify">
8 OpenNAS is an open source VHDL-based Neuromorphic Auditor
project for FPGA. OpenNAS guides designers with a friendl
several audio input interfaces (AC'97 audio codec, I2S AD
output interfaces (parallel AER, Spinnaker). After NAS ge
using manufacturer's tools.
9 </p>
10
```

<https://guides.github.com/features/mastering-markdown/>



# Ejemplo de creación de repositorio

- Preparar el README.md con una descripción del proyecto, siguiendo una estructura clara:
  - Usa imágenes, vídeos, y listas.
- Preparar la estructura de carpetas.
- Preparar una plantilla para los “issues” (si fuera necesario).
- Preparar la Wiki.
- Preparar el entorno de desarrollo y clona tu repositorio localmente.

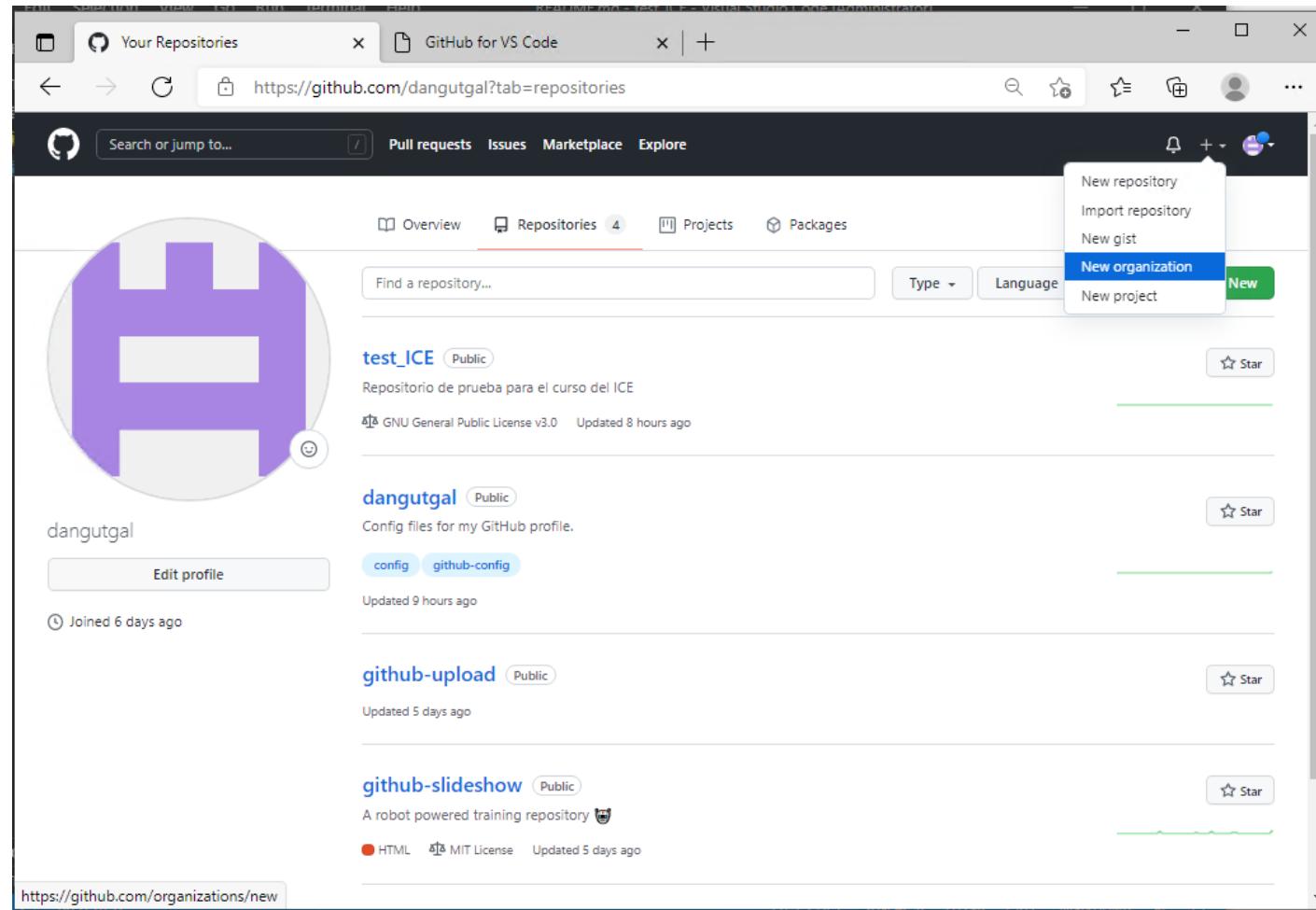
# Índice

- Objetivos.
- Introducción a Git.
- Introducción a GitHub.
- Metodología de creación y gestión de repositorios.
- **Gestión de proyectos colaborativos.**
- Integrando Git en herramientas de trabajo.

# Creando una organización

- Para agrupar y gestionar los repositorios de un departamento o de un grupo de investigación.
- Permite tener todos los repositorios que pertenecen a un mismo grupo, facilitando así a los usuarios la navegación entre repositorios.
- Se pueden crear subgrupos dentro de una misma organización.

# Creando una organización



# Creando una organización

Pick a plan for your organization

How often do you want to pay?

Monthly   Yearly

**Free**

The basics for individuals and organizations

- ① Unlimited public/private repositories
- ② 2,000 automation minutes/month  
Free for public repositories
- ③ 500MB of Packages storage  
Free for public repositories
- ④ New Issues & Projects (beta)
- ⑤ Community support

\$ 0 per user/month

Create a free organization

**Team**

Advanced collaboration for individuals and organizations

- ← Everything included in Free, plus...
- ⑥ Protected branches
- ⑦ Multiple reviewers in pull requests
- ⑧ Draft pull requests
- ⑨ Code owners
- ⑩ Required reviewers
- ⑪ Pages and Wikis
- ⑫ 3,000 automation minutes/month  
Free for public repositories
- ⑬ 2GB of Packages storage  
Free for public repositories
- ⑭ Web-based support

\$ 4 per user/month

Continue with Team

**Enterprise**

Security, compliance, and flexible deployment for enterprises

- ← Everything included in Team, plus...
- ⑮ Automatic security and version updates
- ⑯ SAML single sign-on
- ⑰ Advanced auditing
- ⑱ GitHub Connect
- ⑲ 50,000 automation minutes/month  
Free for public repositories
- ⑳ 50GB of Packages storage  
Free for public repositories

EXCLUSIVE ADD-ONS

- ㉑ Token, secret, and code scanning
- ㉒ Premium support

\$ 21 per user/month

Start Enterprise trial



# Creando una organización

Tell us about your organization

## Set up your organization

Organization account name \*

 ✓

This will be the name of your account on GitHub.  
Your URL will be: <https://github.com/ICEusers>.

Contact email \*

This organization belongs to: \*

My personal account  
i.e., dangutgal

A business or institution  
For example: GitHub, Inc., Example Institute, American Red Cross

Verify your account

Please solve this puzzle so we know you are a real person

[Verify](#)

Start collaborating

## Welcome to ICEusers

Add organization members

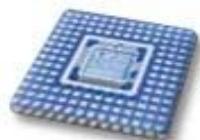
Organization members will be able to view repositories, organize into teams, review code, and tag other members using @mentions.

[Learn more about permissions for organizations →](#)

Search by username, full name or email address

[Complete setup](#)

[Skip this step](#)



# Creando una organización

What do you spend time on most day-to-day?

Please select all that apply

- Writing code
- Managing and coordinating engineering work
- Planning projects
- Billing administration

Other

Please describe

Tell us about your team

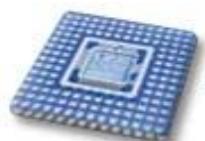
How many people do you expect to actively work within this GitHub organization?

- 0
- 1-5
- 6-15
- 25+

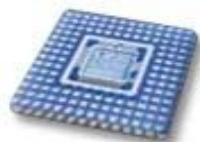
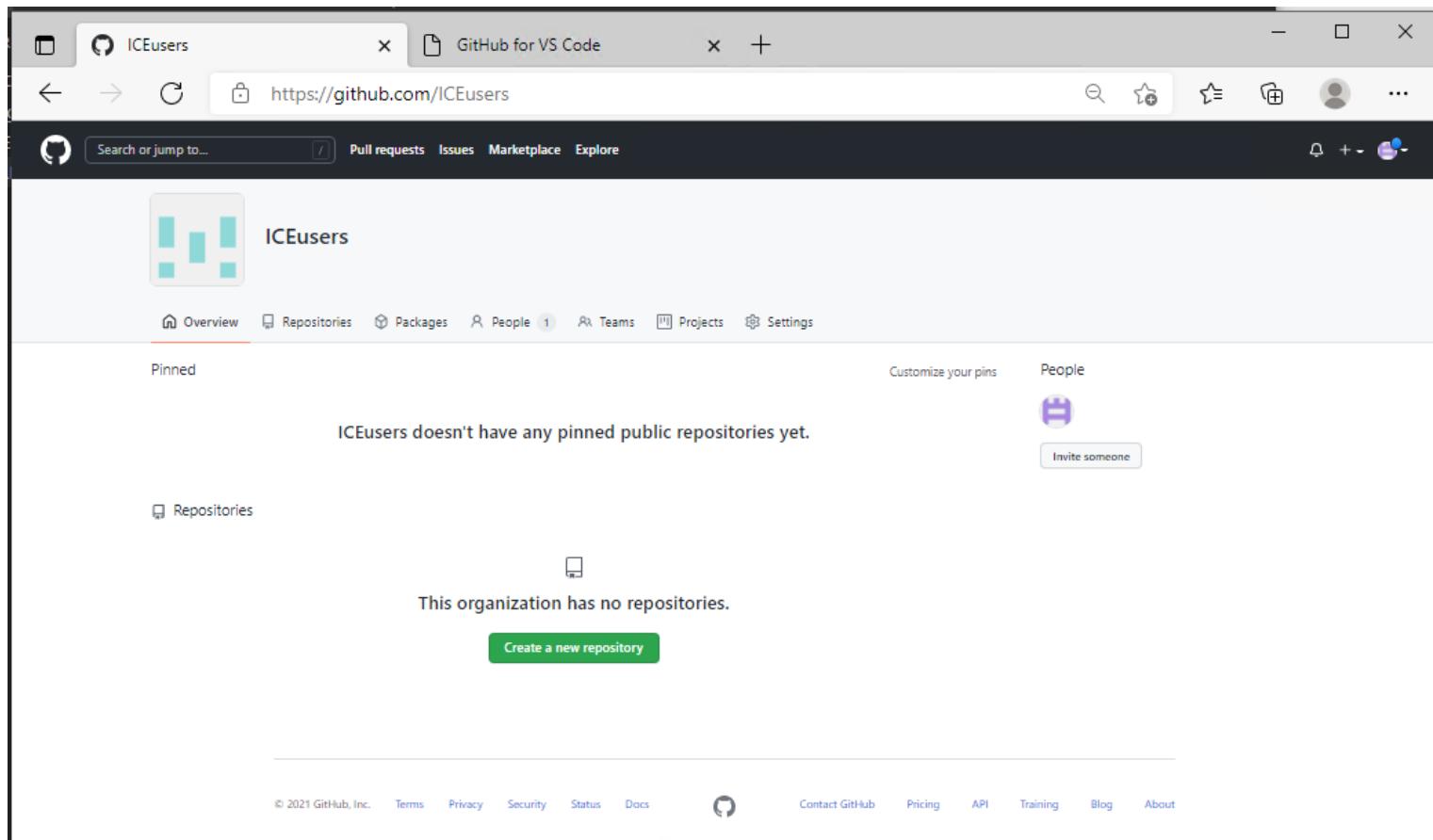
What type of work do you plan to use this organization for?

Please select all that apply

- Open source projects
- Education projects
- Personal projects
- Work projects

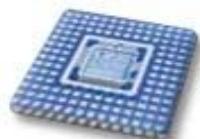


# Creando una organización



# Creando una organización

- A partir de aquí:
  - Invitar a miembros.
  - Crear repositorios.
  - Crear una pantalla de bienvenida y plantilla para colaboradores.
  - Crear proyectos.



# Plantillas en GitHub para colaboradores

- La idea es crear una página principal que recoja la información más importante de tu grupo.
- Enlaces útiles a tutoriales básicos, normas del grupo, formularios, etc.
- Debe ser como la carta de bienvenida a cualquier miembro nuevo de la organización.

# Plantillas en GitHub para colaboradores

- Instrucciones para crear la página principal:
  - Crear nuevo repositorio.
  - Usar el nombre de la organización y añadir “.github.io”:
    - Ejemplo: ICEusers.github.io
  - Hacerlo público.
  - Inicializar el repositorio con un README.md
  - Finalizar el proceso.
- También se conoce como GitHub Pages.

# Plantillas en GitHub para colaboradores

The screenshot shows a GitHub repository interface. At the top, there's a header with a shield icon, a lock icon, and the URL <https://github.com/event-driven-robotics/event-driven-robotics.github.io>. Below the header, there are two files listed: 'index.html' and 'Rename indexNOT.html to index.html'. A timestamp '2 years ago' is shown next to 'index.html'. Underneath these files, there's a 'README.md' file. The main content area features a heading 'Welcome to EDPR Group' with a small emoji of a person with a speech bubble. Below this, a text block says: 'In this repository, you can find some useful information to understand:'. A bulleted list follows:

- What to Do as a New Lab Member 🌈
- Life in the Lab 😊
- ICT 🛡
- iCub (and Lab) Software 📱
- Using the iCub Robot 🤖
- How to Reach IIT 🚅
- Lunch and Dinner 🍽
- How to Open and Close a Mission ✎
- Some Useful Resources 📚

Below this list is a section titled 'What to Do as a New Lab Member' with a rainbow emoji. It contains two numbered steps:

1. The person responsible for welcoming a new lab member should create an issue for him/her using the template available [here](#), and make sure that every step goes smoothly.
2. For getting familiar with the tools used within the team, we suggest to the new lab members to follow the tutorials available in the VVV winter school material. Especially focus on Day 1, 2 and 7.

Another section titled 'Life in the Lab' with a smiling emoji follows. It contains a numbered list of four items:

1. The group has a chat on Microsoft Teams, which you have to install on your pc (there's also a mobile app if you want). You must be added by one of the members. If you have problems accessing with the IIT credentials (like Teams saying you must be invited), write to [ICT\\_Servicedesk@iit.it](mailto:ICT_Servicedesk@iit.it) to grant you the permission.
2. For better understanding the structure of the lab, you can have a look at the [Teams](#) of our GitHub organization.
3. We usually organise an 📅 EDPR group meeting per week and a 📖 Journal Club every two weeks.
4. If you want to invite someone to IIT, you need to ask permission to the Group Leader (aka Chiara Bartolozzi) using [this template email](#). There are a few IIT address that we should put in carbon copy in the email (as indicated in the template) for logistics reasons. The visit permission will be granted if the Group Leader will reply "ok" to the email.

## How to Reach IIT 🚅

IIT offers a shuttle service. The timetable of the shuttle is available in the iCub Facility intranet at this [link](#). Otherwise, you can also use public transportation such as trains (from Bolzaneto or San Biagio train station) and buses (line 7 from Brin - terminus of the metro service).

## Lunch and Dinner 🍽

Currently, our building (CRIS) does not have any canteen therefore the available options for lunch/dinner are:

1. Bring a lunch box from home and use the fridge/microwave available in the kitchen;
2. Order a lunch box from IIT headquarter canteen using the app [Appetie](#) or the [website](#). The price will be the same of the canteen, with an extra cost of 1 euro for the delivery.
3. Go to the canteen at the 6th floor of [IIT headquarter](#) in via Morego, 30 (opening time for lunch is from 12 to 14);
4. Walk about 10 minutes and get some groceries or prepared meals from [IPERCOOP](#) supermarket.
5. Walk about 5 minutes and get some focaccia or pizza [here](#).

## How to Open and Close a Mission ✎

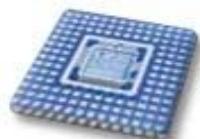
1. When you need to travel for some mission, e.g. to a conference, you should write first an email to your supervisor putting in carbon copy also the responsible person in the administration (Marta Caracalli - [marta.caracalli@iit.it](mailto:marta.caracalli@iit.it)).
2. Once you receive confirmation to your email you have to open a mission in the [SAP travel system](#).
3. When you return from the mission you have to add your expenses ([KEEP](#) the receipts) in the SAP travel system.
4. On a blank paper, attach all your receipts, numbering them with the corresponding number in the SAP system.
5. Add your surname and mission number and bring everything to Marta Caracalli.

## Some Useful Resources 📚



# Creando un proyecto

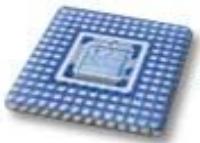
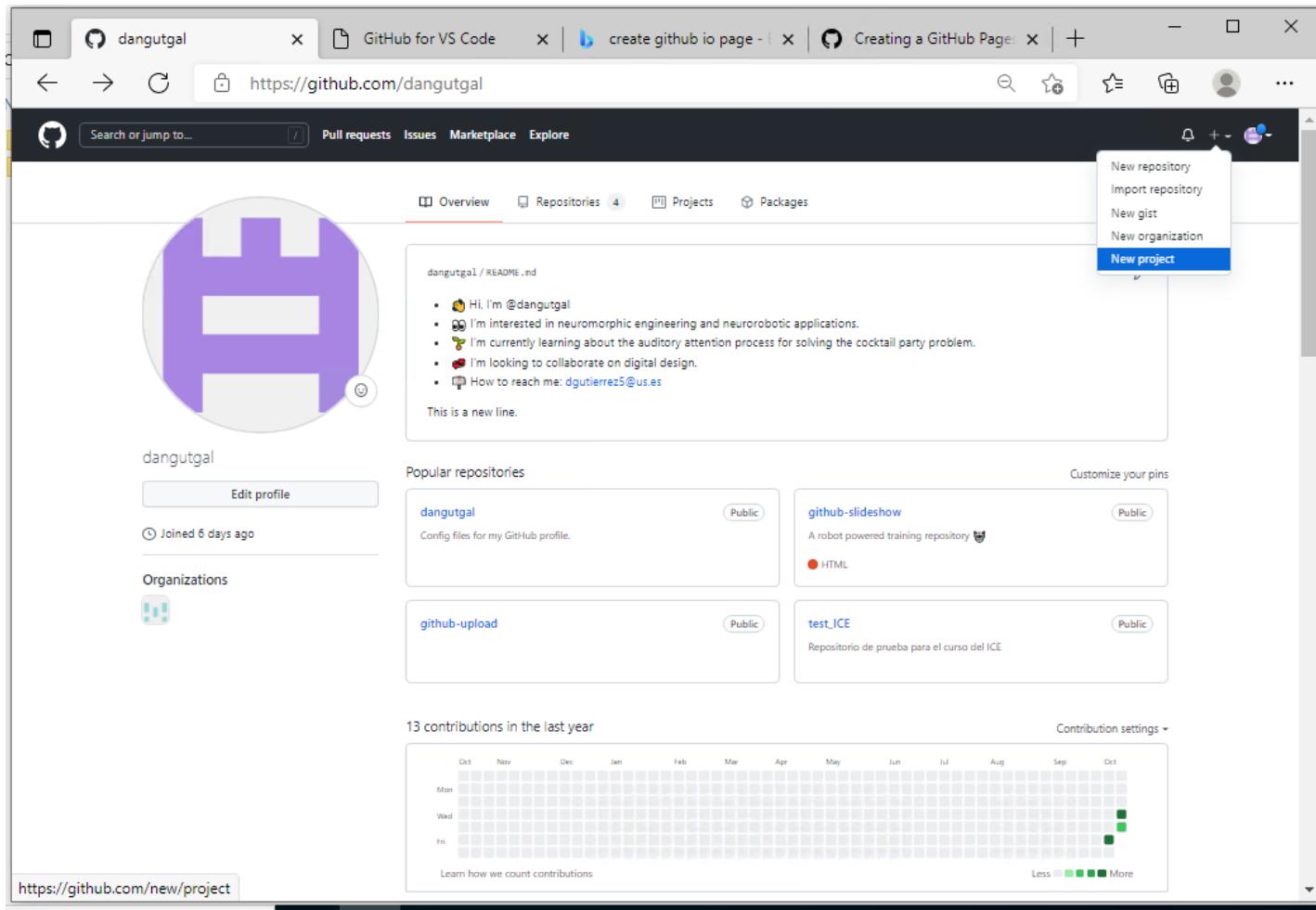
- Sirve para crear, gestionar y llevar un seguimiento de un proyecto asociado a uno o varios repositorios.
- Normalmente hay 3 columnas:
  - To do.
  - In progress.
  - Done.



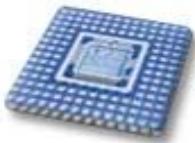
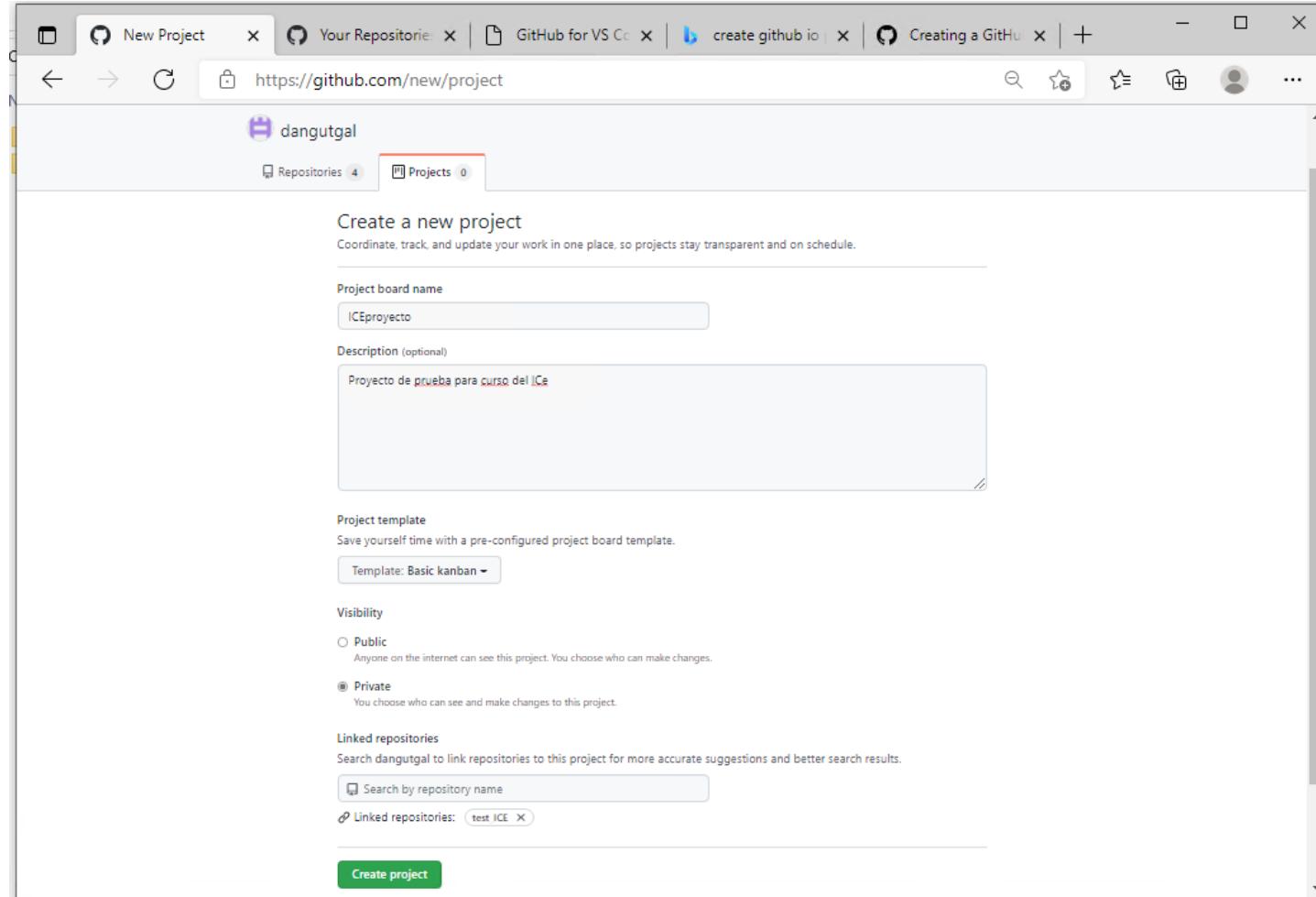
# Creando un proyecto

- Se añaden notas o “tarjetas”, que son tareas sencillas.
- Se pueden asociar tarjetas a Issues y también a ramas concretas del repositorio.
- Las tarjetas pueden contener checkboxs, listas, etc (según Markdown)
- Similar a Trello.
- Es la mejor opción para gestionar un proyecto de desarrollo.

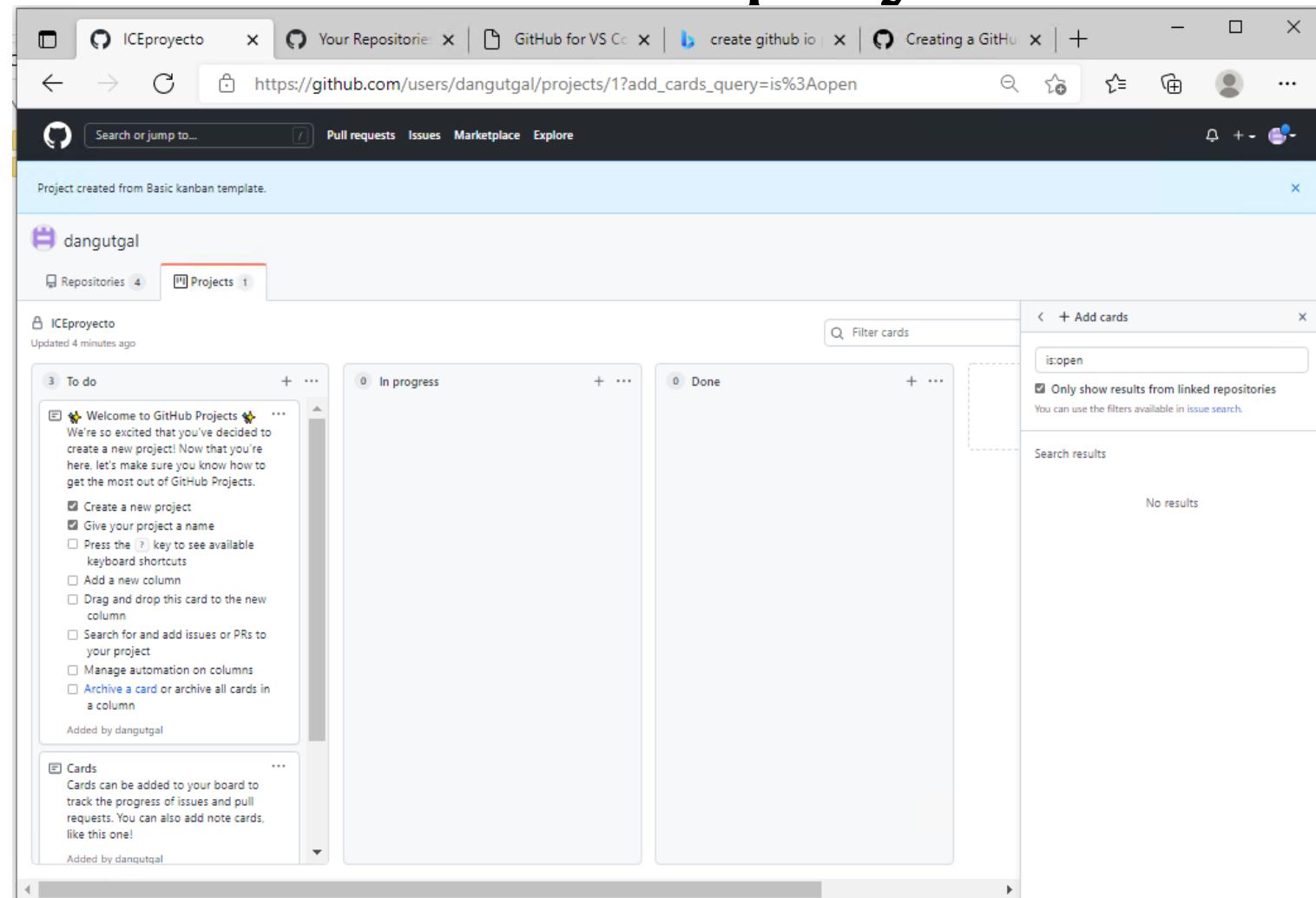
# Creando un proyecto



# Creando un proyecto



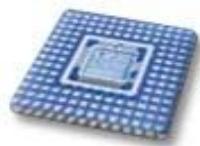
# Creando un proyecto



# Creando un proyecto

The screenshot shows a GitHub project board for the repository "Event Driven Perception for Robotics". The project is titled "attend-track-predict" and was last updated on 27 Mar 2019. The board has four columns: "Tasks To Do", "In Process", "Closed this week", and "Closed".

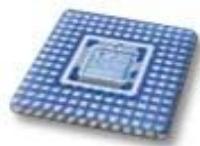
- Tasks To Do:** 1 item: "create simple linear prediction module" (opened by arenglover)
- In Process:** 0 items
- Closed this week:** 2 items:
  - "create state machine module that calls start/stop of all modules" (opened by arenglover)
  - "add robot control to state machine" (opened by arenglover)
- Closed:** 5 items:
  - "add start/stop commands" (opened by arenglover)
  - "create draft image of signal flow" (opened by arenglover)
  - "create simple circle detection" (opened by arenglover)
  - "add RPC interface for start and stop of circle-attention" (opened by arenglover)
  - "efficient pause/interrupt of modules" (opened by arenglover)
    - in event-driven port.h



# Creando un proyecto

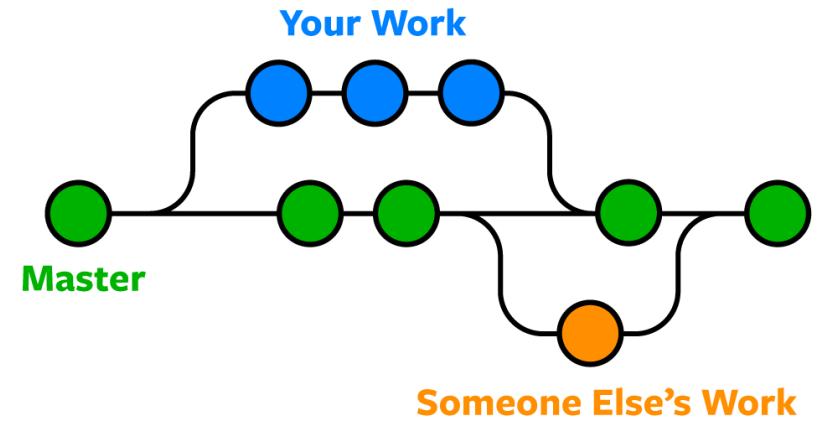
The screenshot shows a GitHub project board for the repository "Event Driven Perception for Robotics". The project is named "attend-track-predict" and was last updated on 27 Mar 2019. The board has four columns: "Tasks To Do", "In Process", "Closed this week", and "Closed".

- Tasks To Do:** 1 item: "create simple linear prediction module" (opened by arrenglover)
- In Process:** 0 items
- Closed this week:** 2 items:
  - "create state machine module that calls start/stop of all modules" (opened by arrenglover)
  - "add robot control to state machine" (opened by arrenglover)
- Closed:** 5 items:
  - "add start/stop commands" (opened by arrenglover)
  - "create draft image of signal flow" (opened by arrenglover)
  - "create simple circle detection" (opened by arrenglover)
  - "add RPC interface for start and stop of circle-attention" (opened by arrenglover)
  - "efficient pause/interrupt of modules" (in event-driven port.h)
    - Added by arrenglover



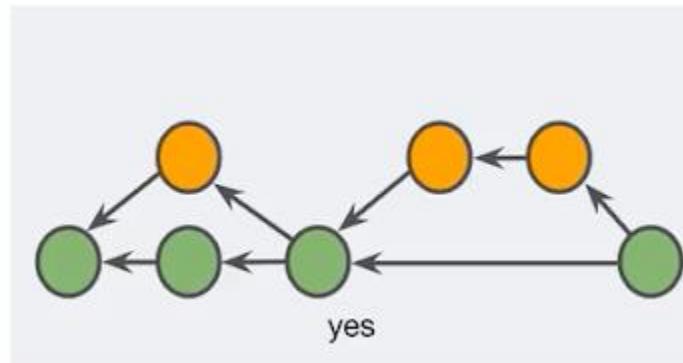
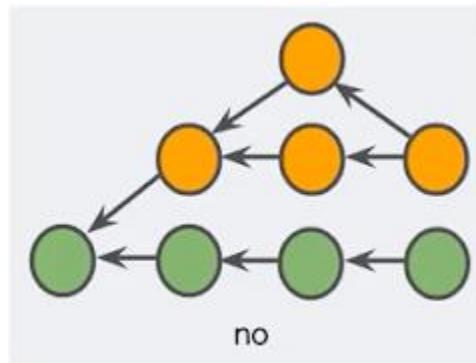
# ¿Qué es y cómo crear una rama?

- Una rama es una versión paralela del repositorio que está contenida en el repositorio, pero no afecta a la rama principal u original.
- Permite trabajar en nuevas funcionalidades sin “estropear” la rama activa o principal.
- Las ramas se pueden fusionar a la rama principal.



# ¿Qué es y cómo crear una rama?

- Las ramas es la principal funcionalidad de Git.
- Permite desarrollo paralelo de varios componentes del equipo.
- Hay que tener cuidado con las ambigüedades y con las modificaciones múltiples.



# ¿Qué es y cómo crear una rama?

The image shows two screenshots of a GitHub repository named 'dangutgal/test\_ICE'.  
The left screenshot shows the repository page with a 'Code' tab selected. A modal window titled 'Switch branches/tags' is open, showing a search bar with 'Find or create a branch...' and a list of branches: 'main' (selected), 'desarrollo' (highlighted with a blue border), and 'View all branches'.  
The right screenshot shows the repository page after creating a new branch. It displays the 'main' branch (1 branch, 0 tags). A modal window titled 'Switch branches/tags' shows the newly created branch 'desarrollo' selected. Below it, a message says 'Create branch: desarrollo from 'main''. The repository content includes a 'README.md' file with the text 'test\_ICE' and 'Repositorio de prueba para el curso del ICE'.

# ¿Qué es y cómo crear una rama?

The screenshot shows a GitHub repository page for the user 'dangutgal' named 'test\_ICE'. The repository is public and has 2 branches and 0 tags. The 'Code' tab is selected. A message indicates that the 'desarrollo' branch is even with 'main'. The commit history shows an initial commit by 'dangutgal' from 9 hours ago, which includes files '.gitignore', 'LICENSE', and 'README.md'. The README file contains the text 'test\_ICE' and 'Repositorio de prueba para el curso del ICE'. The right sidebar provides links for 'About', 'Readme', 'GPL-3.0 License', 'Releases', and 'Packages'.

dangutgal / test\_ICE Public

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

desarrollo 2 branches 0 tags Go to file Add file Code

This branch is even with main. Contribute

dangutgal Initial commit 9 hours ago 1 commit  
.gitignore Initial commit 9 hours ago  
LICENSE Initial commit 9 hours ago  
README.md Initial commit 9 hours ago

README.md

test\_ICE

Repositorio de prueba para el curso del ICE

About

Repository de prueba para el curso del ICE

Readme

GPL-3.0 License

Releases

No releases published Create a new release

Packages

No packages published Publish your first package

# ¿Qué es y cómo crear una rama?

The screenshot shows a GitHub repository page for `dangutgal/test_ICE`. The top navigation bar includes links for Pull requests, Issues, Marketplace, and Explore. Below the repository name, there are buttons for Unwatch (1), Star (0), Fork (0), and Insights. The Insights tab is currently selected. On the left, a sidebar menu lists Pulse, Contributors, Community, Traffic, Commits, Code frequency, Dependency graph, Network (which is highlighted with a red border), and Forks. The main content area is titled "Network graph" and displays a timeline of commits. The timeline shows two commits from the owner `dangutgal` on October 13 and 14. The commit on October 13 is labeled "main" and the commit on October 14 is labeled "desarrollo". A blue arrow points from the "main" commit to the "desarrollo" commit, indicating a branch relationship.

# ¿Qué es y cómo crear un fork?

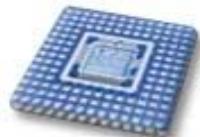
- Un fork es una copia personal de un repositorio de otro usuario en nuestra propia cuenta.
- Te permite hacer cambios libremente de otro proyecto sin afectar al original.
- Sin embargo, están sincronizados, y los cambios en el repositorio origen pueden afectar a tu repositorio.
- Se puede solicitar un pull-request.

# ¿Qué es y cómo crear un fork?

- Un fork es una copia personal de un repositorio de otro usuario en nuestra propia cuenta.
- Te permite hacer cambios libremente de otro proyecto sin afectar al original.
- Sin embargo, están sincronizados, y los cambios en el repositorio origen pueden afectar a tu repositorio.
- Se puede solicitar un pull-request.

# ¿Qué es y cómo crear un fork?

- Ejemplo:
  - Buscar algún repositorio útil: pyNAVIS, de jpdominguez.
  - Hacer un fork.
  - Añadir nuevas funcionalidades.
  - Si lo permite y lo acepta, sugerir la integración con el original.



# ¿Qué es y cómo crear un fork?

The screenshot shows a web browser window with two tabs: 'Search · pyNAVIS' and 'Your Repositories'. The main content area displays a GitHub search results page for the query 'pyNAVIS'. The results section is titled '2 repository results'.

**Repositories** (2)

- Code (64)
- Commits (10)
- Issues (4)
- Discussions (0)
- Packages (0)
- Marketplace (0)
- Topics (0)
- Wikis (0)
- Users (0)

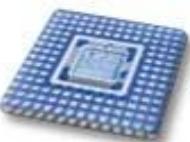
**Languages**

- Python (2)

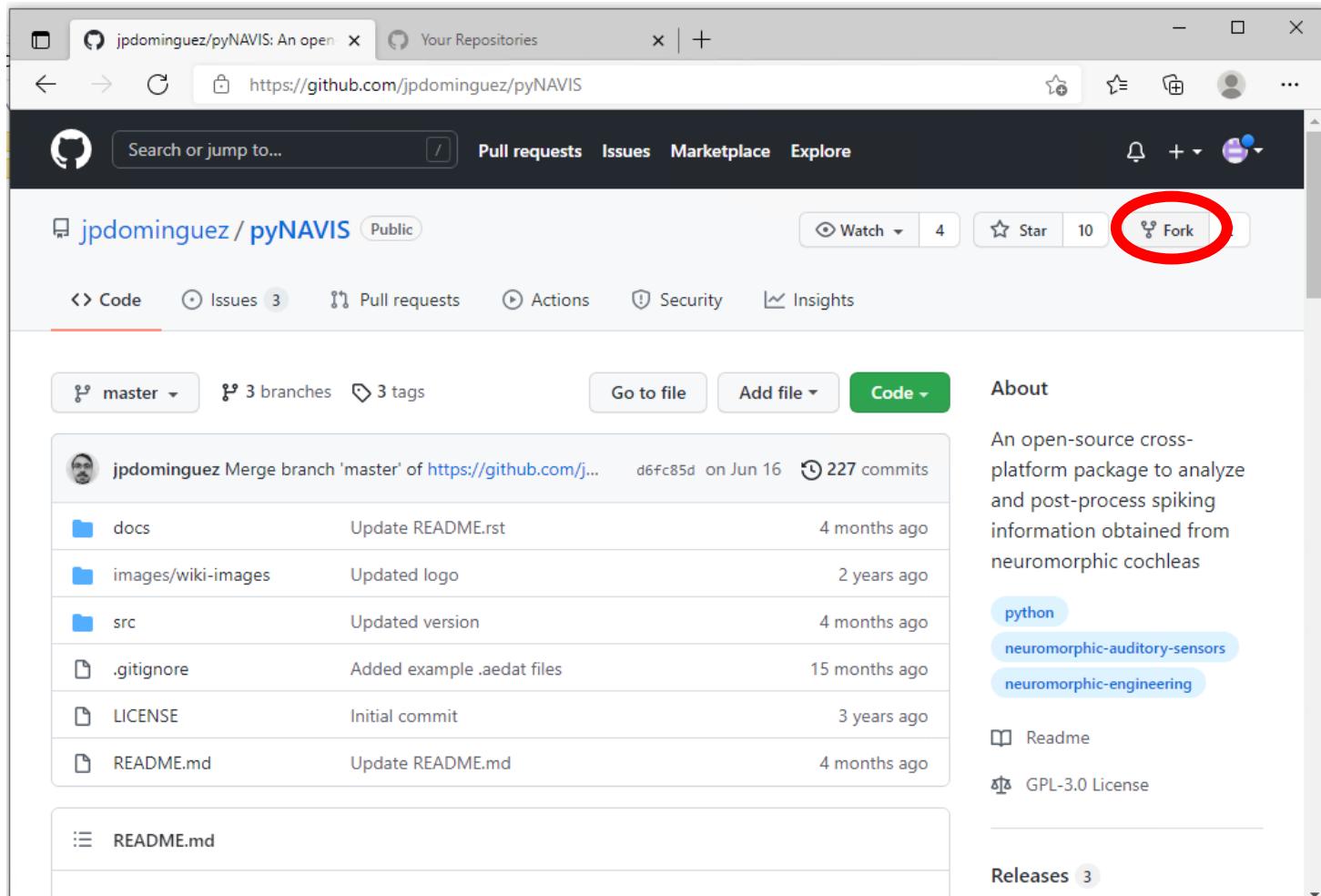
**2 repository results**

**jpdominguez/pyNAVIS**  
An open-source cross-platform package to analyze and post-process spiking information obtained from neuromorphic coch...  
python neuromorphic-auditory-sensors neuromorphic-engineering  
10 Python GPL-3.0 license Updated on Jun 16

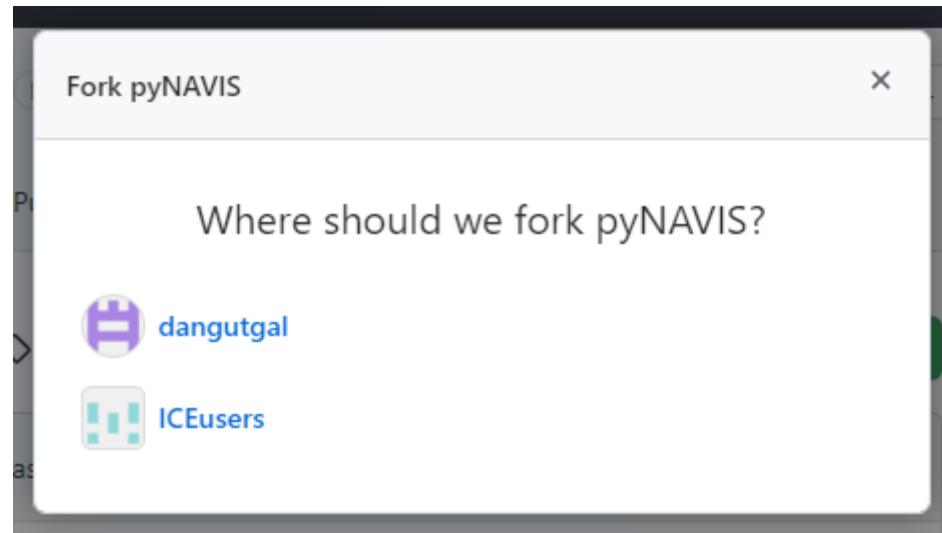
**degoldschmidt/PyNaviSim**  
Python version of NaviSim using Pygame for GUI  
Python GPL-3.0 license Updated on Oct 29, 2014



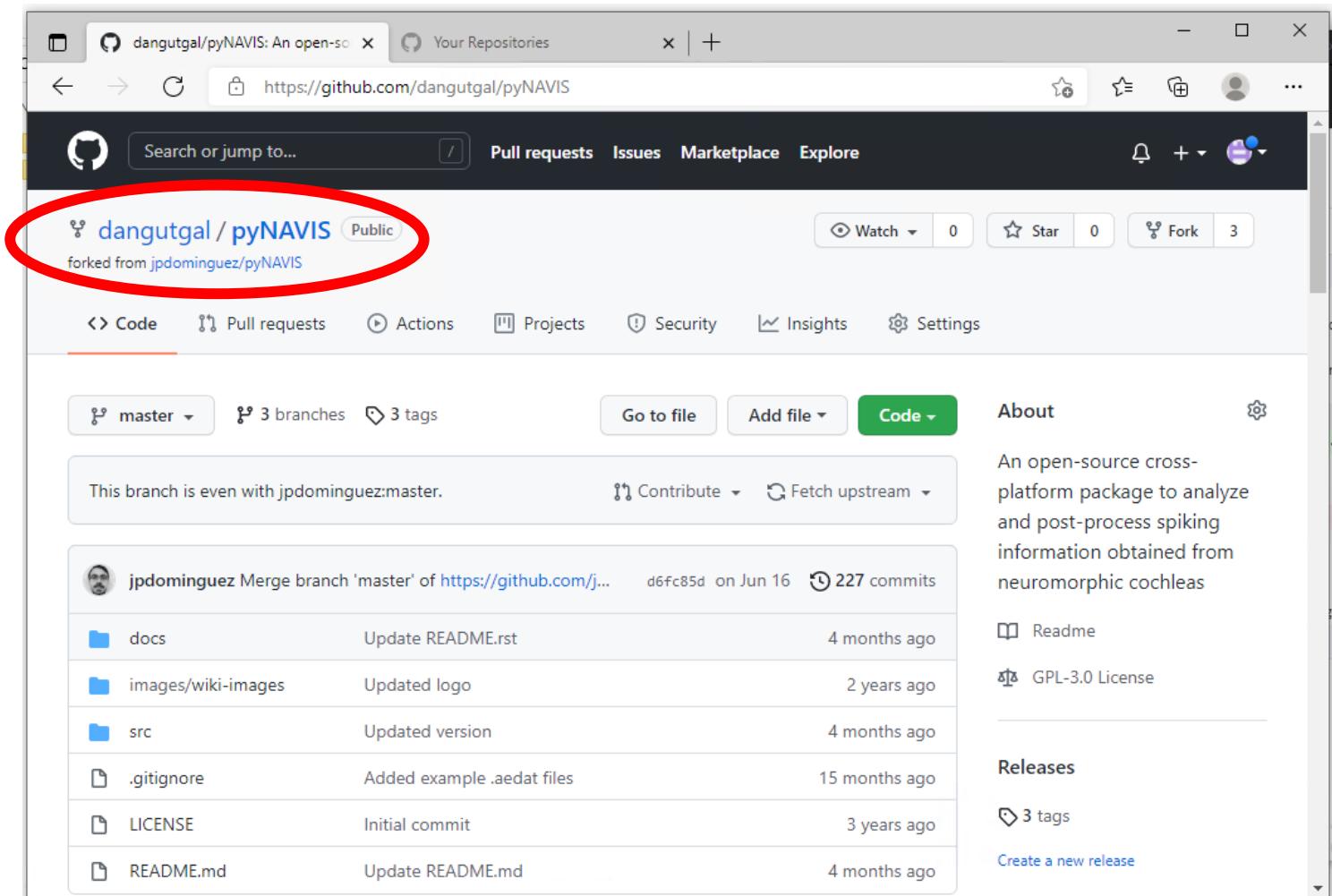
# ¿Qué es y cómo crear un fork?



# ¿Qué es y cómo crear un fork?



# ¿Qué es y cómo crear un fork?



# Integración de cambios a través de pull request

- Un pull request es una característica de la plataforma GitHub.
- Su objetivo es realizar una integración (o merge) de una rama que incluye algunas nuevas características en la rama principal del proyecto.
- Permite la comunicación entre miembros del equipo con:
  - Notificaciones, comentarios, revisiones de código, foro de discusión, y toma de decisiones.

# Integración de cambios a través de pull request

- **¡Requiere documentación!**
- En este caso, los commits, así como sus descripciones, son muy importantes.
- Normalmente se requieren cambios, que se van añadiendo automáticamente al pull request.
- Se pueden añadir comentarios a archivos o a líneas específicas de un archivo.

# Integración de cambios a través de pull request

The screenshot shows a GitHub pull request page for the repository `robotology / event-driven`. The pull request is titled "Adds Windows api export symbols + small fixes for windows compilation #124". The status is "Open", and it shows a merge from the `windows` branch into the `master` branch. The pull request has 13 conversations, 1 commit, 0 checks, and 17 files changed.

**Comments:**

- aerydna** commented on 3 Dec 2020:

adds the windows compatibility. what was failing was particular static member variables that were used in template inline function, so directly compiled in the user compilation target but not properly exported. in linux you can't see those kind of errors because everything get exported by default. if this is not useful or important feel free to ignore the pull request or closing it. it miss extensive test, i just tested the compilation and the FramerLite application.
- arrenglover** self-assigned this on 3 Dec 2020
- arrenglover** commented on 3 Dec 2020:

Thanks Andrea - this is most certainly useful. something that has been on the to-do list for a while. I'll review it as soon as I can actually.
- traversaro** commented on 4 Dec 2020:

Thanks @aerydna , a few comments:

  - In my experience using `EXPORT` as a macro definition is dangerous, as you could have collisions with other libraries that use the same definition. Using a project specific macro such as `EVENT_DRIVEN_EXPORT` may be probably more safe.

**Reviewers:** No reviews

**Assignees:** arrenglover

**Labels:** None yet

**Projects:** None yet

**Milestone:** No milestone

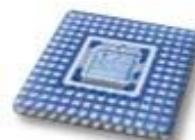
**Linked issues:** Successfully merging this pull request will close these issues: None yet

**Notifications:** [Subscribe](#)

# Integración de cambios a través de pull request

The screenshot shows a GitHub pull request page for the repository 'robotology / event-driven'. The pull request is titled 'Cochlea integration #132'. It is marked as 'Merged' and shows a merge commit from 'arrenglover' into the 'master' branch from the 'cochlea\_integration' branch. The pull request has 140 commits, 1 check, and 28 files changed. The conversation section shows a comment from 'arrenglover' stating 'No description provided.' Below the commits, a list of individual commits is shown, each with a author, message, and commit hash.

Author	Commit Message	Commit Hash
arrenglover	Update eventcodecs.md	1515559
arrenglover	Revert "Update eventcodecs.md"	8f419ca
arrenglover	Created codec_CochleaEvent	eee91fb
arrenglover	Updated vCodec.h with CochleaCodec	35dcdb9
arrenglover	Added CochleaEvent	7bade3a
arrenglover	Update eventcodecs.md	a2546ab
arrenglover	Bug in class CochleaEvent fixed	775a468



# Integración de cambios a través de pull request

Cochlea integration #132

Merged arenglover merged 140 commits into master from cochlea\_integration on 16 Mar

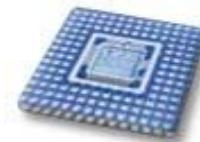
Conversation 1 Commits 140 Checks 1 Files changed 28 +2,591 -15

Changes from 1 commit ▾ File filter ▾ Conversations ▾ Jump to ▾ Review changes ▾

Update eventcodecs.md

dgtierrezATC authored and arenglover committed on 15 Mar commit 1515559150b16f6f8680b521d2fa18b9c63f2628

```
diff --git a/documentation/eventcodecs.md b/documentation/eventcodecs.md
@@ -37,6 +37,10 @@ A **GaussianAE** extends a cluster event with a 2 dimensional Gaussian distribut
37 ...
38 [sxssxsxsxsxsxsx sxssxsxsxsxsx sxssxsxsxsxsx sxssxsxsxsxsx sysysysysysy sysysysysysy sysysysysysy
39 sysysysysysy sysxsysxsysxsysxsysx sysxsysxsysxsysxsysxsysx sysxsysxsysxsysxsysxsysxsysx]
40 ...
41 # Coding in YARP
42 ...
43 + A **CochleaEvent** uses 4 bytes to encode frequency channel (_F_), polarity (_P_), neuron id of the Jeffress model
44 (_N_), the olive model (_O_), the auditory model (_M_), and channel (_C_). Importantly, as CochleaEvent is of type
45 `vEvent` the timestamp information of this event is always encoded as well.
46 + ...
47 + [00000000 0C000M00 0NNNNNNN 0FFFFFFP]
48 ...
49 ...
50 256 4 4 2 'A' 'E' 257 4 -2140812352 15133 -2140811609 13118 4 4 'F' 'L' 'O' 'W' 257 4 -2140812301 13865 -1056003417
51 - ...
52 ...
53 ...
54 256 4 4 2 'A' 'E' 257 4 -2140812352 15133 -2140811609 13118 4 4 'F' 'L' 'O' 'W' 257 4 -2140812301 13865 -1056003417
55 + ...
```



# Integración de cambios a través de pull request

Cochlea integration #132

Merged arrenglover merged 140 commits into master from cochlea\_integration on 16 Mar

Conversation 1 Commits 140 Checks 1 Files changed 28 +2,591 -15

Changes from all commits ▾ File filter ▾ Conversations ▾ Jump to ▾ Review changes ▾

Viewed

`@@ -37,6 +37,10 @@ A **GaussianAE** extends a cluster event with a 2 dimensional Gaussian distribut`

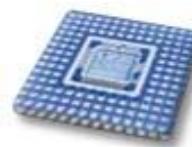
`37 ...
38 [sxssxsxsxsxsxsx sxssxsxsxsxsxsx sxssxsxsxsxsxsx sxssxsxsxsxsxsx sysysysysysy sysysysysysy]`

`39 ...
40 + A **CochleaEvent** uses 4 bytes to encode frequency channel (_F_), polarity (_P_), neuron id of the Jeffress model (_L_), the olive model (_O_), the auditory model (_M_), and channel (_C_). Importantly, as CochleaEvent is of type `vEvent` the timestamp information of this event is always encoded as well.
41 +
42 + [00000100 0C100NNN NNNNRRM0 FFFFFFFP]
43 + ...`

`44 ...
45 # Coding in YARP
46 ...`

`@@ -27,7 +27,9 @@ set(folder_source`

`27 src/codecs/codec_LabelledAE.cpp
28 src/codecs/codec_SkinEvent.cpp
29 src/codecs/codec_SkinSample.cpp
30 - src/codecs/codec_vEvent.cpp
31 ...
32 if(OpenCV_FOUND)
33 list(APPEND folder_source src/vIPT.cpp
34 ...
35 list(APPEND folder_source src/vIPT.cpp)`



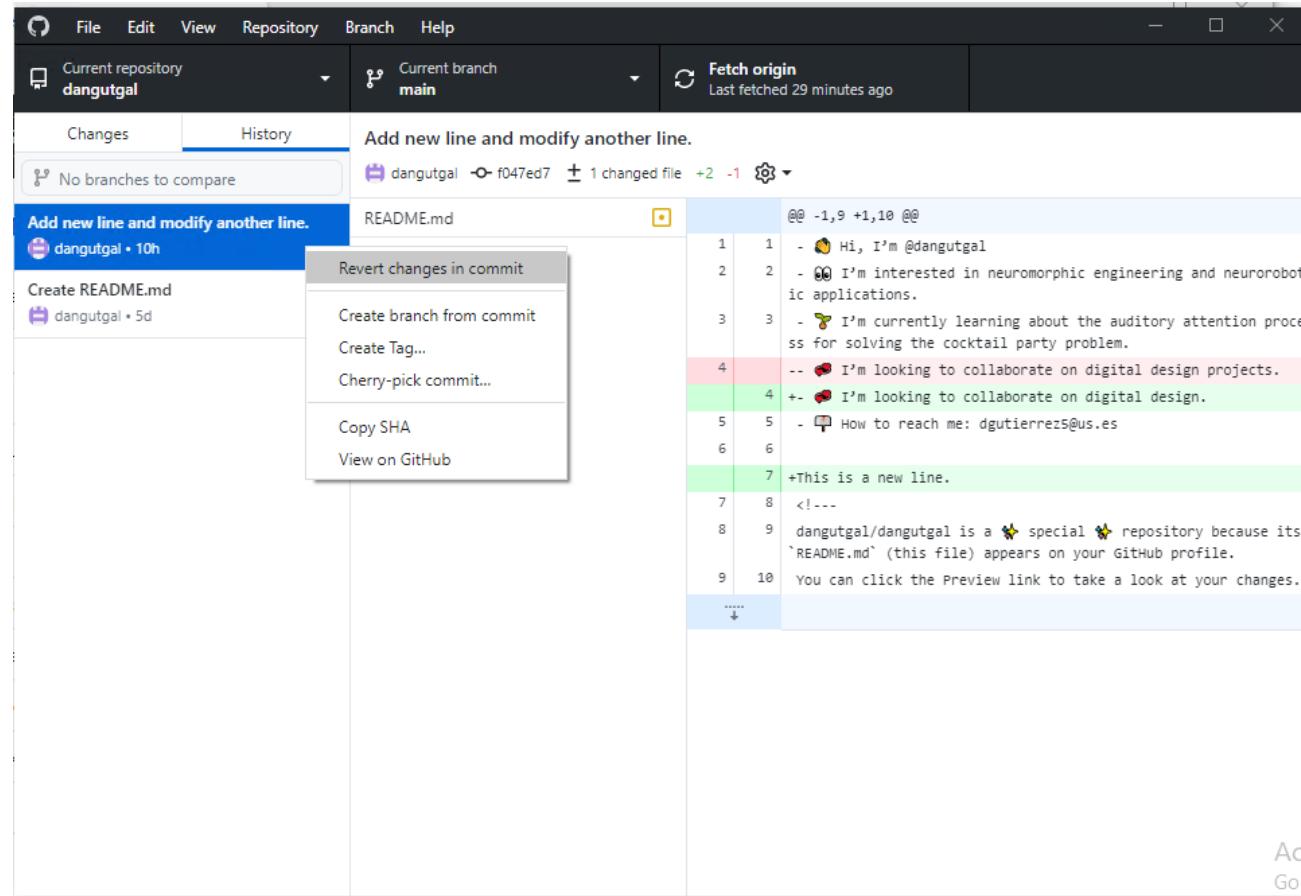
# ¿Qué ocurre cuando hay conflicto entre versiones?

- Hay que revisar los cambios uno por uno:
  - Si el conflicto es solo por haber modificado el mismo documento pero en distintas partes → fácil solución.
  - Si el conflicto es por haber modificado el mismo documento, y la misma parte de código → GRAN PROBLEMA.
- Hay que decidir cómo proceder en función de las implicaciones.

# Retrocediendo a una versión anterior usando el historial

- Útil cuando un commit no funciona correctamente o cuando se ha producido un error de conflicto.
- Sin embargo, los cambios retrocedidos no desaparecen del historial, sino que aparecen como un nuevo cambio.
- De esta forma, queda constancia de cada cambio.

# Retrocediendo a una versión anterior usando el historial



# Retrocediendo a una versión anterior usando el historial

File Edit View Repository Branch Help

Current repository: dangutgal Current branch: main Push origin: Last fetched just now

Changes History Add new line and modify another line.

No branches to compare

Revert "Add new line and modi..." dangutgal · just now

Add new line and modify another line. dangutgal · 10h

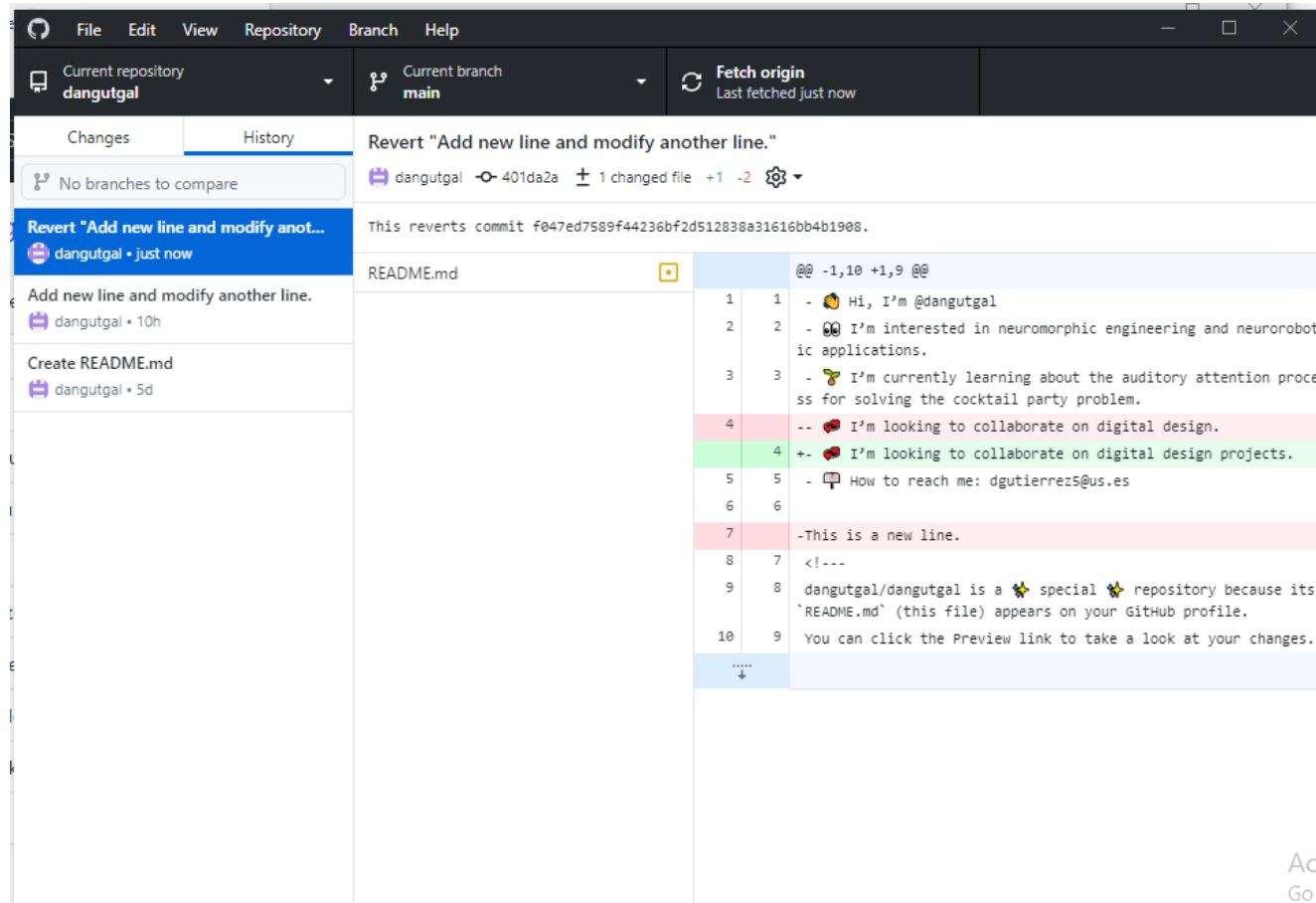
Create README.md dangutgal · 5d

README.md

```
@@ -1,9 +1,10 @@
1 1 - 🎉 Hi, I'm @dangutgal
2 2 - 🎉 I'm interested in neuromorphic engineering and neurorobotic applications.
3 3 - 🎉 I'm currently learning about the auditory attention process for solving the cocktail party problem.
4 4 -- 🎉 I'm looking to collaborate on digital design projects.
4 4 +- 🎉 I'm looking to collaborate on digital design.
5 5 - 📧 How to reach me: dgutierrez5@us.es
6 6
7 +This is a new line.
8 <!--
9 dangutgal/dangutgal is a 🎉 special 🎉 repository because its `README.md` (this file) appears on your GitHub profile.
10 You can click the Preview link to take a look at your changes.
```



# Retrocediendo a una versión anterior usando el historial



# Índice

- Objetivos.
- Introducción a Git.
- Introducción a GitHub.
- Metodología de creación y gestión de repositorios.
- Gestión de proyectos colaborativos.
- **Integrando Git en herramientas de trabajo.**

# ¿Qué herramientas lo permiten?

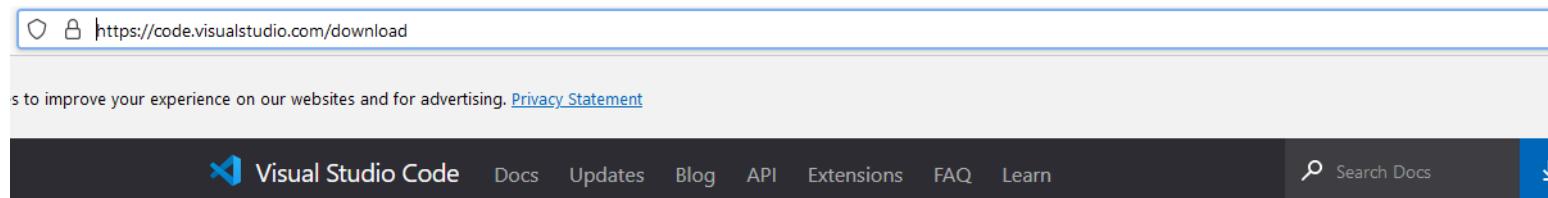
- Actualmente, casi cualquier entorno de desarrollo lo integra:
  - Visual Studio y Visual Studio Code.
  - NotePad++.
  - Sublime.
  - Overleaf.
  - Otros...
- Incluso herramientas CAD tienen soporte para Git:
  - Altium, Eagle, AutoCAD...

# ¿Qué herramientas lo permiten?

- Para desarrollos software:
  - Es recomendable Visual Studio Code (incluso GitHub lo recomienda por defecto)
- Para la instalación de Visual Studio Code:
  - <https://code.visualstudio.com/download>
  - Seleccionar tu sistema operativo.
  - [Documentation for Visual Studio Code](#)



# Instalando Visual Studio Code



A screenshot of a web browser displaying the Visual Studio Code download page at <https://code.visualstudio.com/download>. The page has a dark header with the Visual Studio Code logo and navigation links for Docs, Updates, Blog, API, Extensions, FAQ, and Learn. A search bar is on the right. Below the header, there's a message about improving the experience and a privacy statement link. The main content area is titled "Download Visual Studio Code" and describes it as "Free and built on open source. Integrated Git, debugging and extensions." It features icons for Windows, Linux, and macOS, each with a download button.

## Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.



[↓ Windows](#)  
Windows 7, 8, 10, 11

User Installer  
System Installer  
.zip



[↓ .deb](#)  
.deb  
Debian, Ubuntu

.deb  
.rpm  
.tar.gz

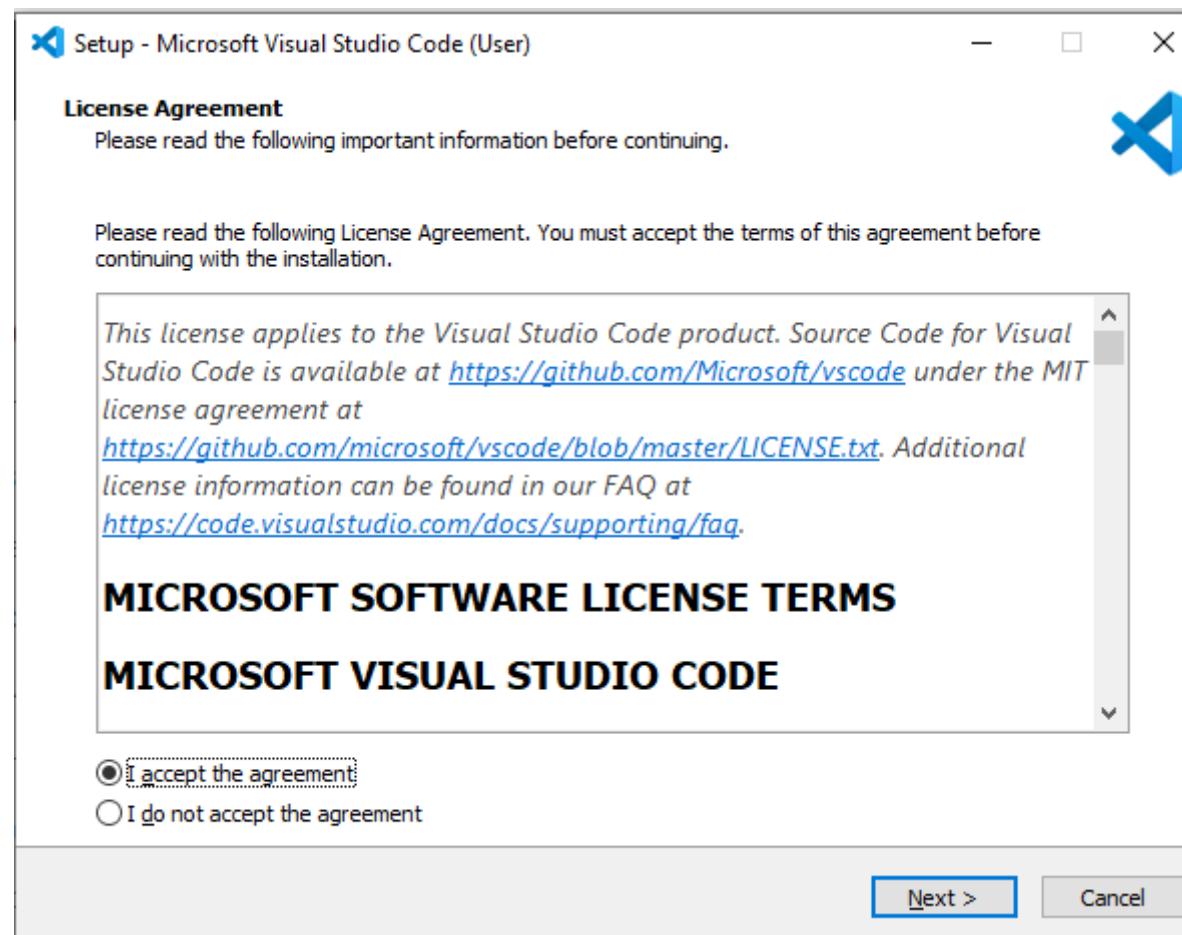


[↓ Mac](#)  
macOS 10.11+

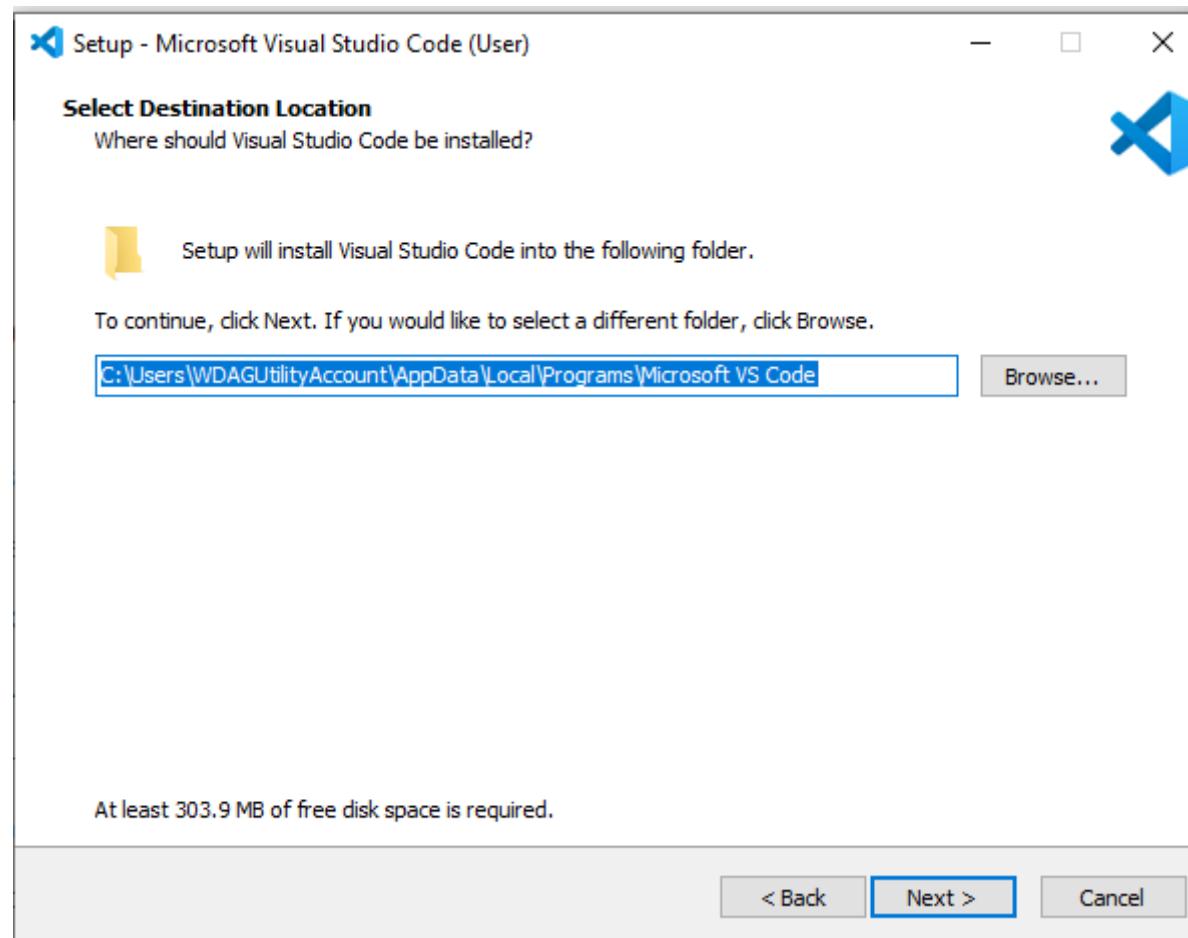
.zip  
Universal  
Intel Chip  
Apple Silicon

Snap Store

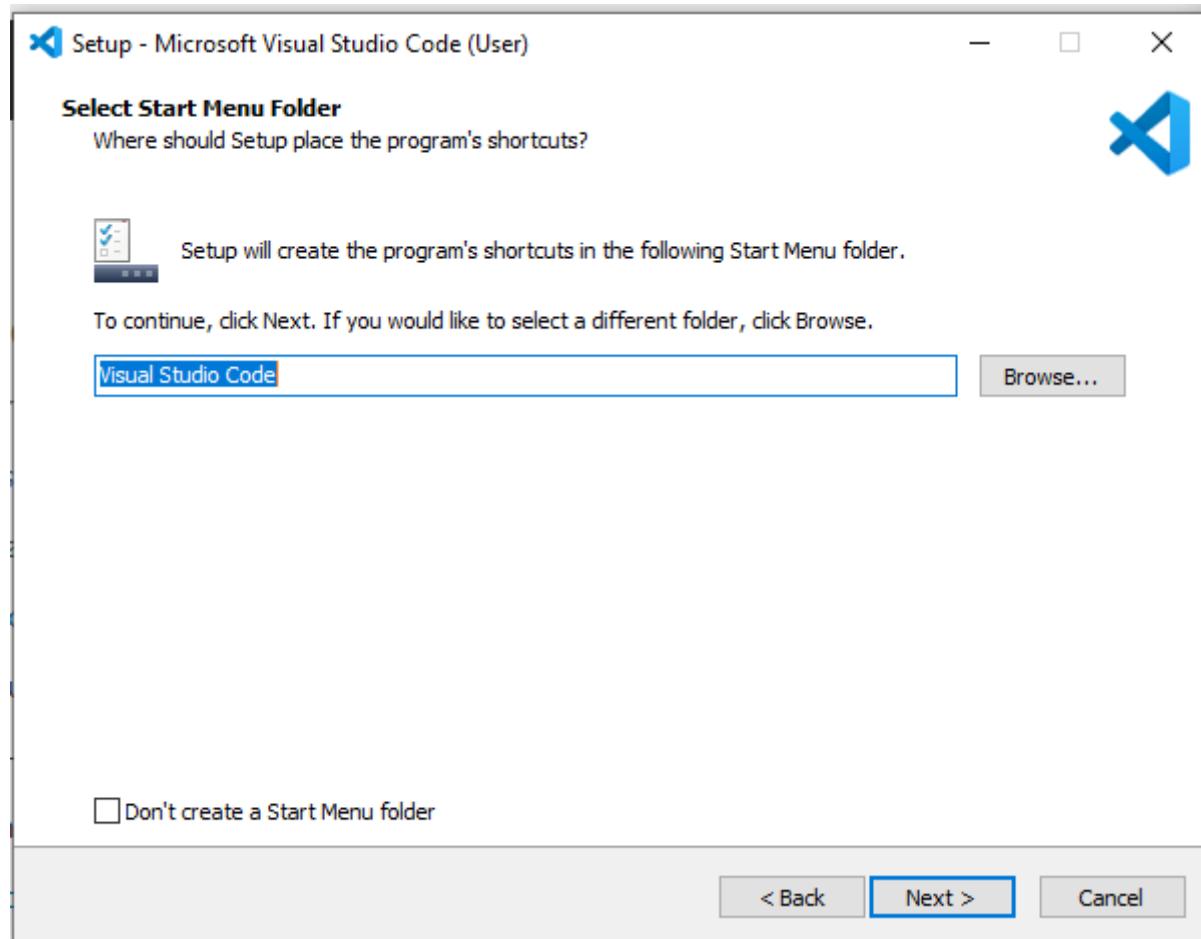
# Instalando Visual Studio Code



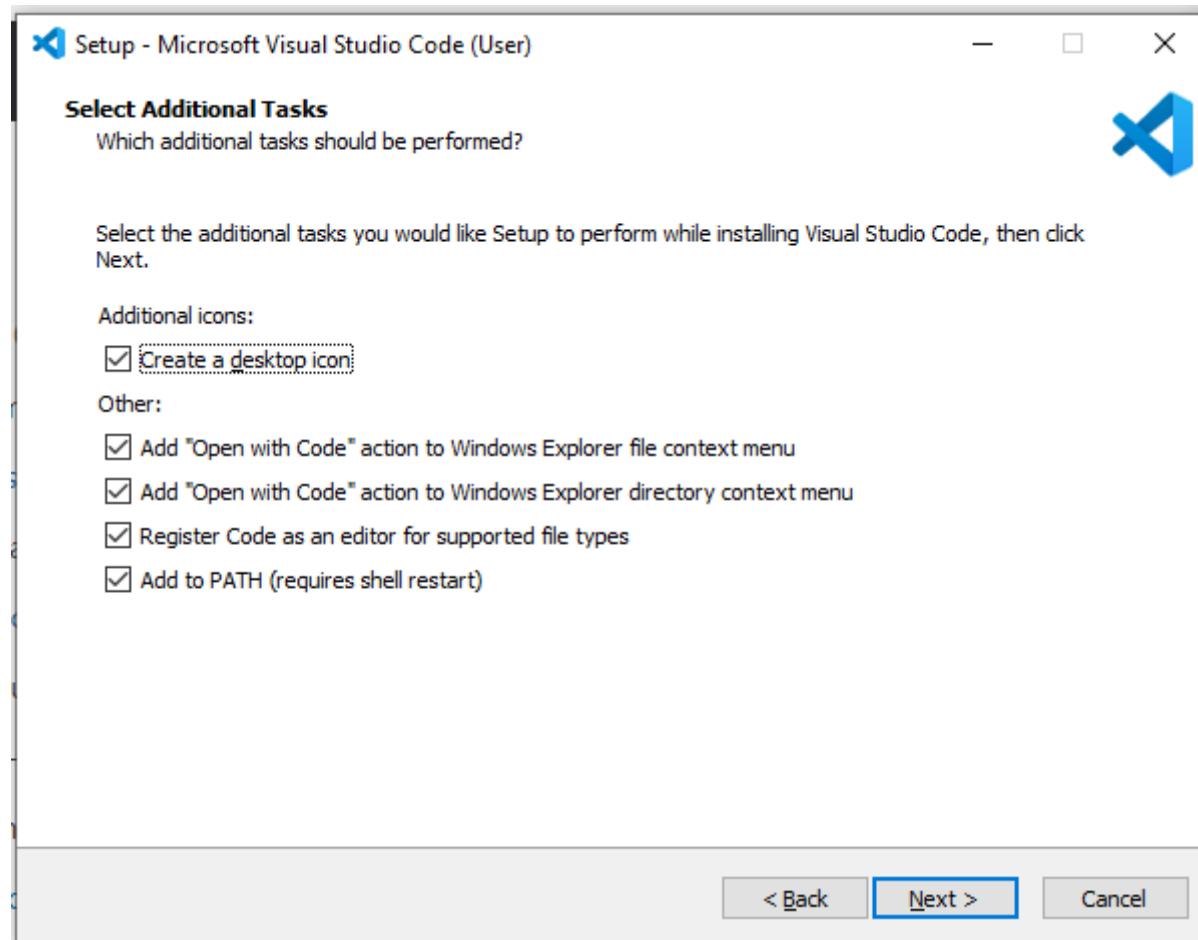
# Instalando Visual Studio Code



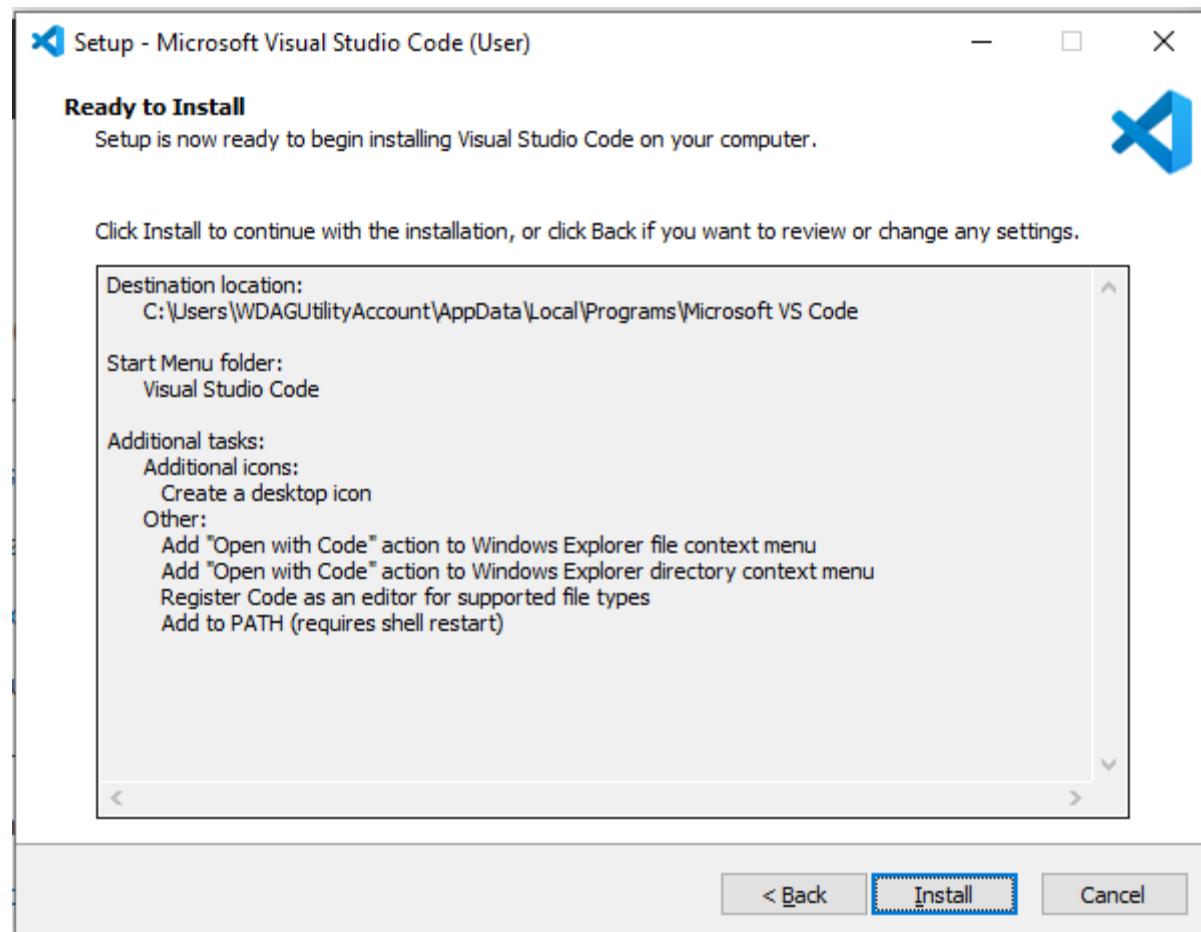
# Instalando Visual Studio Code



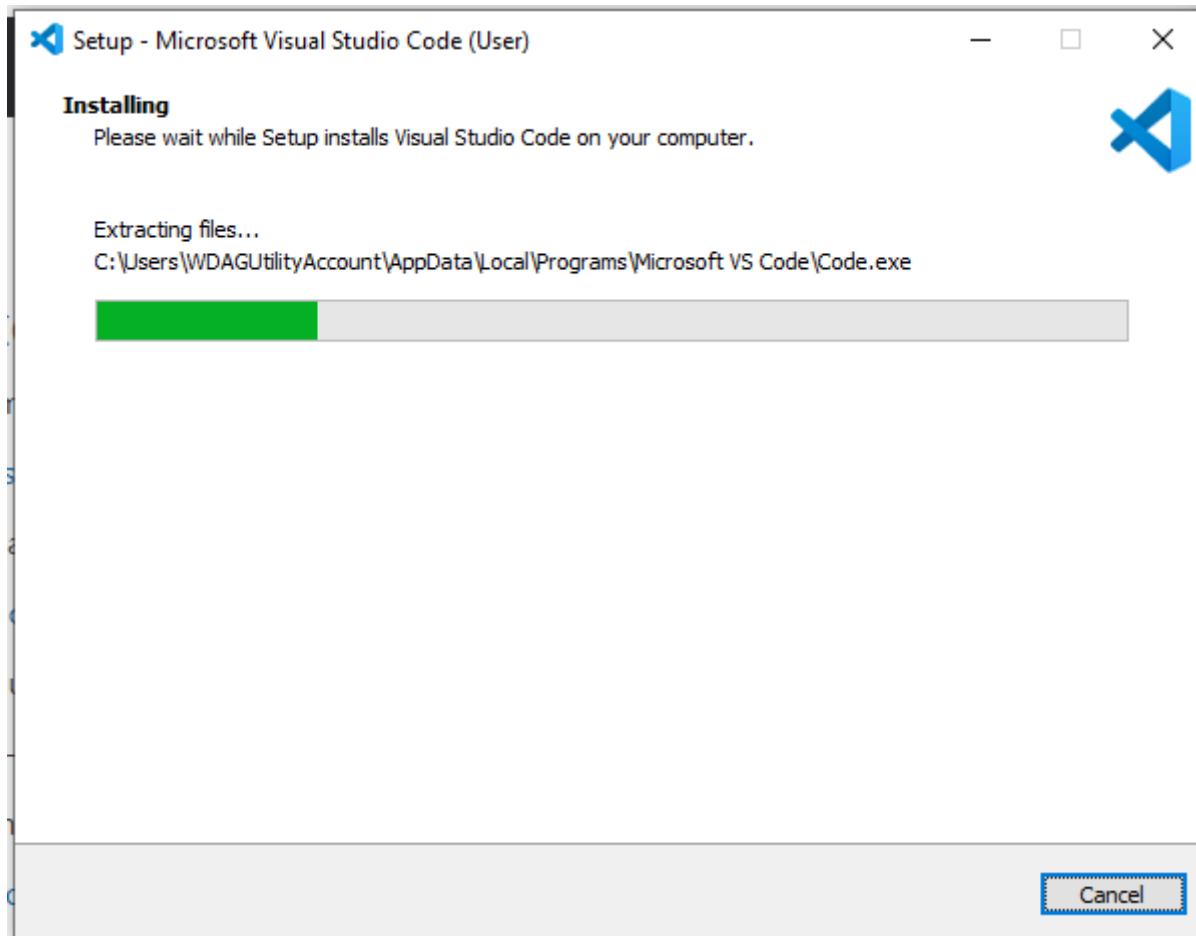
# Instalando Visual Studio Code



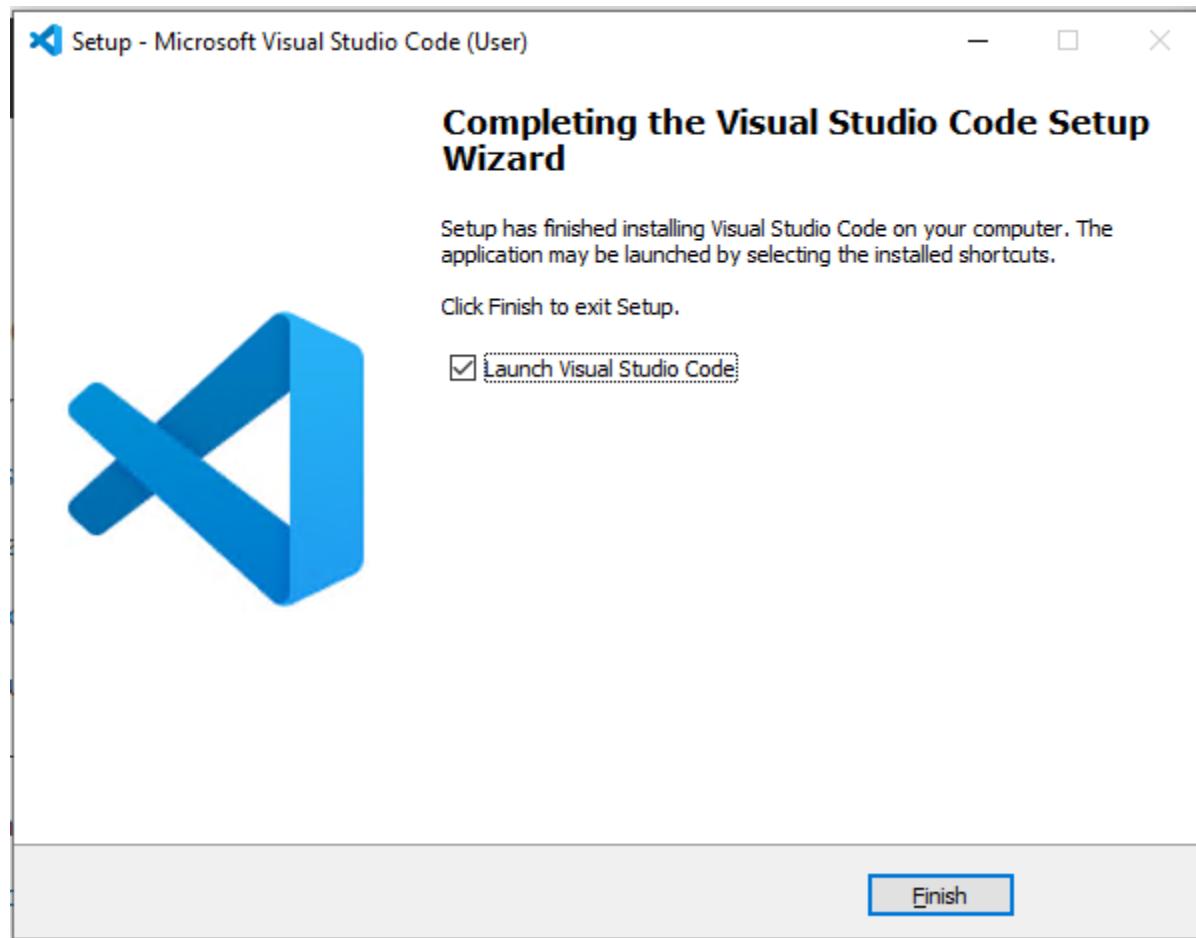
# Instalando Visual Studio Code



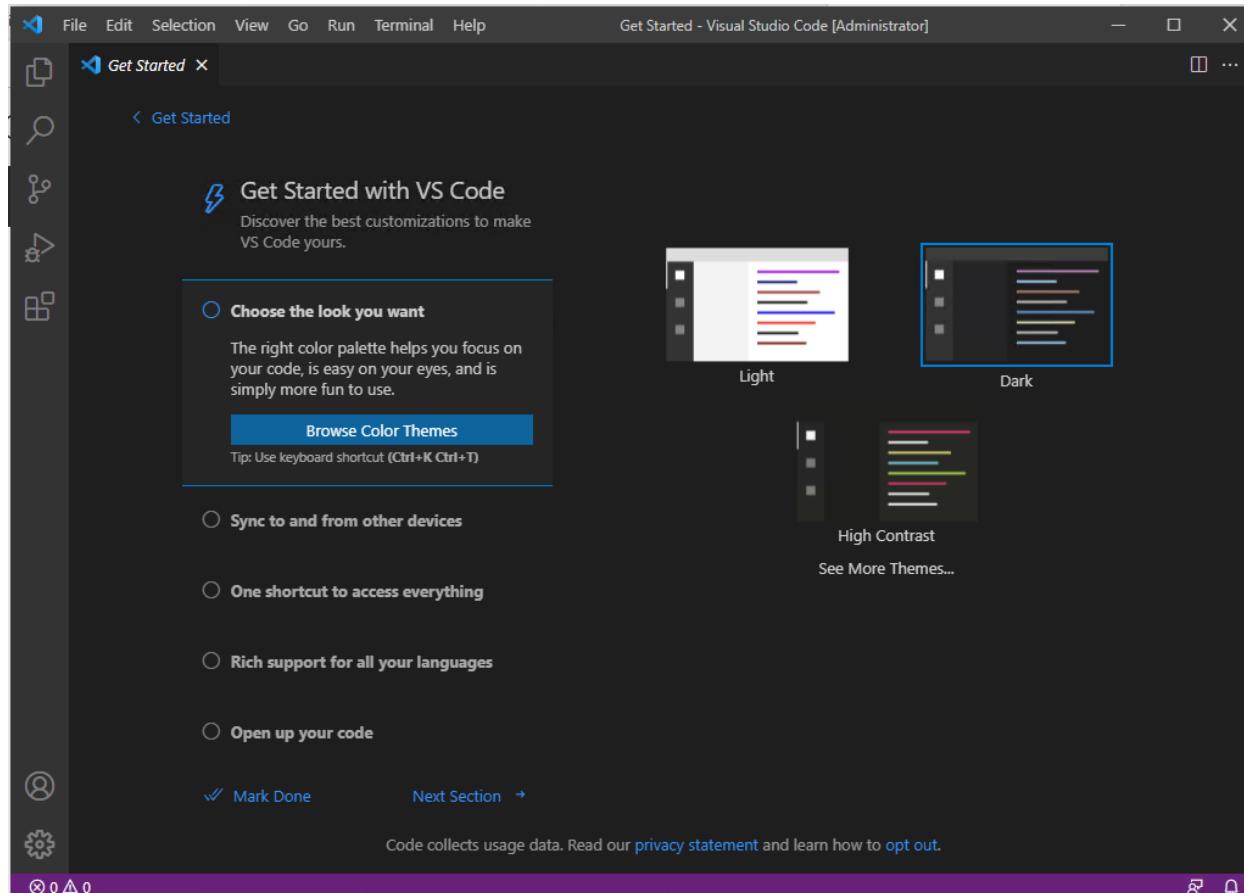
# Instalando Visual Studio Code



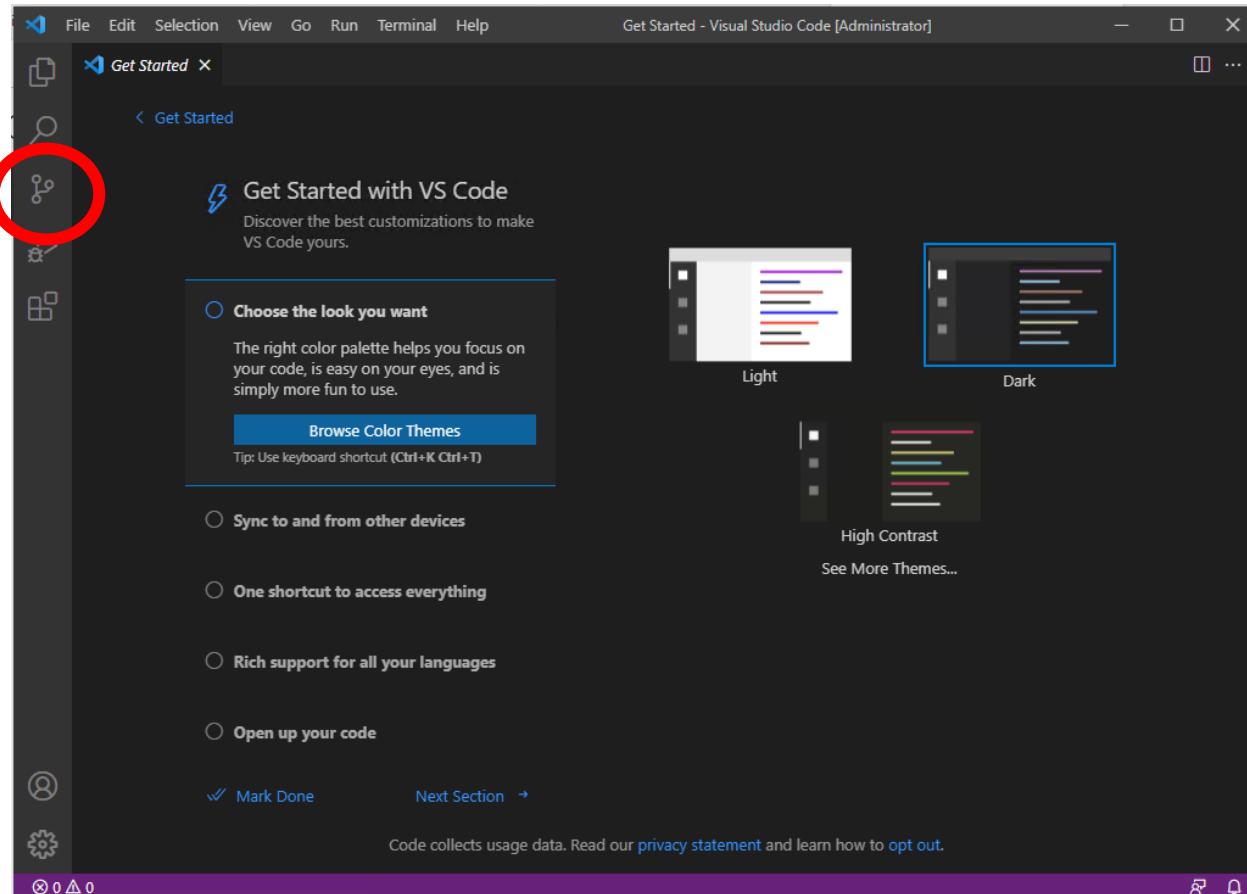
# Instalando Visual Studio Code



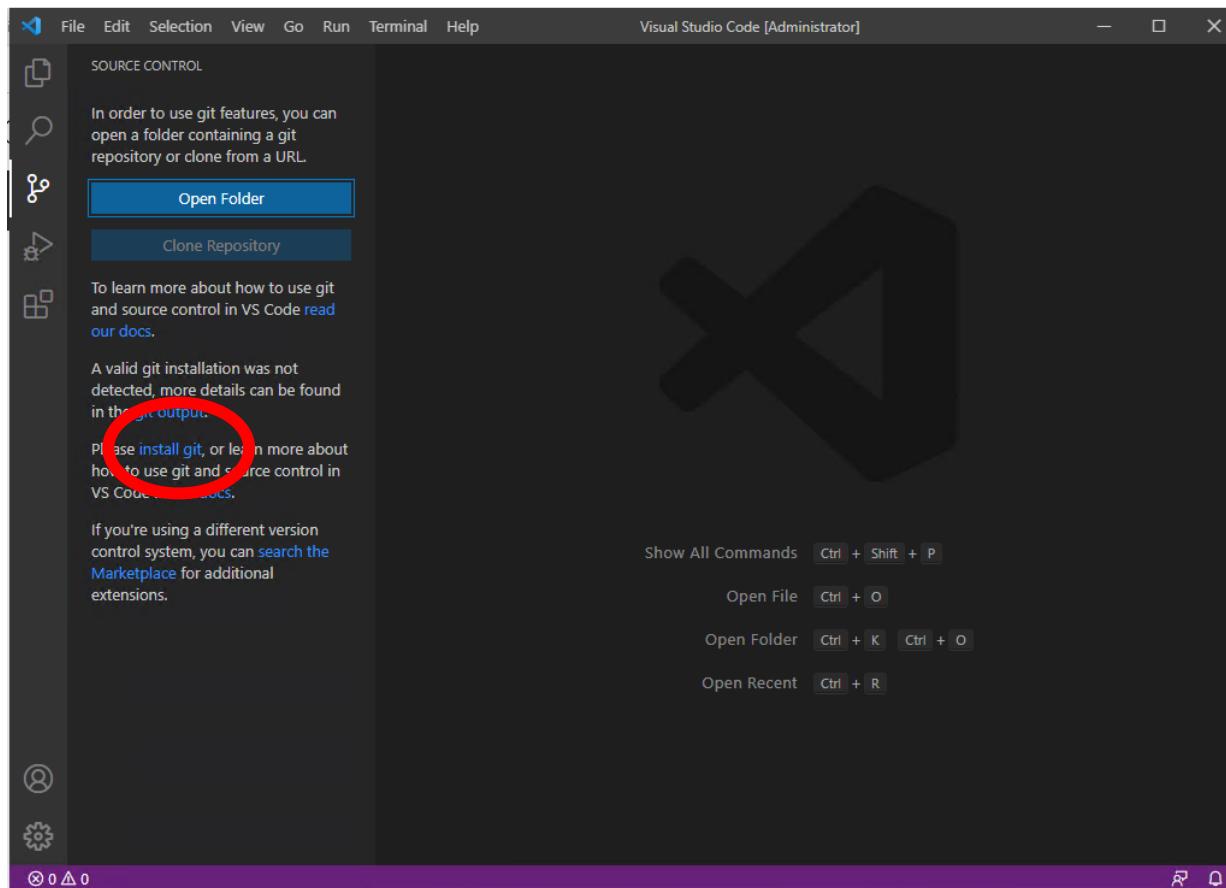
# Instalando Visual Studio Code



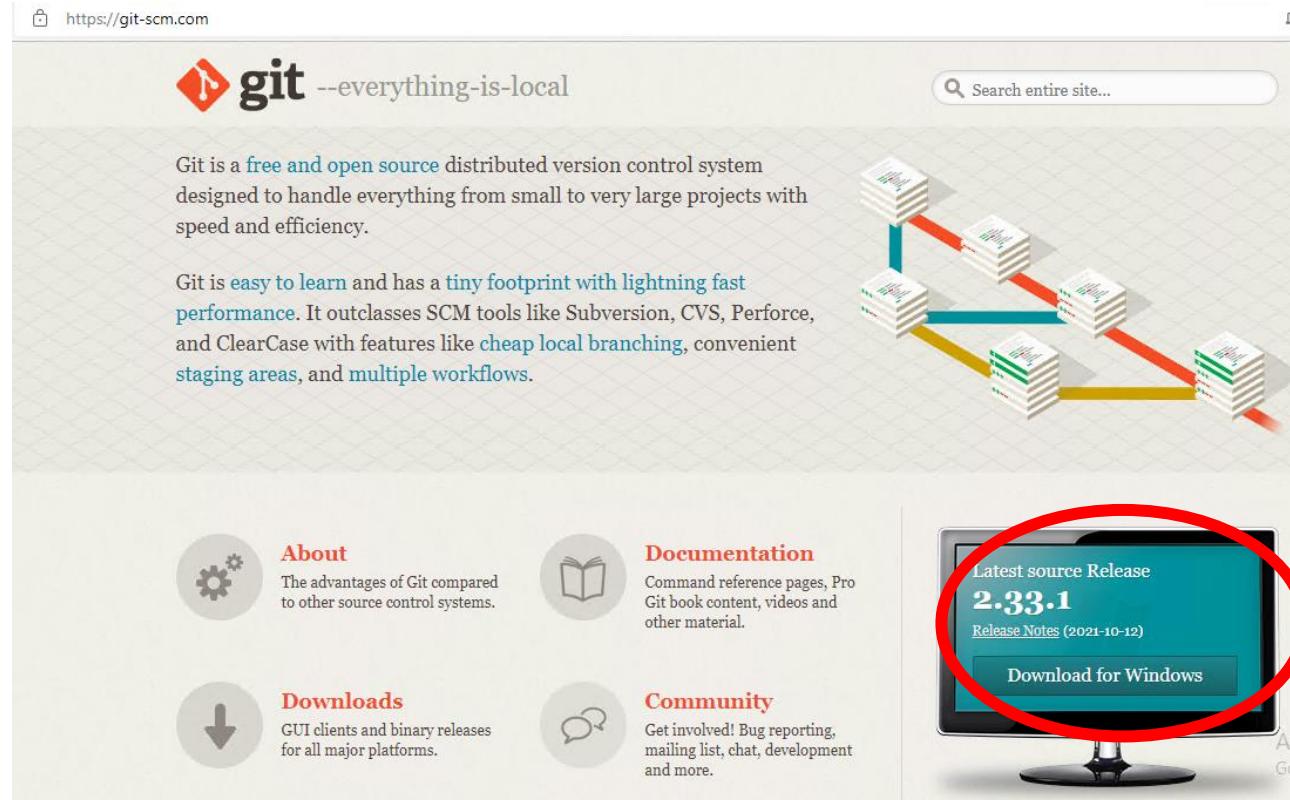
# ¿Cómo configurar mi usuario y mi contraseña?



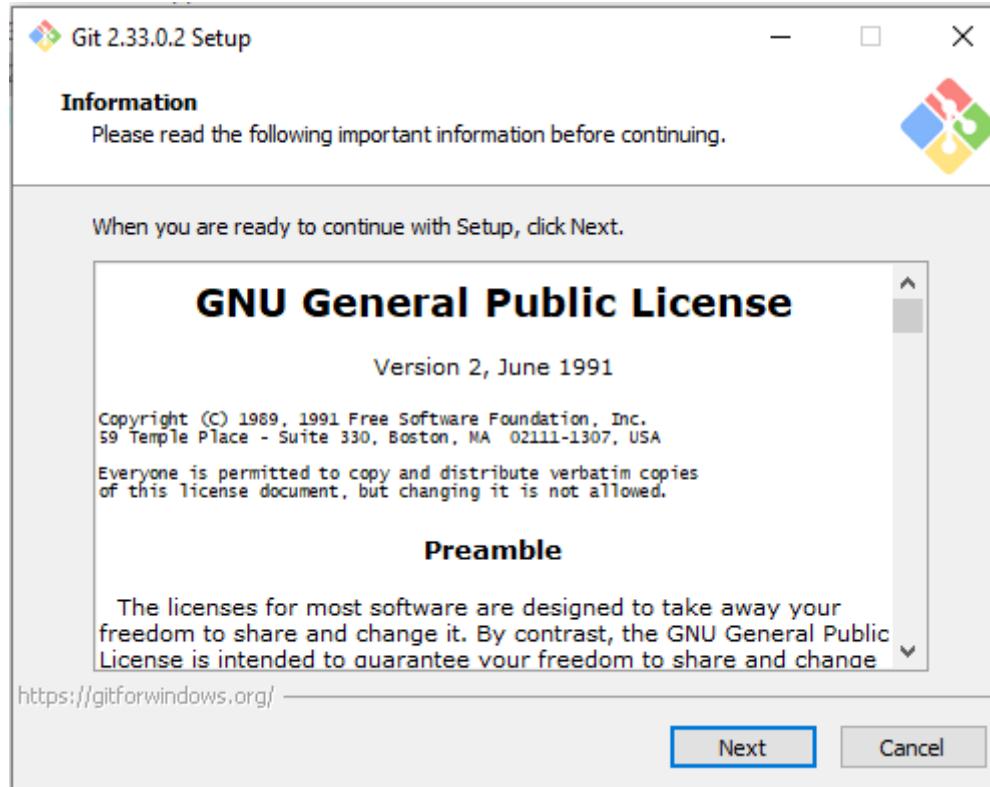
# ¿Cómo configurar mi usuario y mi contraseña?



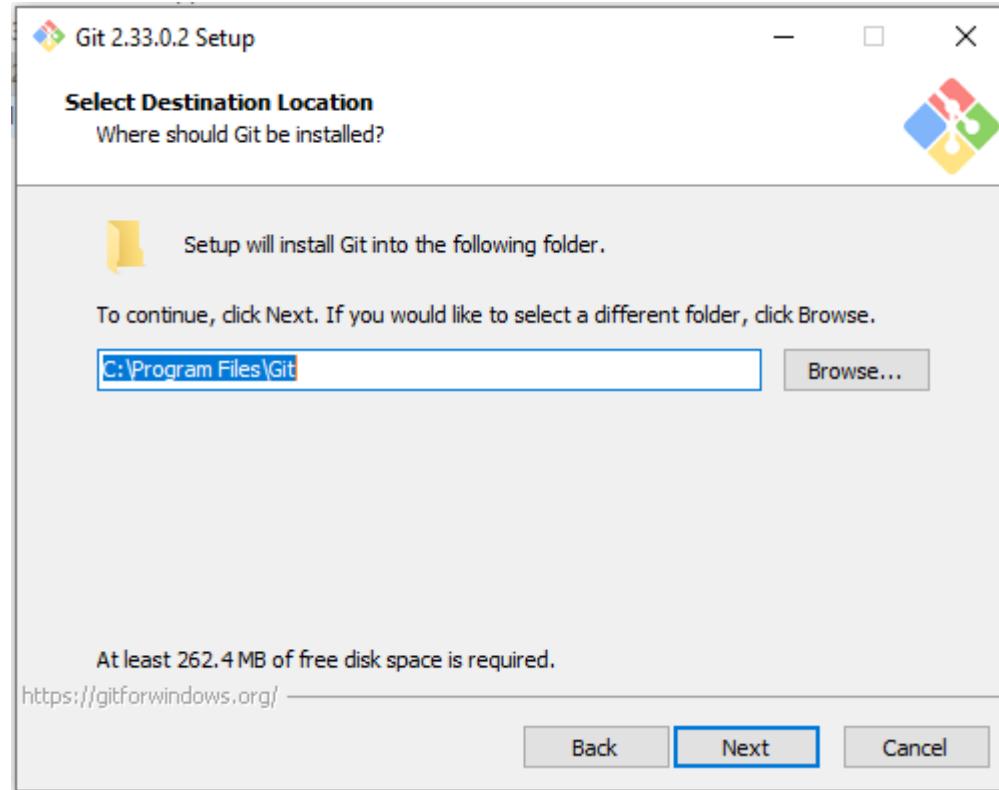
# ¿Cómo configurar mi usuario y mi contraseña?



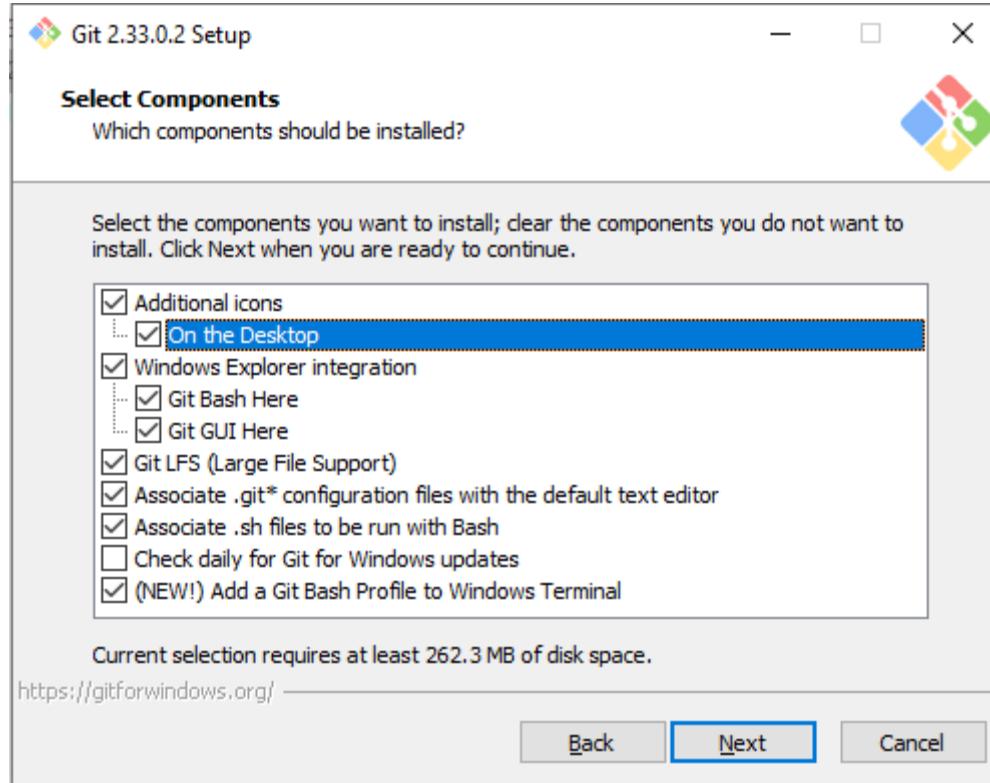
# ¿Cómo configurar mi usuario y mi contraseña?



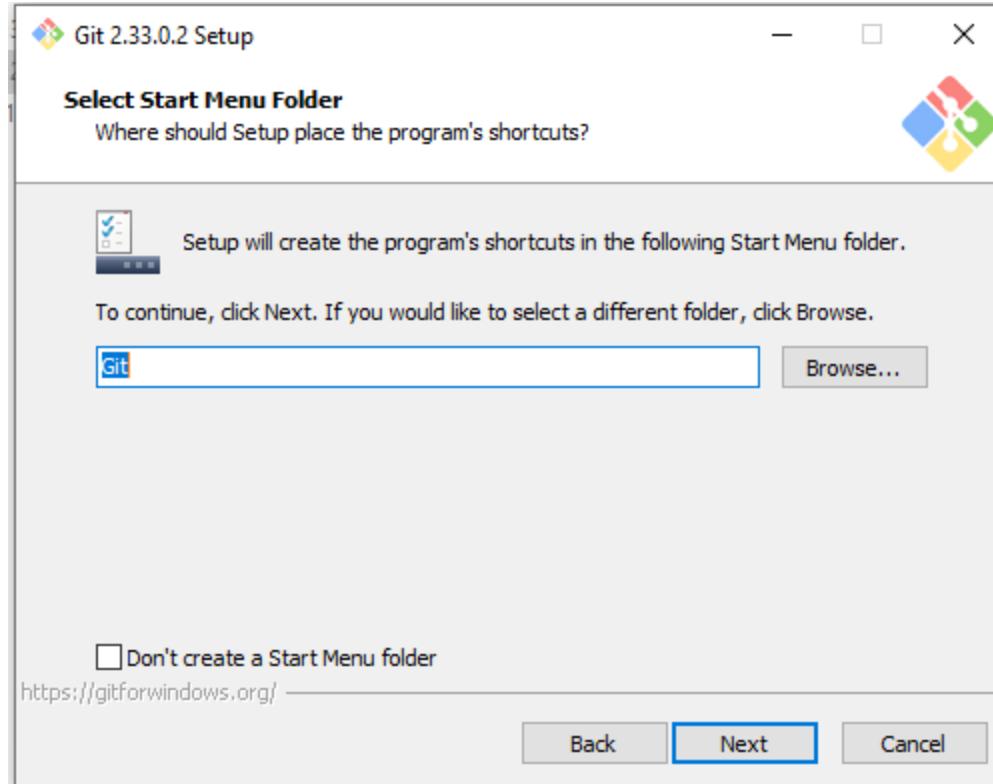
# ¿Cómo configurar mi usuario y mi contraseña?



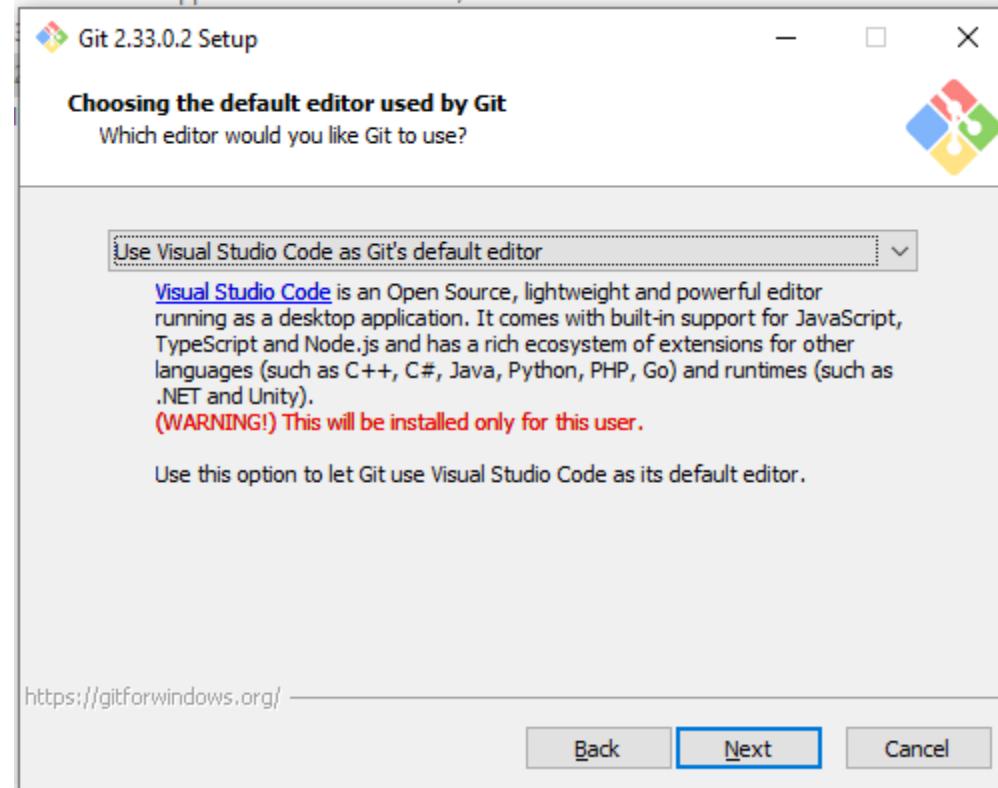
# ¿Cómo configurar mi usuario y mi contraseña?



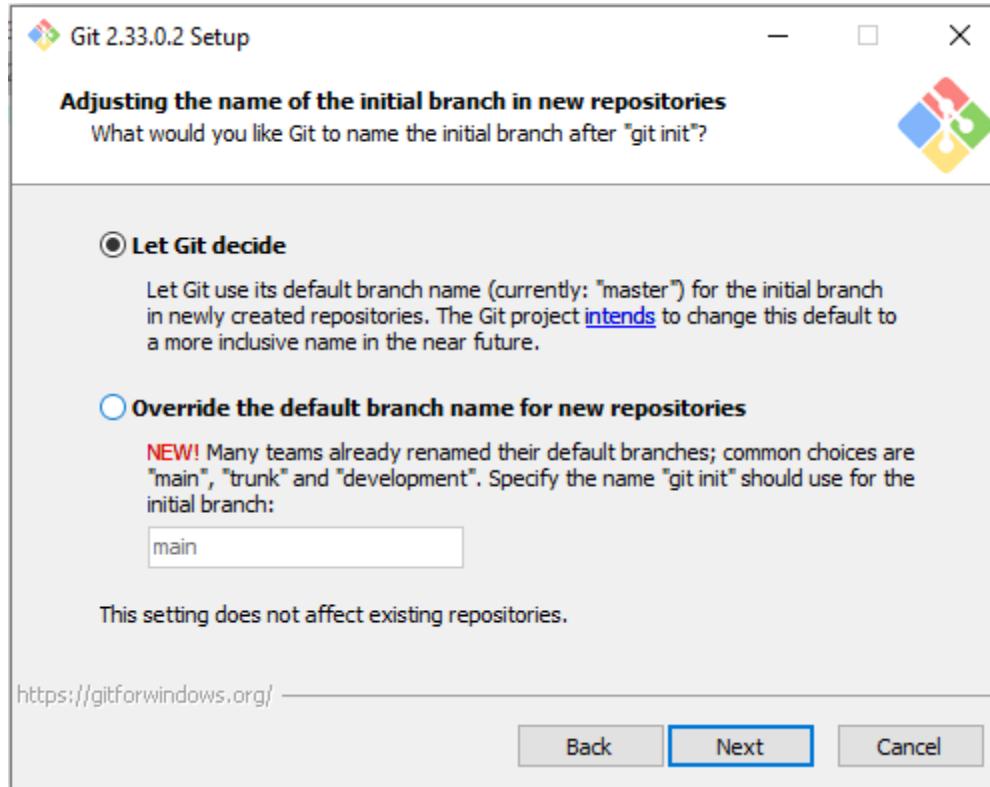
# ¿Cómo configurar mi usuario y mi contraseña?



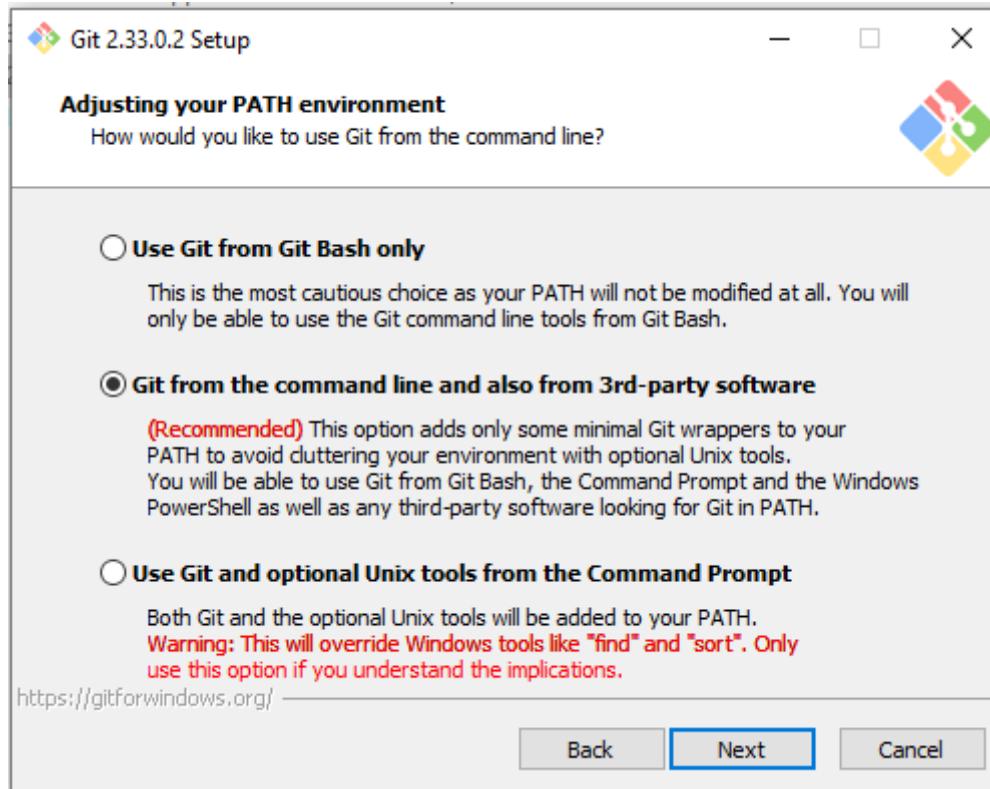
# ¿Cómo configurar mi usuario y mi contraseña?



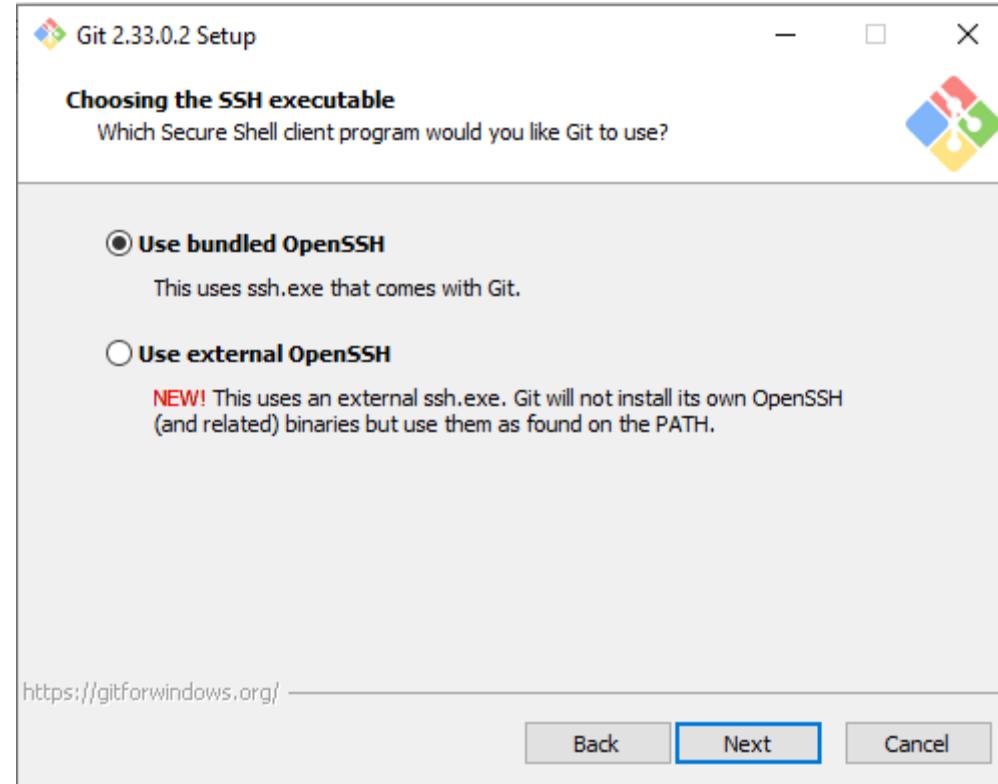
# ¿Cómo configurar mi usuario y mi contraseña?



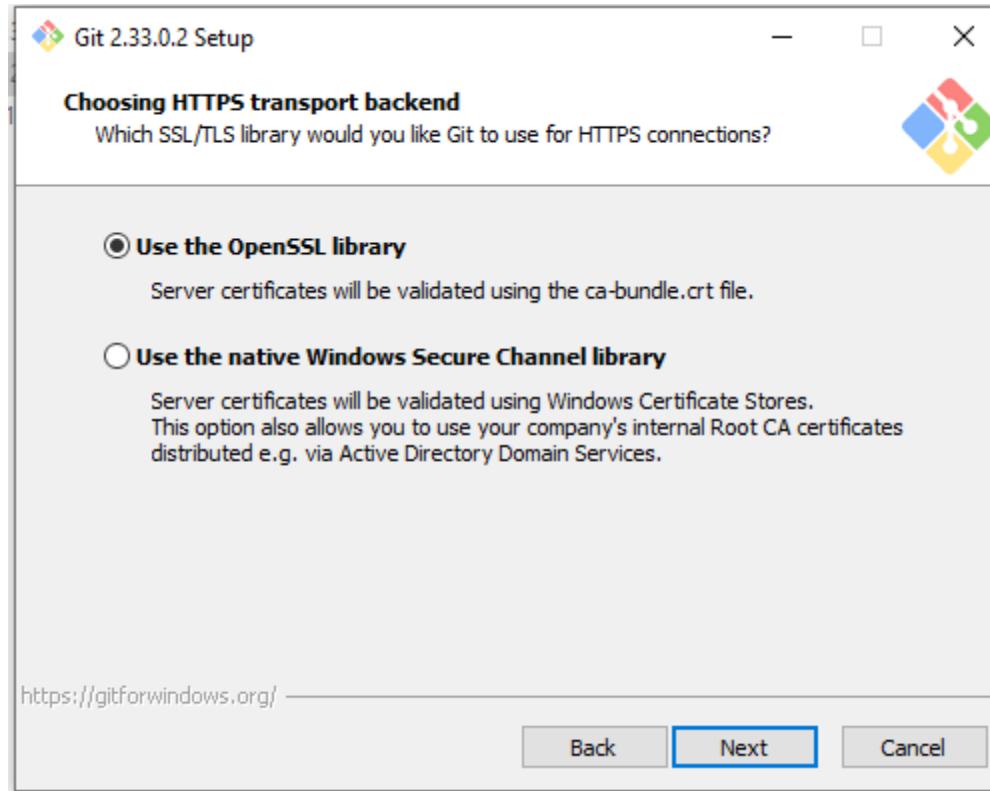
# ¿Cómo configurar mi usuario y mi contraseña?



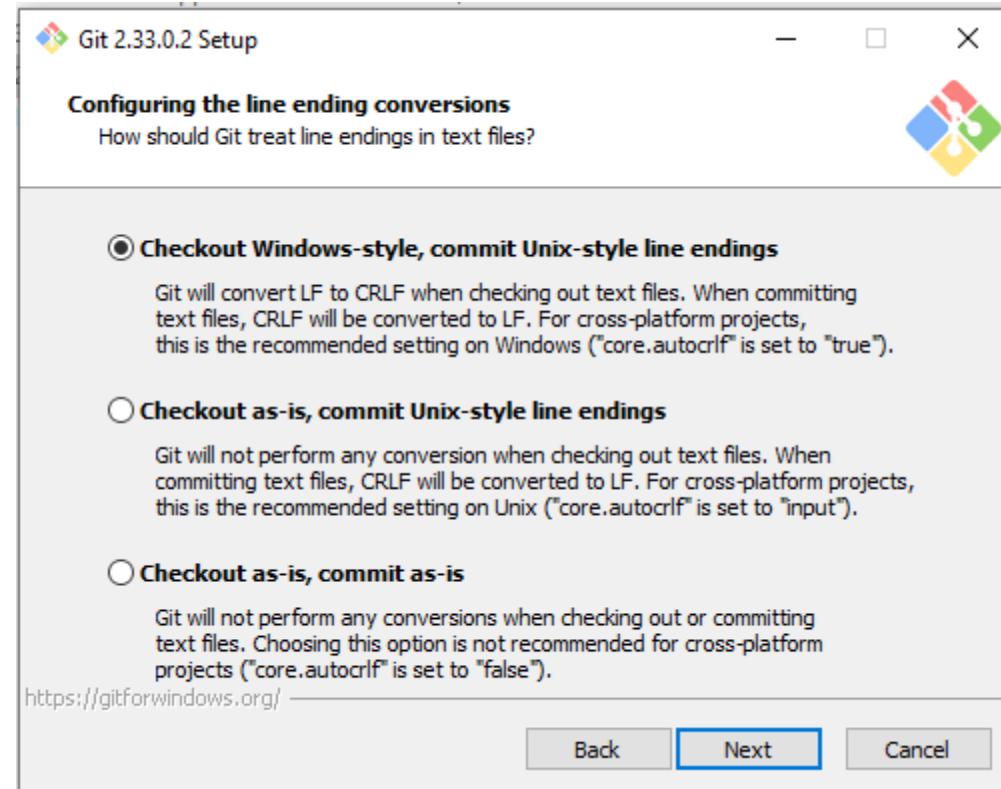
# ¿Cómo configurar mi usuario y mi contraseña?



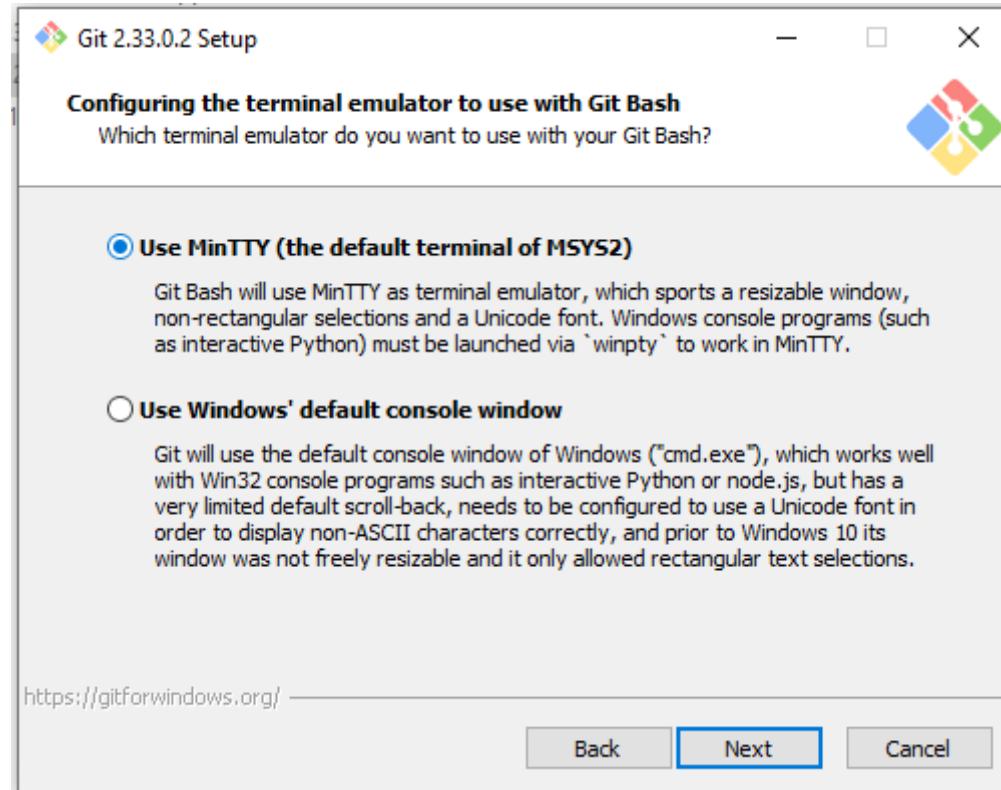
# ¿Cómo configurar mi usuario y mi contraseña?



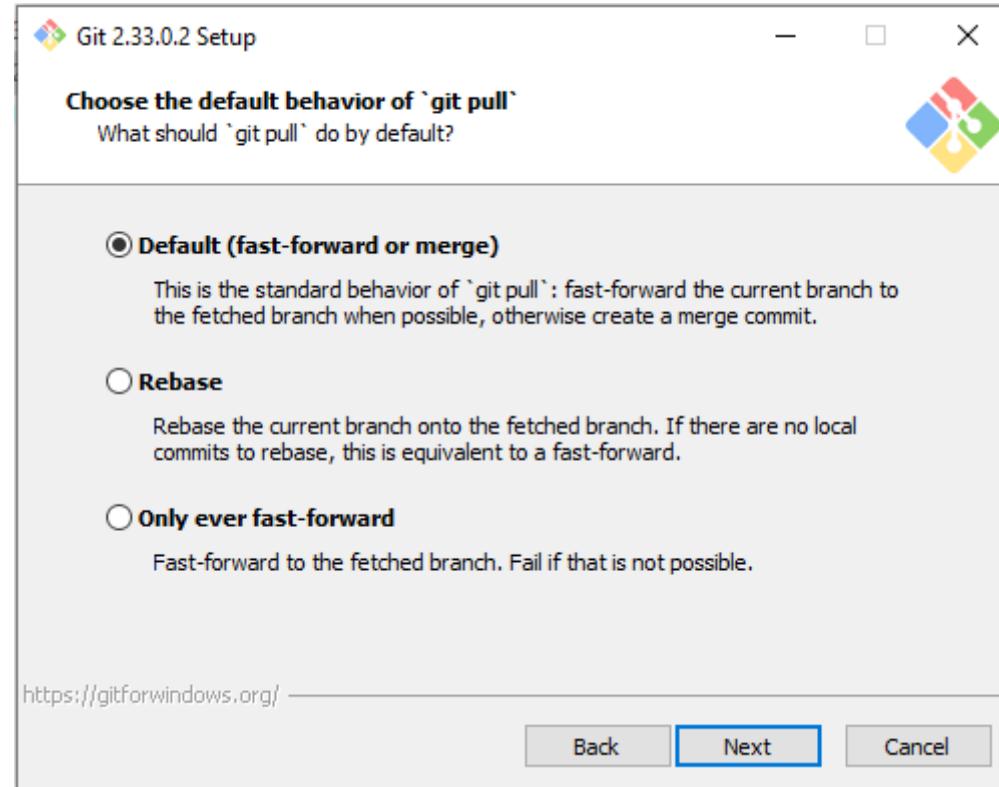
# ¿Cómo configurar mi usuario y mi contraseña?



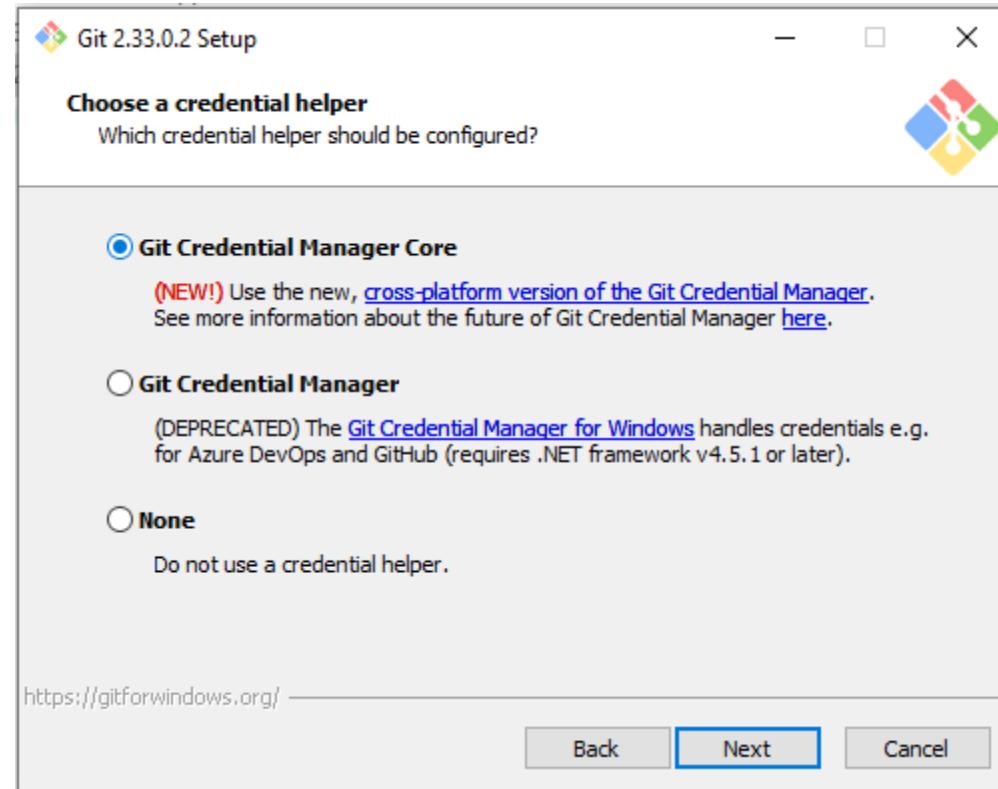
# ¿Cómo configurar mi usuario y mi contraseña?



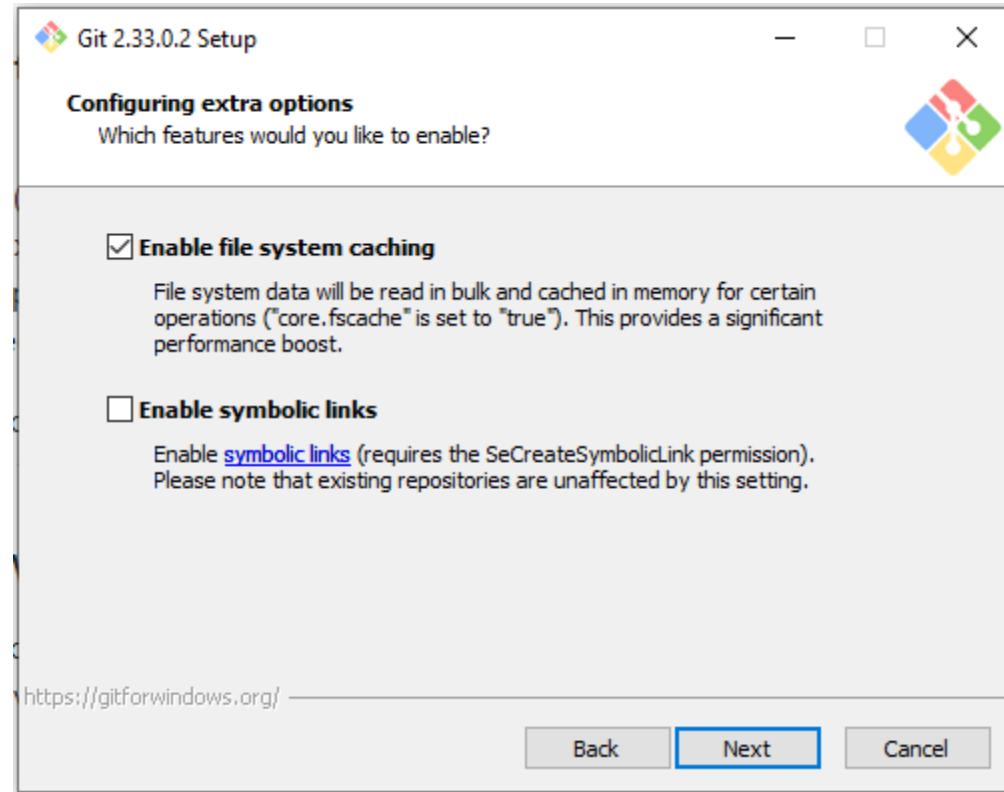
# ¿Cómo configurar mi usuario y mi contraseña?



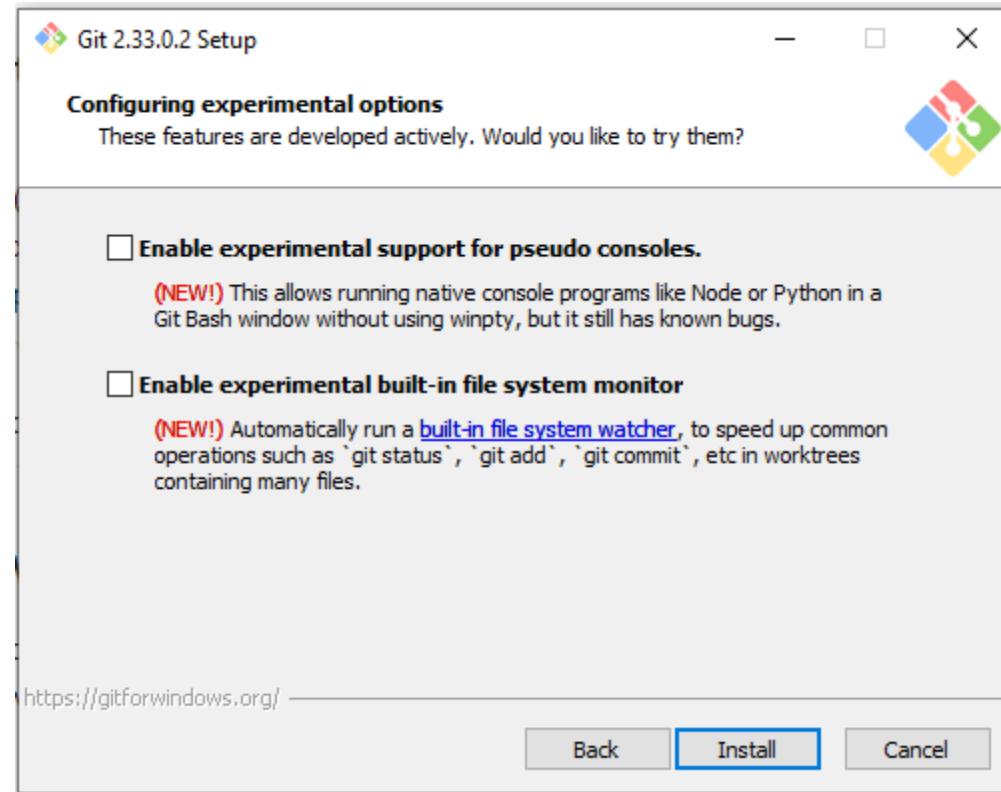
# ¿Cómo configurar mi usuario y mi contraseña?



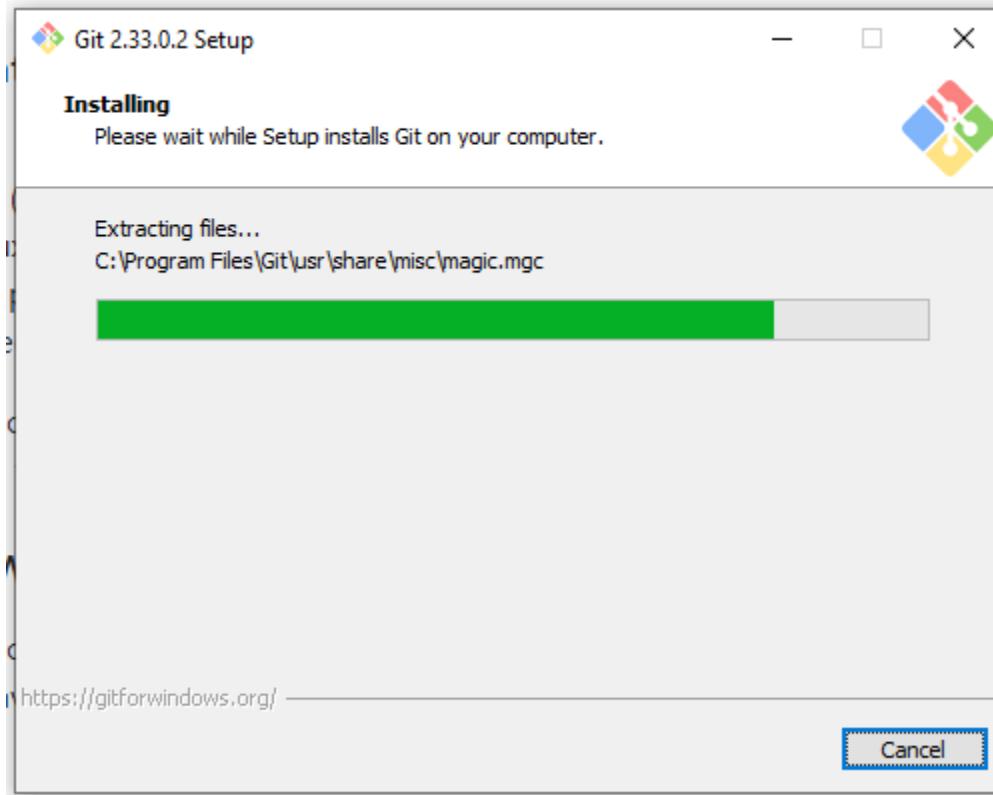
# ¿Cómo configurar mi usuario y mi contraseña?



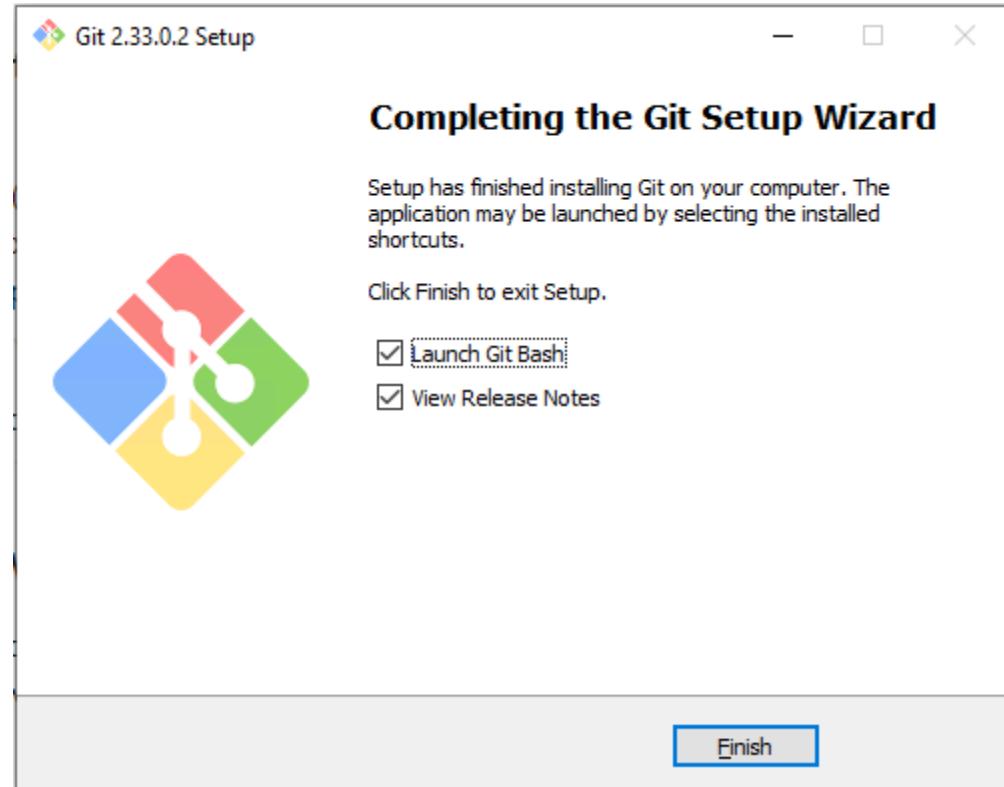
# ¿Cómo configurar mi usuario y mi contraseña?



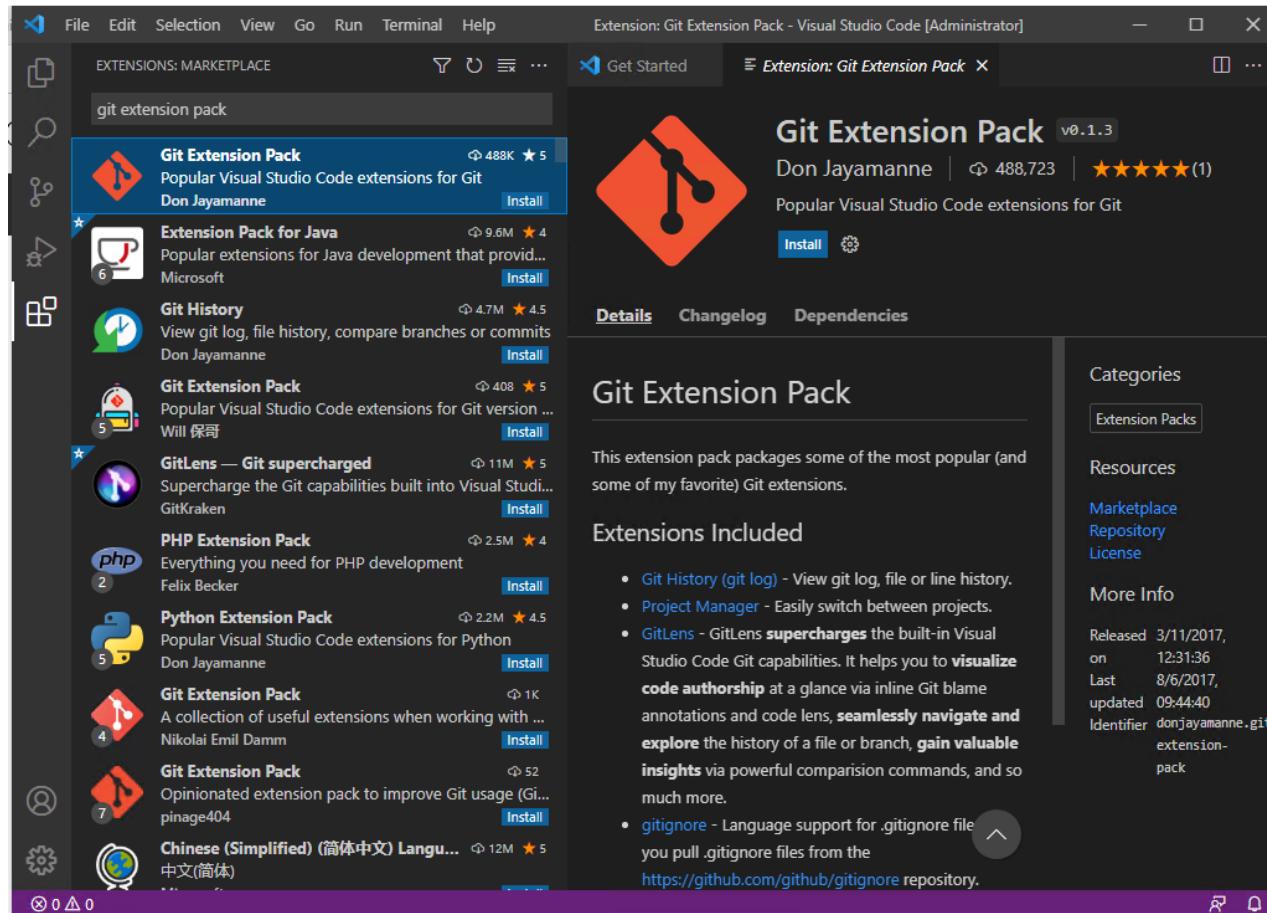
# ¿Cómo configurar mi usuario y mi contraseña?



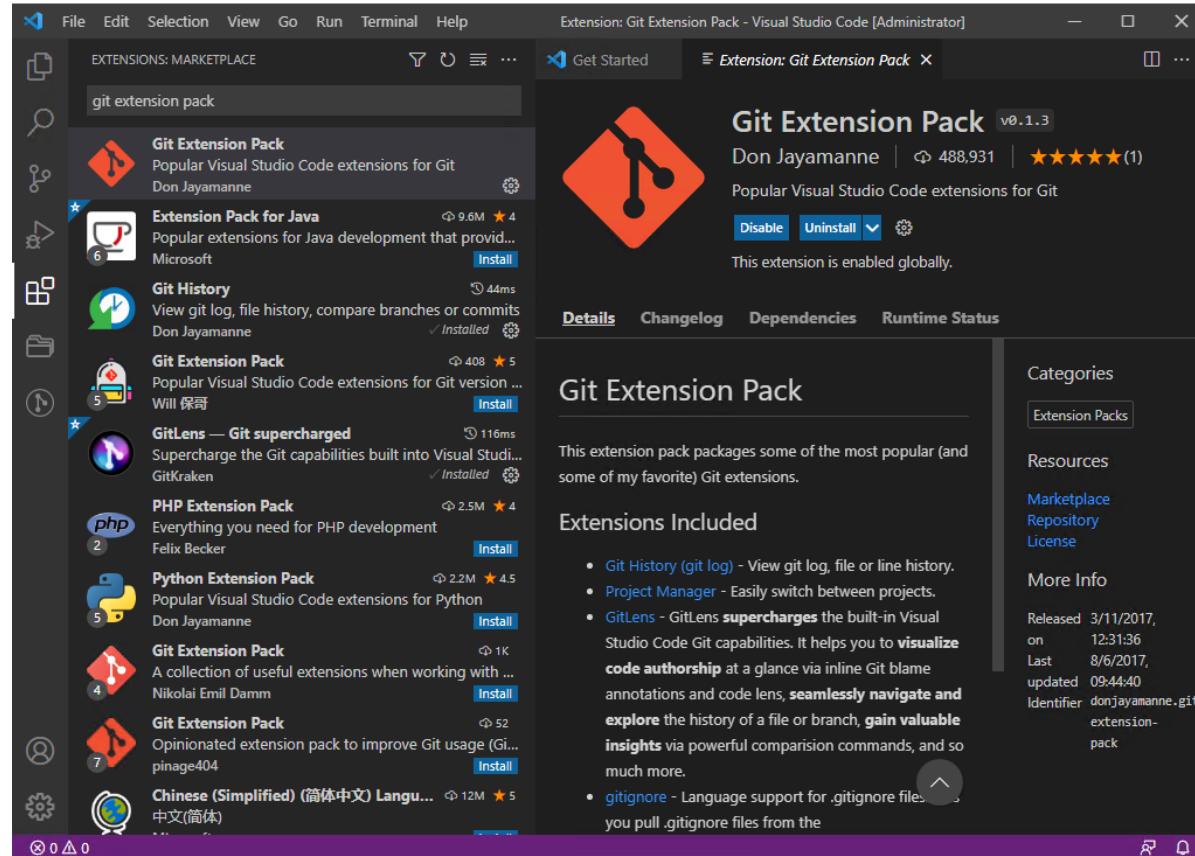
# ¿Cómo configurar mi usuario y mi contraseña?



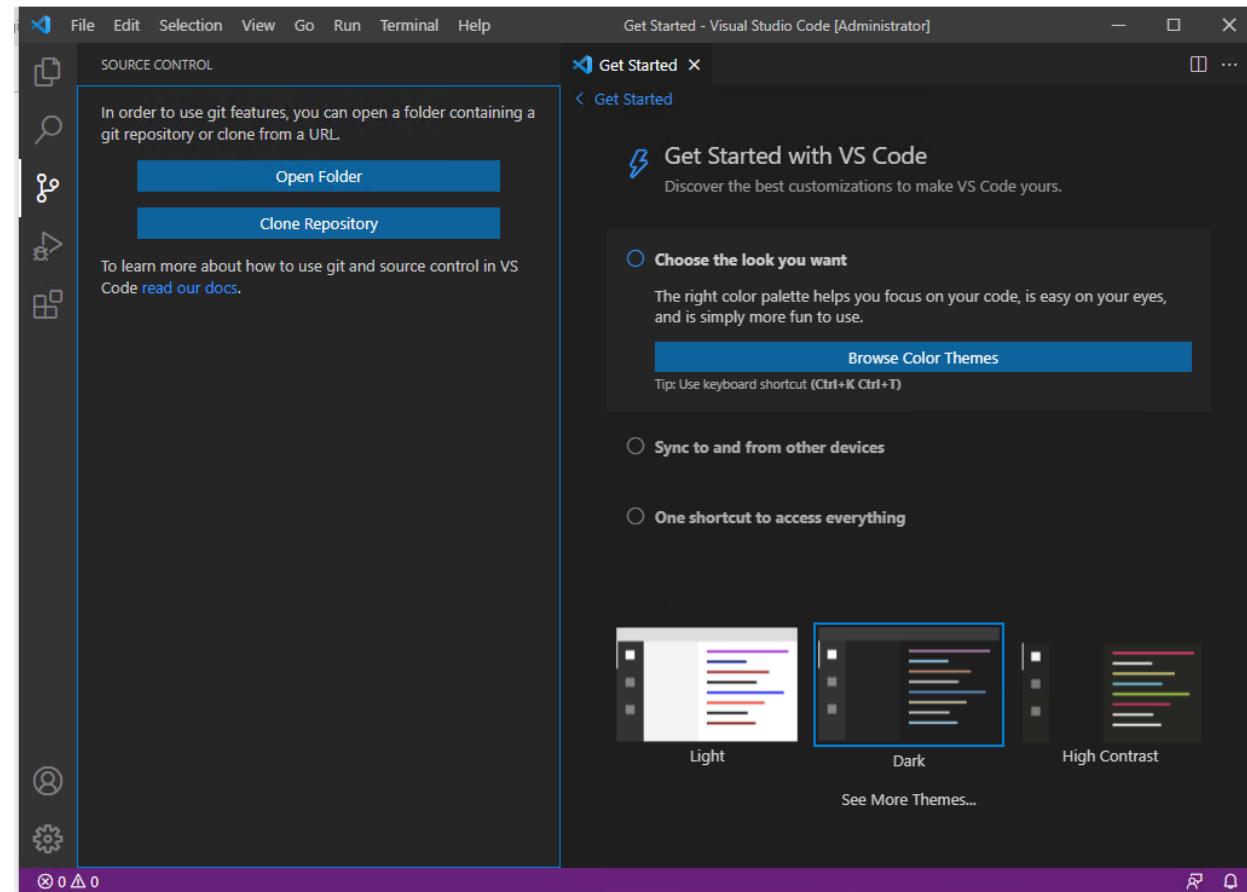
# ¿Cómo configurar mi usuario y mi contraseña?



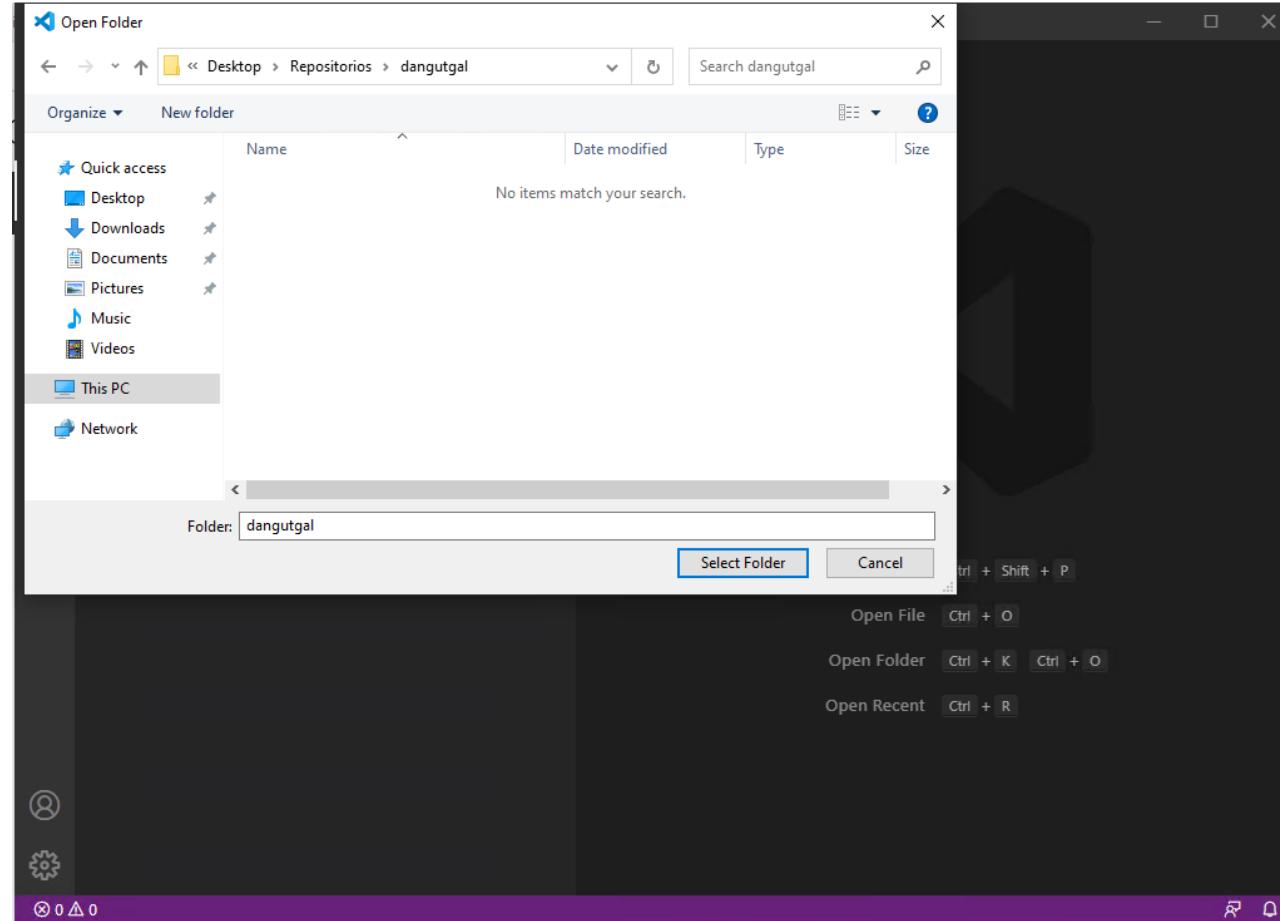
# ¿Cómo configurar mi usuario y mi contraseña?



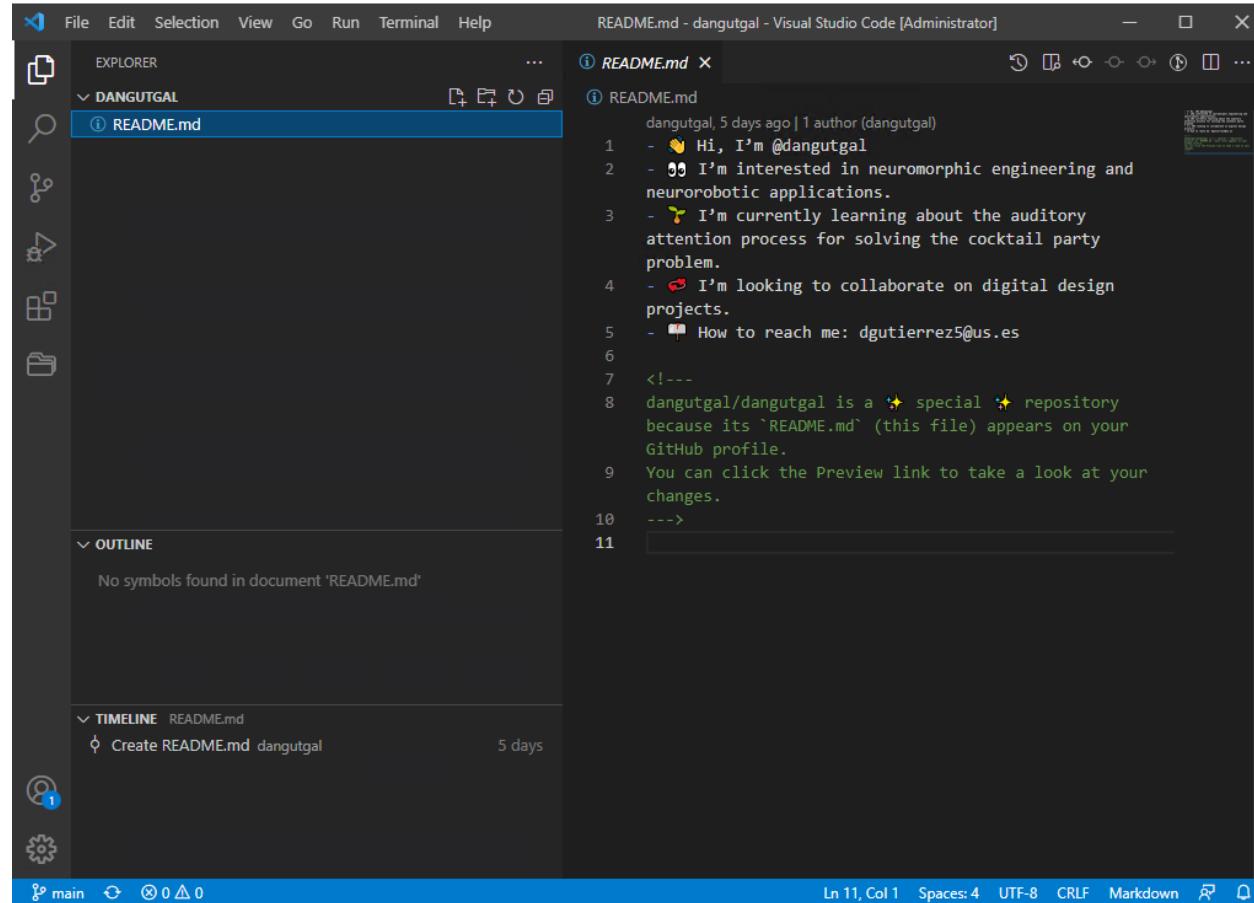
# ¿Cómo configurar mi usuario y mi contraseña?



# ¿Cómo configurar mi usuario y mi contraseña?



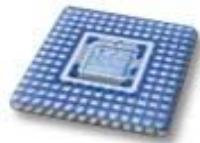
# ¿Cómo configurar mi usuario y mi contraseña?



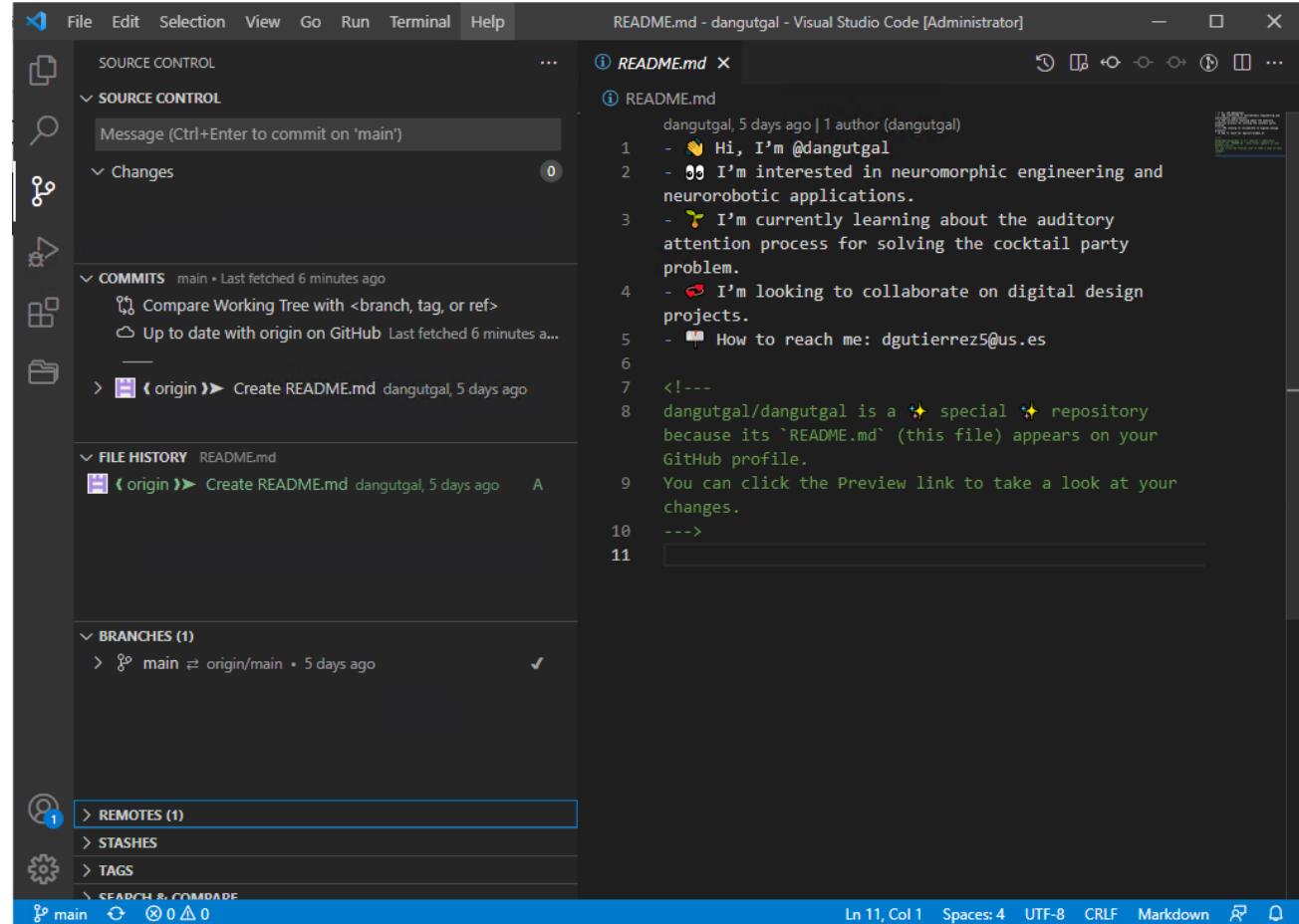
The screenshot shows the Visual Studio Code interface with a dark theme. The left sidebar has icons for Explorer, Search, Open, and Folders. The Explorer panel shows a folder named 'DANGUTGAL' containing a file named 'README.md'. The main editor area displays the contents of 'README.md'.

```
dangutgal, 5 days ago | 1 author (dangutgal)
1 - 🌟 Hi, I'm @dangutgal
2 - 💬 I'm interested in neuromorphic engineering and
neurorobotic applications.
3 - 🎧 I'm currently learning about the auditory
attention process for solving the cocktail party
problem.
4 - 🖥️ I'm looking to collaborate on digital design
projects.
5 - 📩 How to reach me: dgutierrez5@us.es
6 -
7 <!--
8 dangutgal/dangutgal is a ✨ special ✨ repository
because its `README.md` (this file) appears on your
GitHub profile.
9 You can click the Preview link to take a look at your
changes.
10 -->
11
```

The bottom status bar shows 'main' and 'Ln 11, Col 1'.



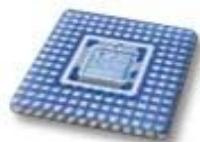
# ¿Cómo configurar mi usuario y mi contraseña?



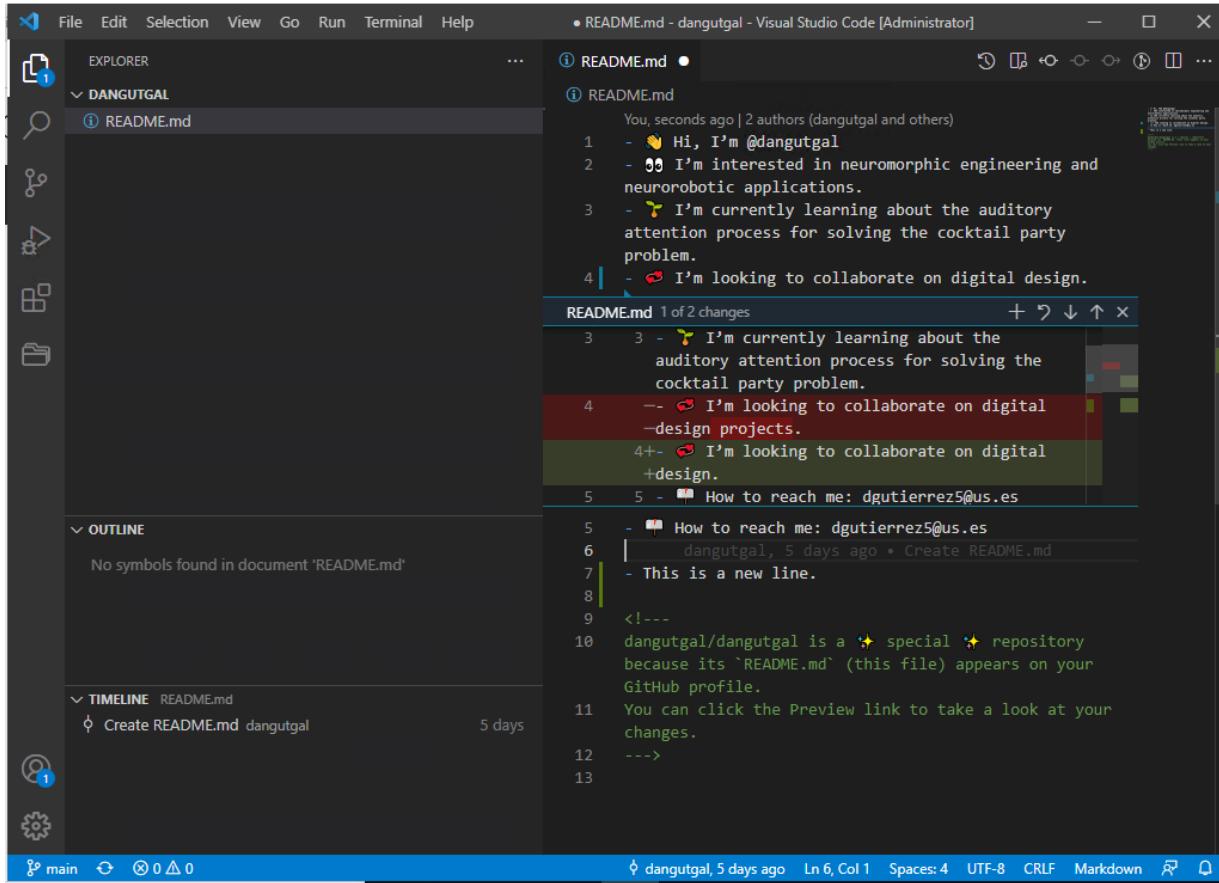
The screenshot shows the Visual Studio Code interface with a GitHub repository for a file named README.md. The repository has one commit from the user dangutgal, dated 5 days ago. The commit message is:

```
Hi, I'm @dangutgal  
I'm interested in neuromorphic engineering and  
neurorobotic applications.  
I'm currently learning about the auditory  
attention process for solving the cocktail party  
problem.  
I'm looking to collaborate on digital design  
projects.  
How to reach me: dgutierrez5@us.es
```

The repository also has one file history entry for the README.md file, which was created by the user dangutgal 5 days ago. There is one branch named main, which is up-to-date with origin/main. The REMOTES section shows one remote named origin.



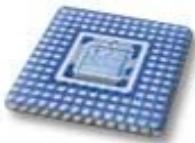
# ¿Cómo configurar mi usuario y mi contraseña?



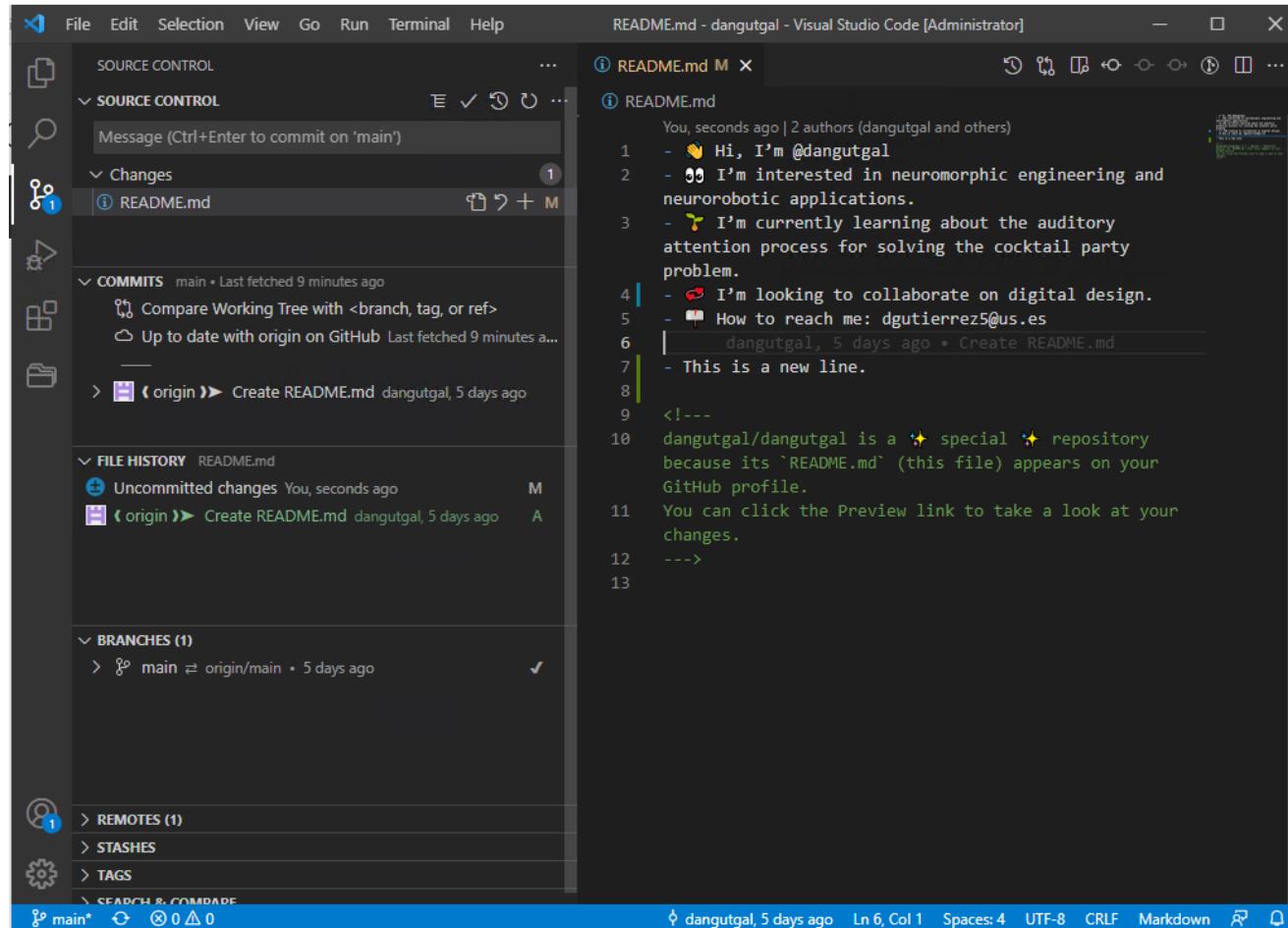
The screenshot shows a Visual Studio Code window with a dark theme. The title bar reads "README.md - dangutgal - Visual Studio Code [Administrator]". The left sidebar has sections for "EXPLORER" (showing a folder "DANGUTGAL" with "README.md"), "OUTLINE" (empty), and "TIMELINE" (listing a single commit: "Create README.md" by "dangutgal" 5 days ago). The main editor area displays a GitHub Markdown file named "README.md". The content of the file is:

```
You, seconds ago | 2 authors (dangutgal and others)
1 - 🌟 Hi, I'm @dangutgal
2 - 🌟 I'm interested in neuromorphic engineering and
neurorobotic applications.
3 - 🌟 I'm currently learning about the auditory
attention process for solving the cocktail party
problem.
4 - 🌟 I'm looking to collaborate on digital design.

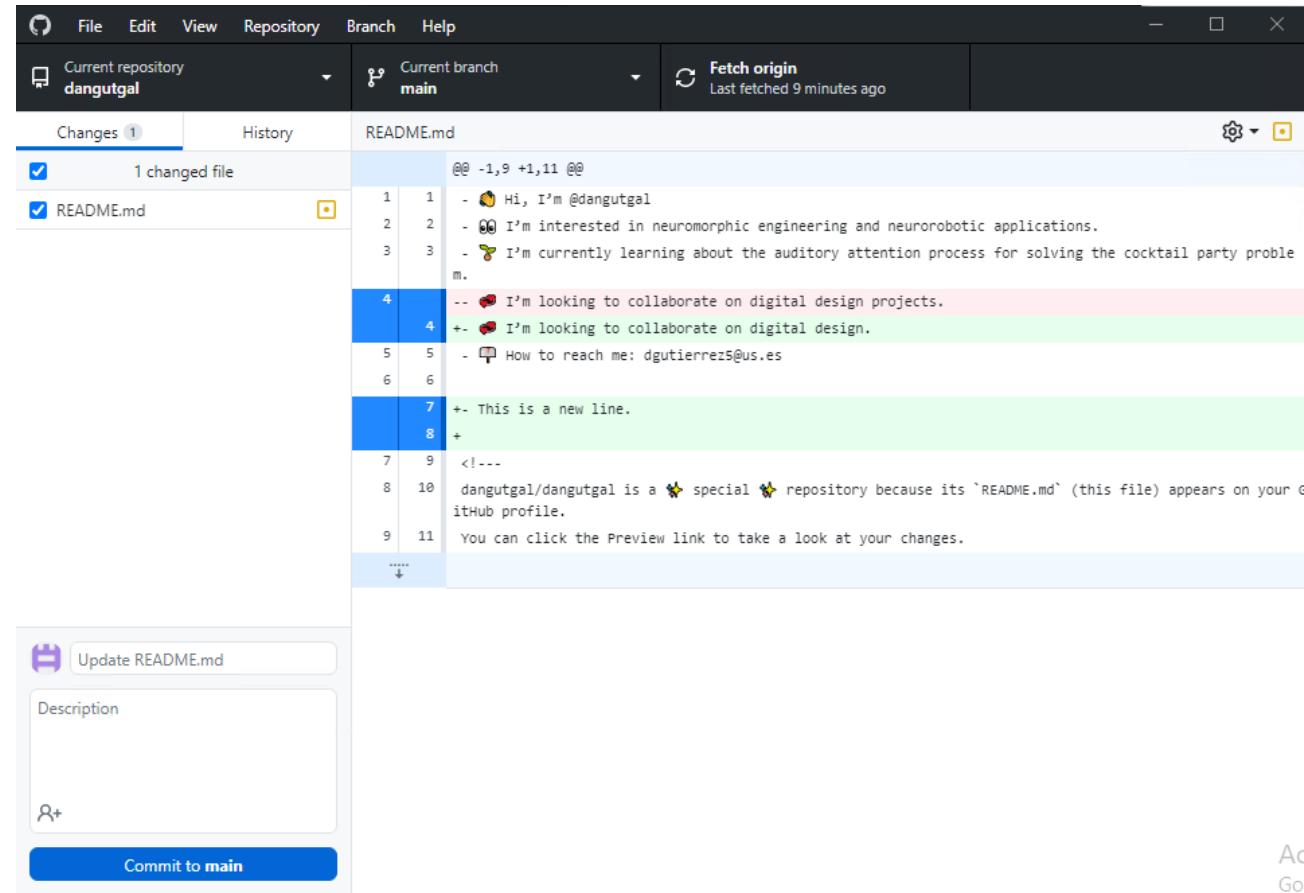
README.md 1 of 2 changes
3 - 🌟 I'm currently learning about the
auditory attention process for solving the
cocktail party problem.
4 - 🌟 I'm looking to collaborate on digital
design projects.
4+- 🌟 I'm looking to collaborate on digital
+design.
5 - 🌟 How to reach me: dgutierrez5@us.es
5 - 🌟 How to reach me: dgutierrez5@us.es
6 | dangutgal, 5 days ago * Create README.md
7 - This is a new line.
8
9 <!--
10 dangutgal/dangutgal is a ✨ special ✨ repository
because its `README.md` (this file) appears on your
GitHub profile.
11 You can click the Preview link to take a look at your
changes.
12 -->
13
```



# ¿Cómo configurar mi usuario y mi contraseña?



# ¿Cómo configurar mi usuario y mi contraseña?



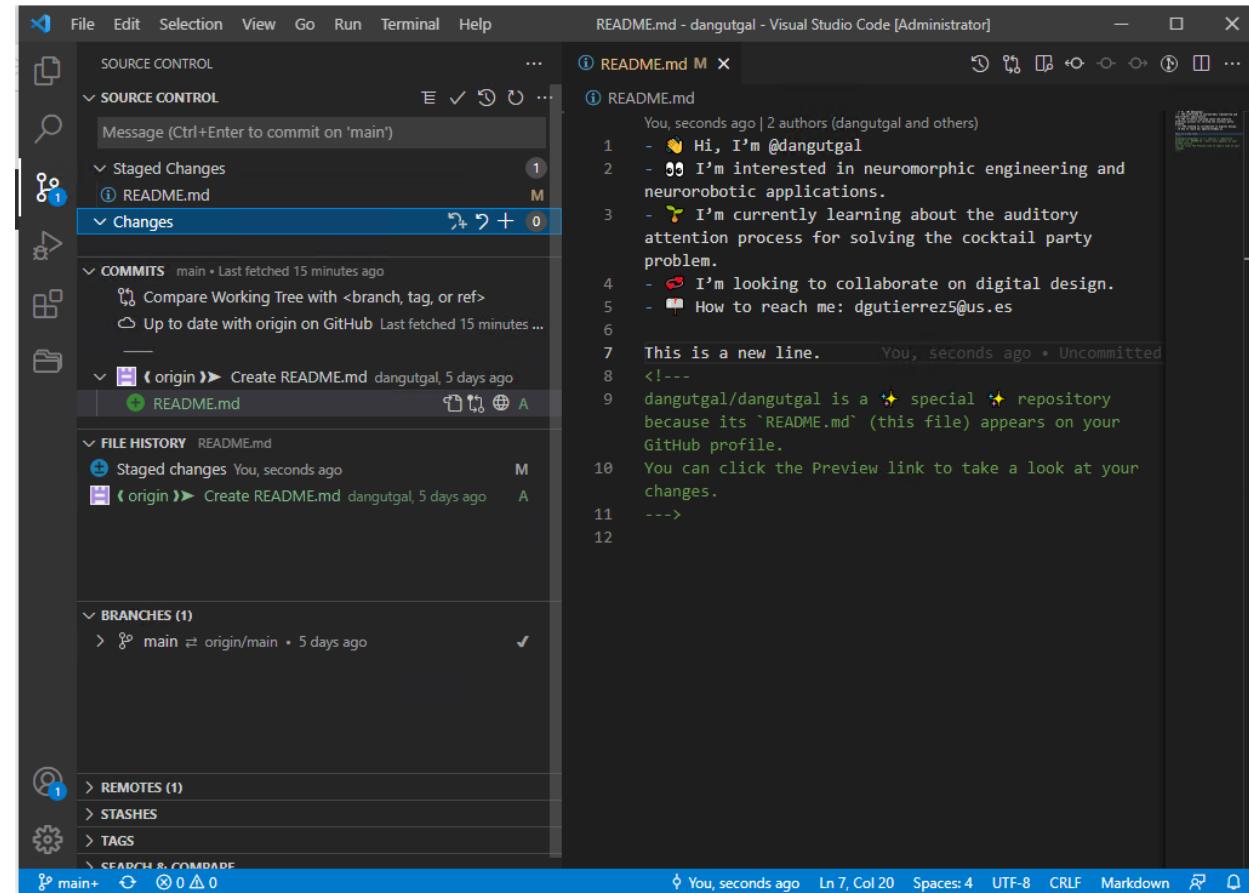
# ¿Cómo configurar mi usuario y mi contraseña?

The screenshot shows a Visual Studio Code window with the following details:

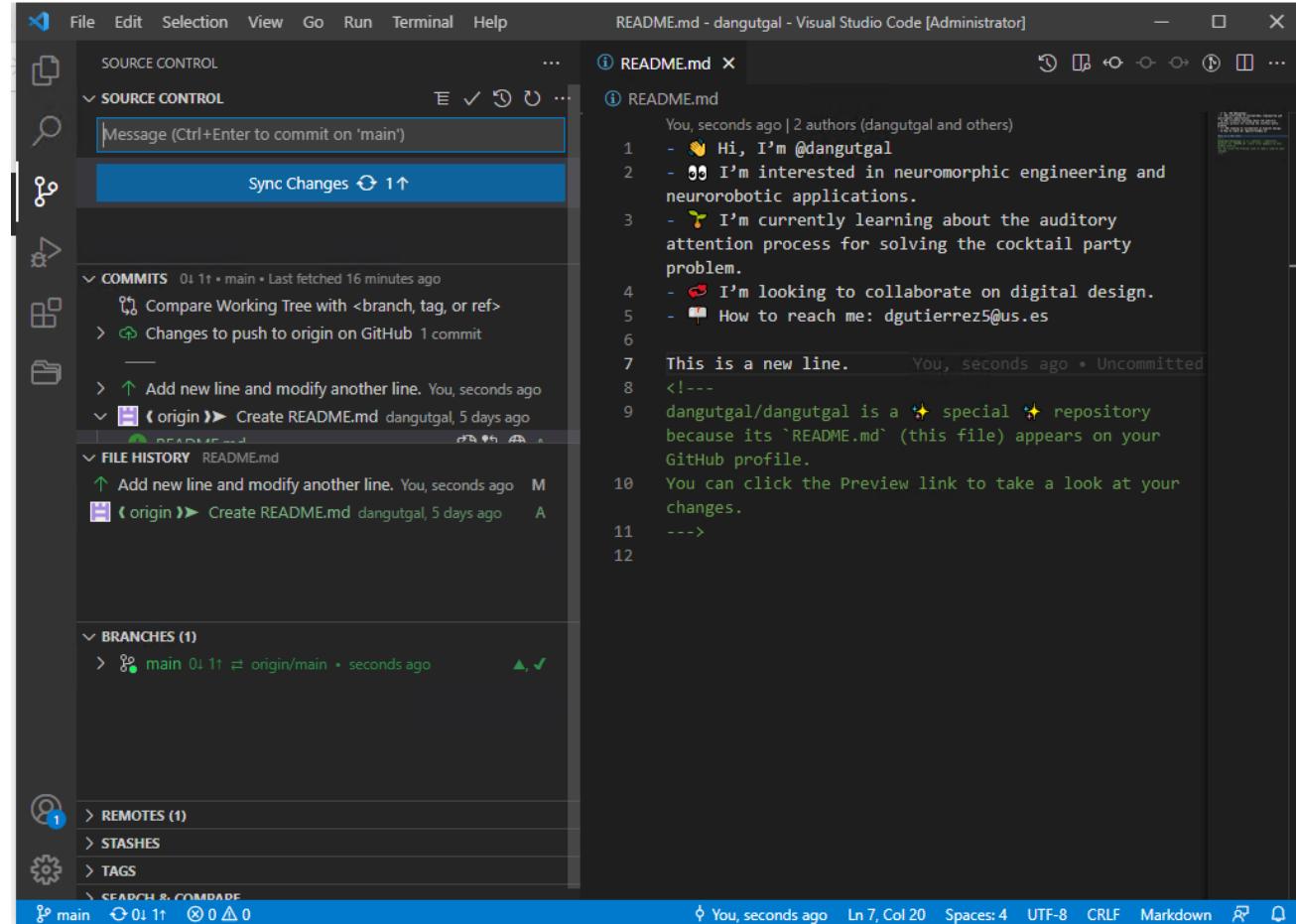
- File Explorer:** Shows a tree structure with 'SOURCE CONTROL', 'Changes' (containing 'README.md'), 'COMMITS' (main), 'FILE HISTORY' (uncommitted changes, origin), and 'BRANCHES' (main).
- Source Control View:** Shows a message 'Message (Ctrl+Enter to commit on 'main')' and a commit for 'README.md' by 'origin' (dangutgal) 5 days ago.
- Editor:** Displays the 'README.md' file with a diff view. The changes are color-coded:
  - Red highlights: 'I'm looking to', 'collaborate on digital', 'design projects.', 'How to reach me:', 'dgtierrez5@us.es'.
  - Green highlights: 'This is a new line.'
- Bottom Status Bar:** Shows 'main\*' and other status information like 'Spaces: 4', 'UTF-8', 'CRLF', 'Markdown'.



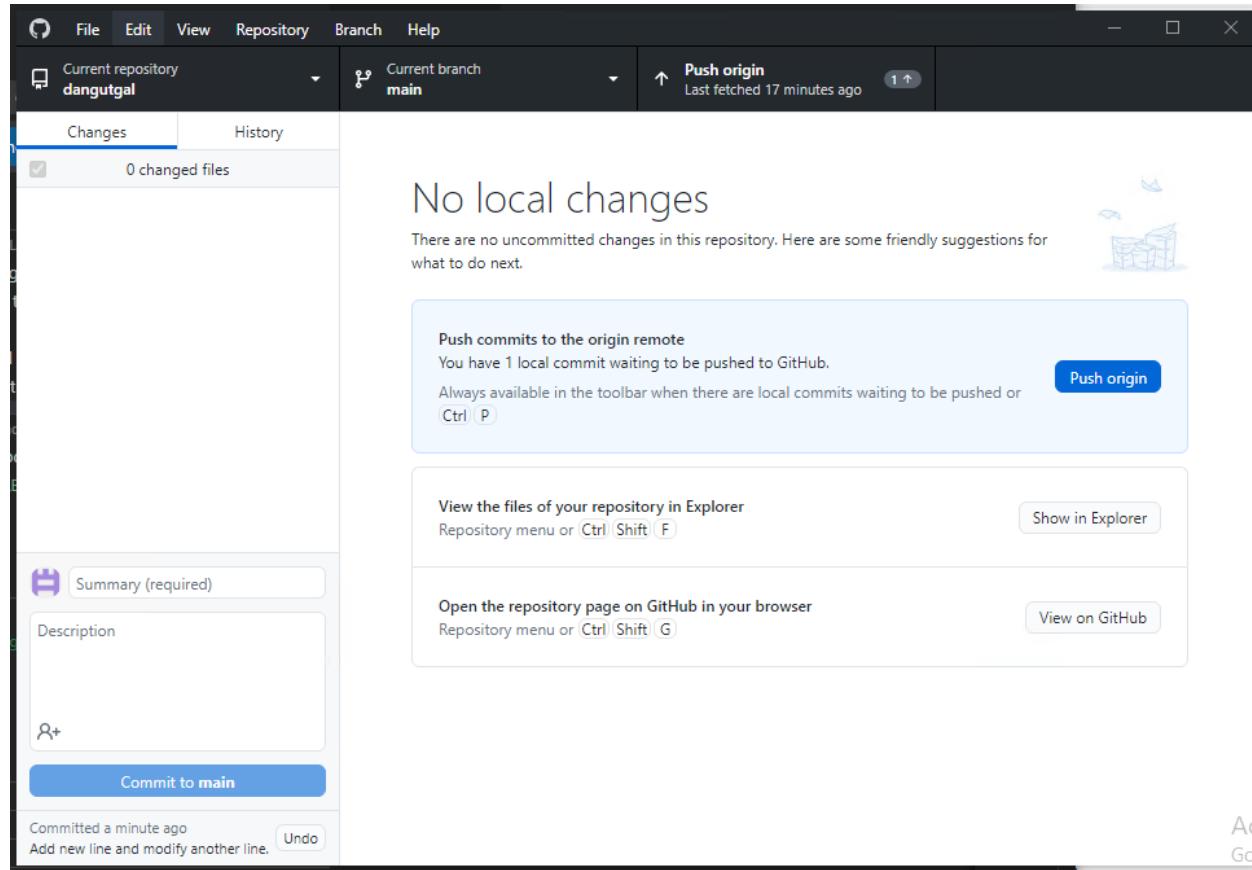
# ¿Cómo configurar mi usuario y mi contraseña?



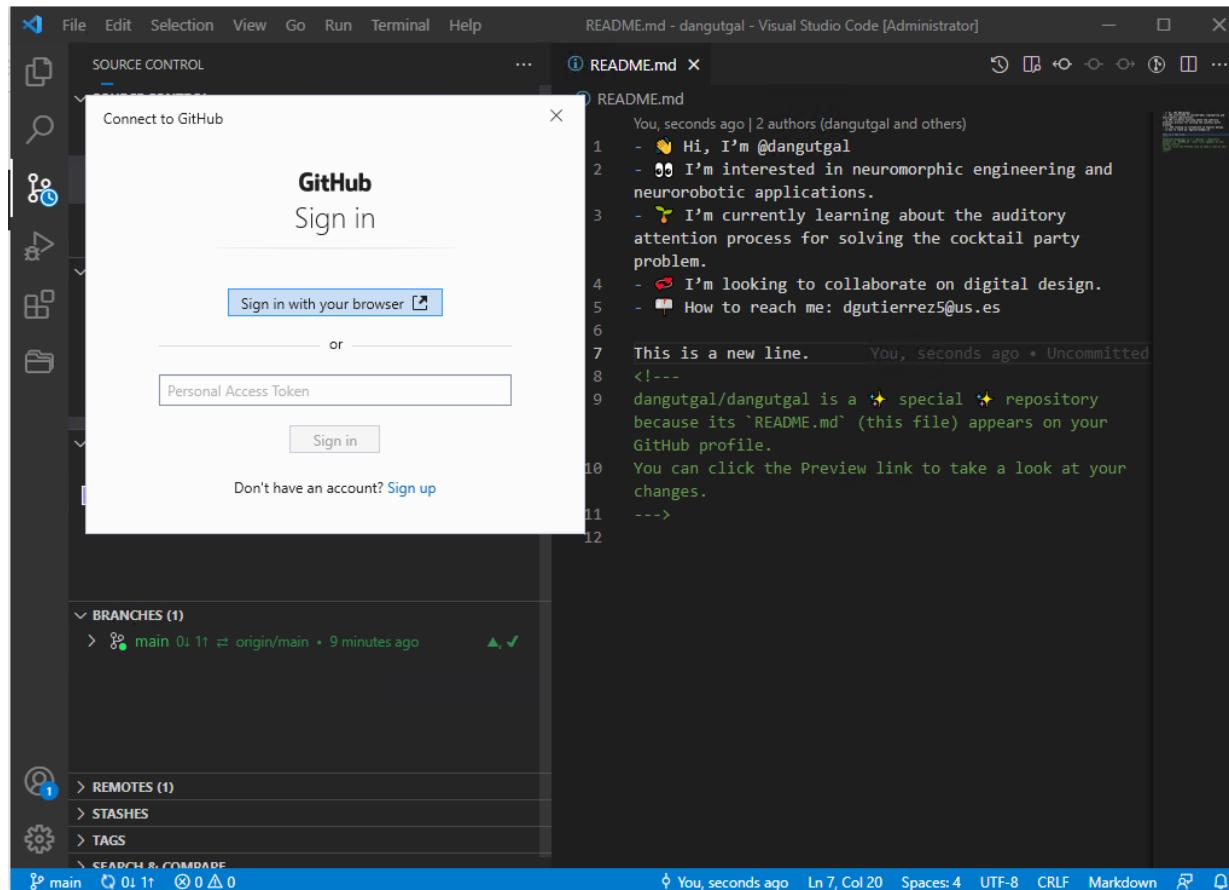
# ¿Cómo configurar mi usuario y mi contraseña?



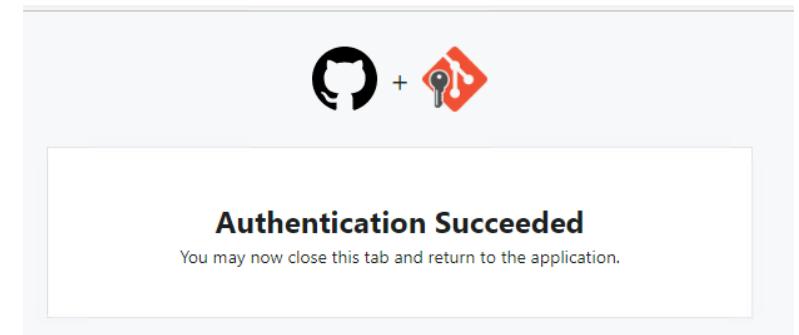
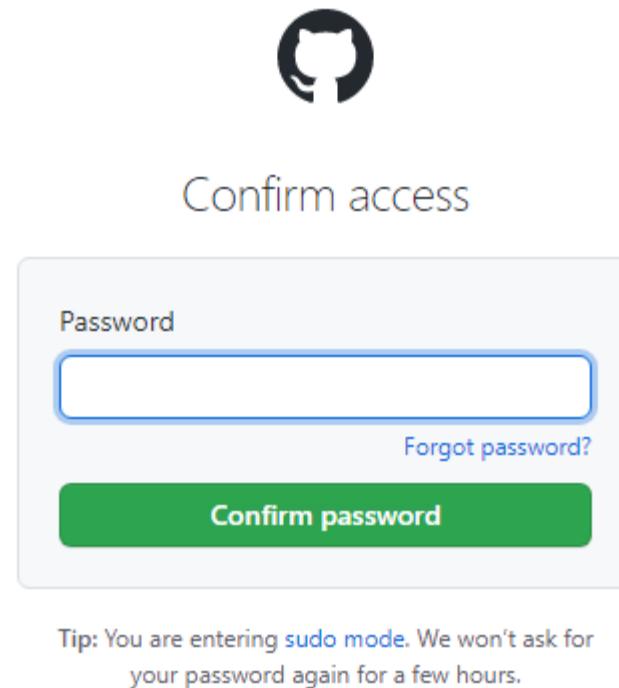
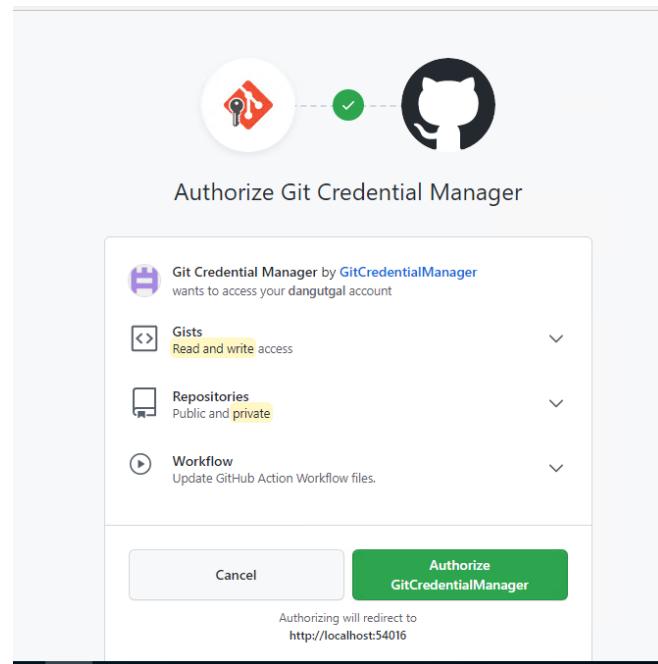
# ¿Cómo configurar mi usuario y mi contraseña?



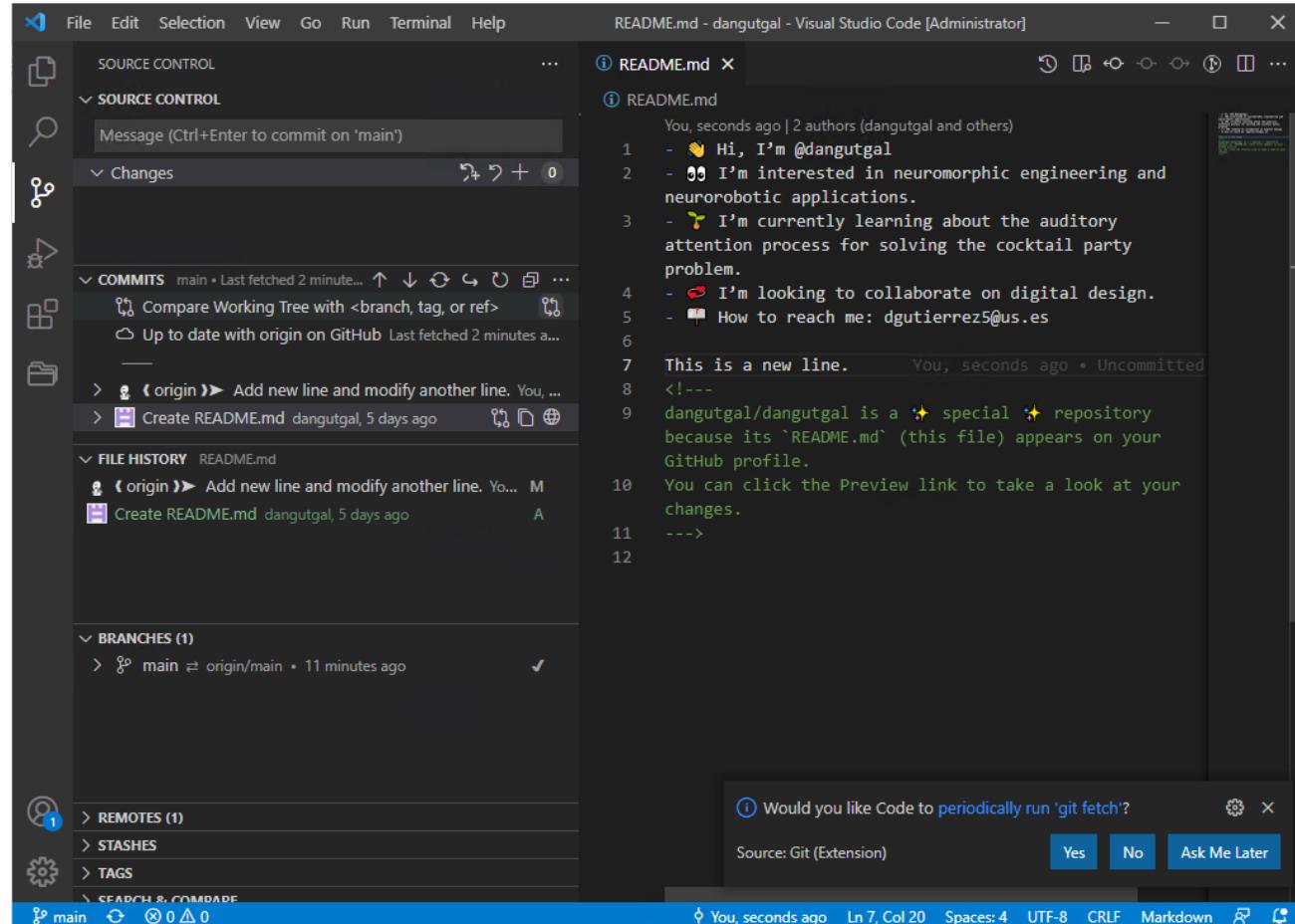
# ¿Cómo configurar mi usuario y mi contraseña?



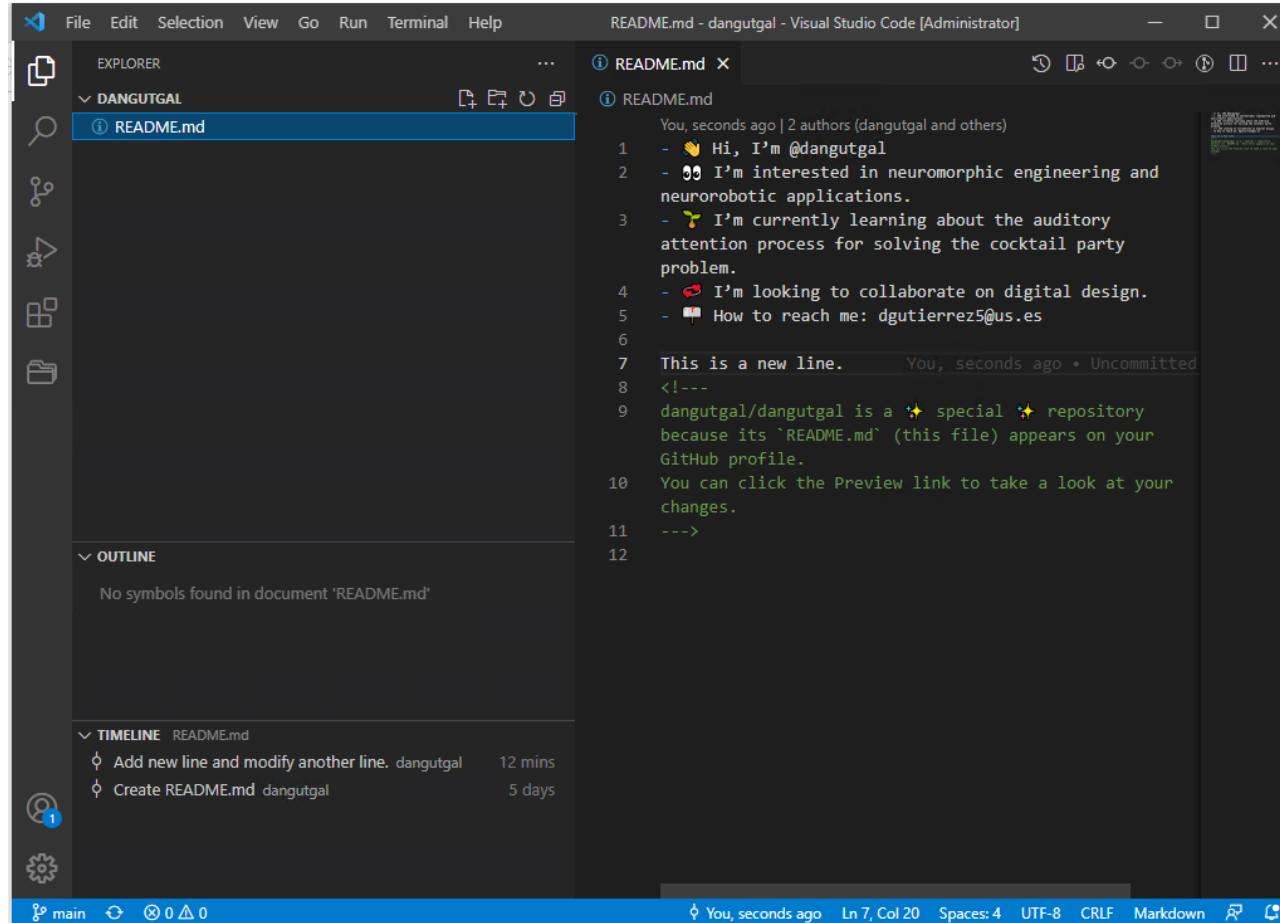
# ¿Cómo configurar mi usuario y mi contraseña?



# ¿Cómo configurar mi usuario y mi contraseña?

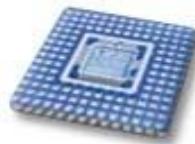


# ¿Cómo configurar mi usuario y mi contraseña?

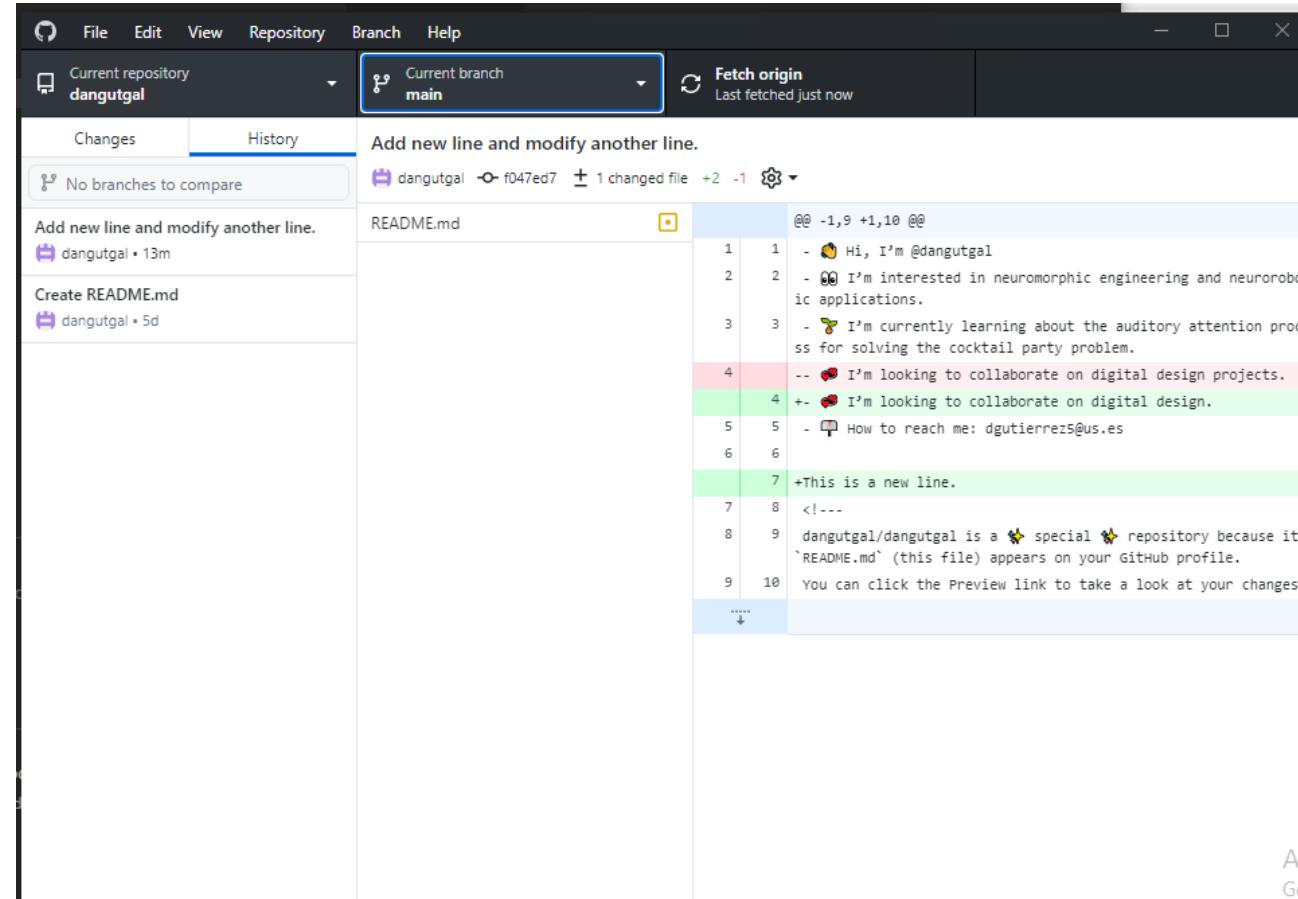


The screenshot shows the Visual Studio Code interface with a dark theme. The left sidebar has sections for EXPLORER (showing a folder named 'DANGUTGAL' containing 'README.md'), OUTLINE (empty), and TIMELINE (listing recent changes like 'Add new line and modify another line' and 'Create README.md'). The main editor area displays the 'README.md' file with the following content:

```
You, seconds ago | 2 authors (dangutgal and others)
1 - 🌟 Hi, I'm @dangutgal
2 - 💬 I'm interested in neuromorphic engineering and
neurorobotic applications.
3 - 🎨 I'm currently learning about the auditory
attention process for solving the cocktail party
problem.
4 - 🖥️ I'm looking to collaborate on digital design.
5 - 📩 How to reach me: dgutierrez5@us.es
6
7 This is a new line. You, seconds ago • Uncommitted
<!--
9 dangutgal/dangutgal is a ✨ special ✨ repository
because its `README.md` (this file) appears on your
GitHub profile.
10 You can click the Preview link to take a look at your
changes.
11 -->
12
```

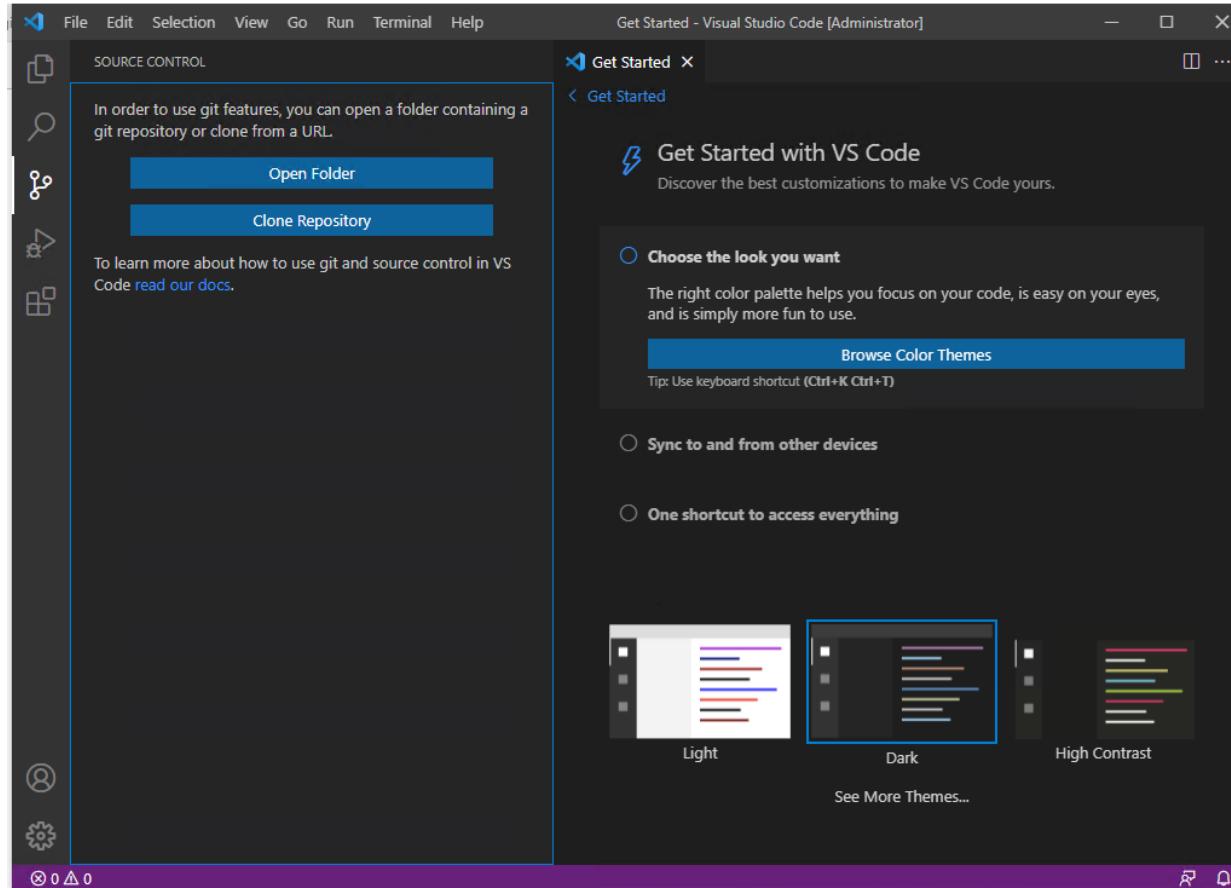


# ¿Cómo configurar mi usuario y mi contraseña?

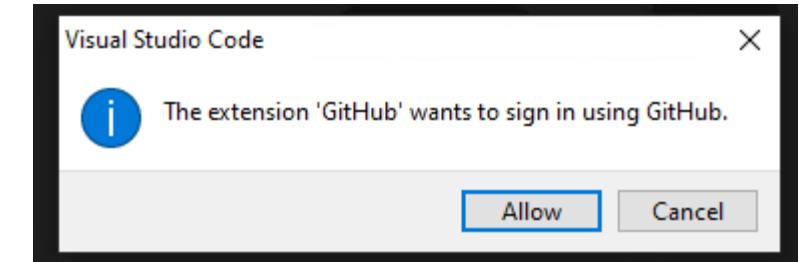
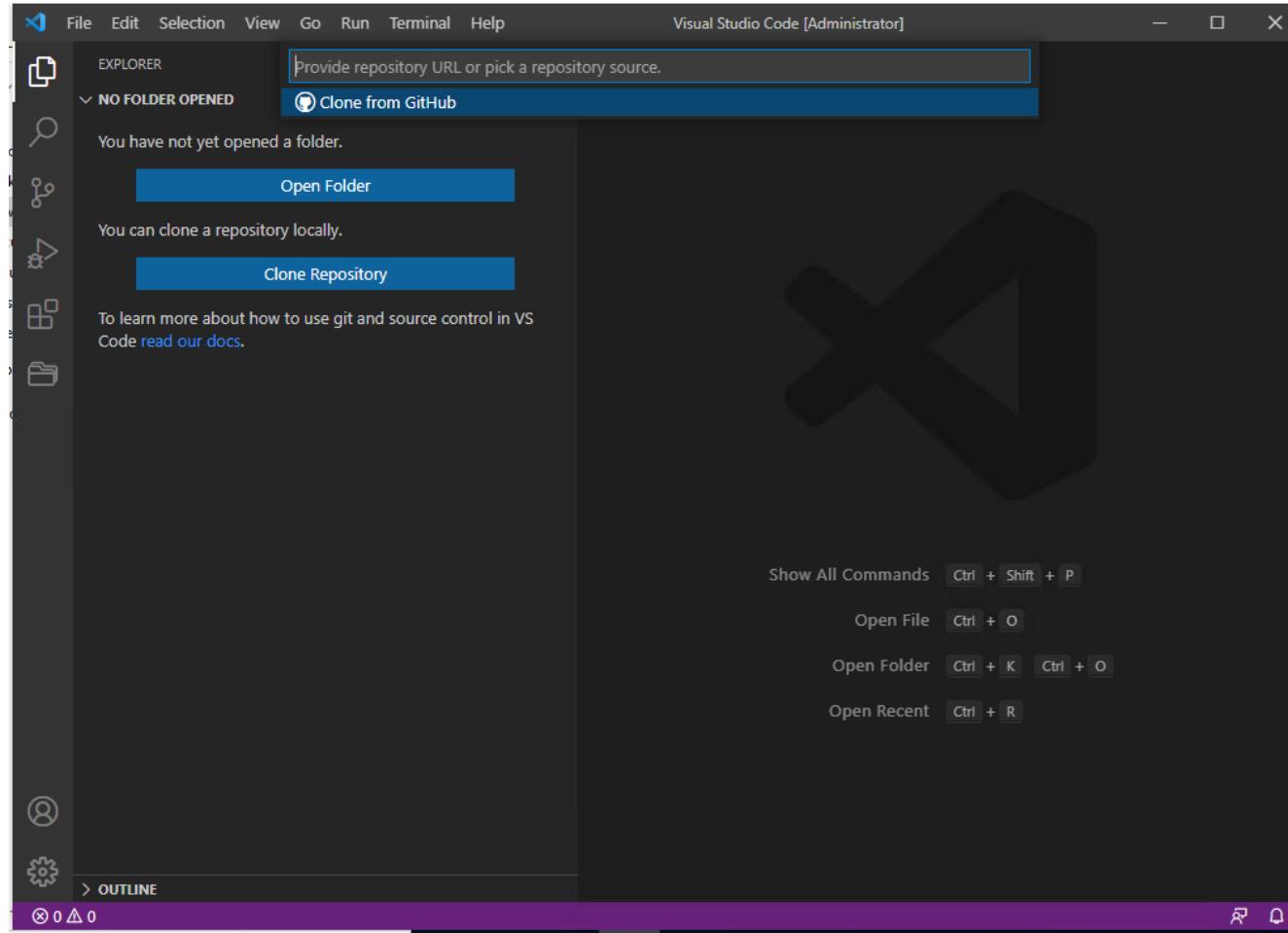


# ¿Cómo configurar mi usuario y mi contraseña?

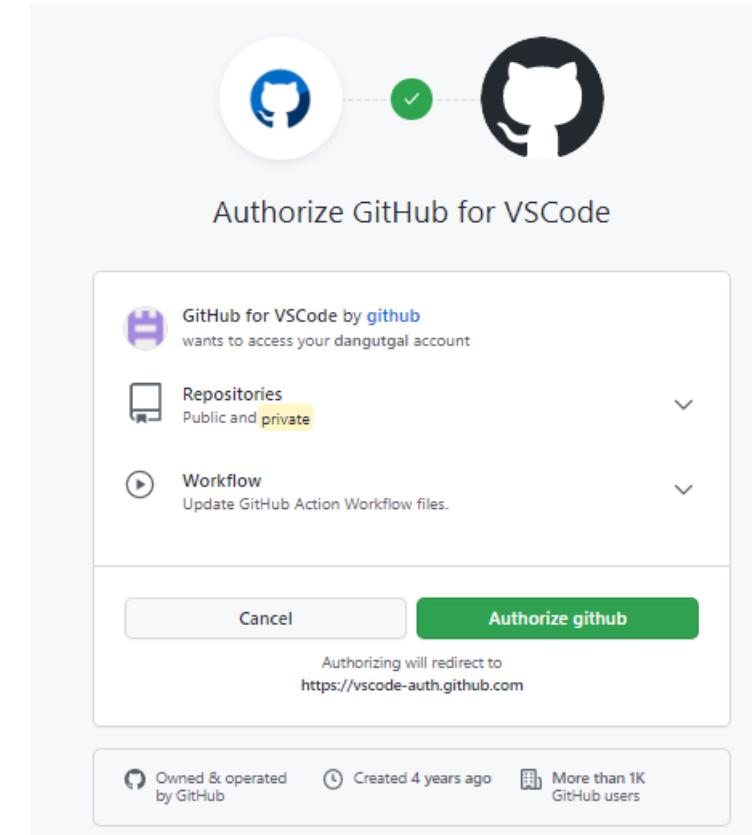
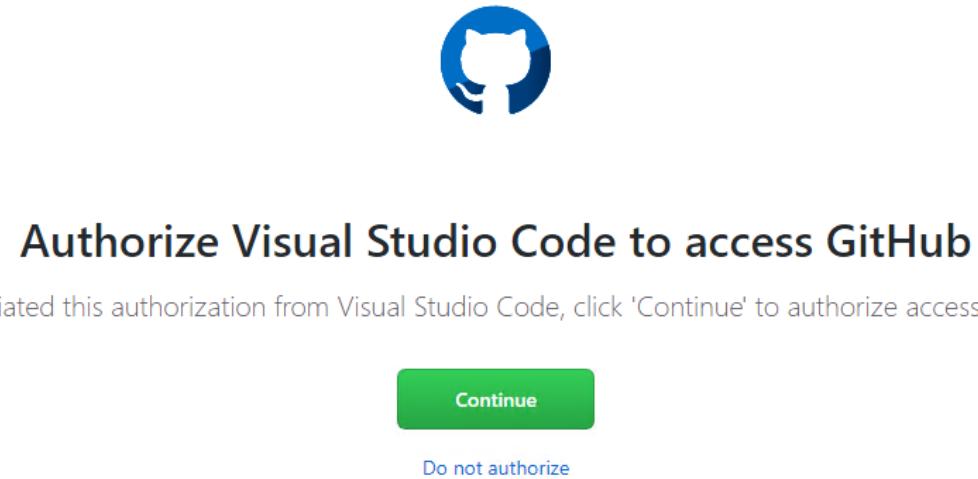
## Clonando desde Visual Studio Code



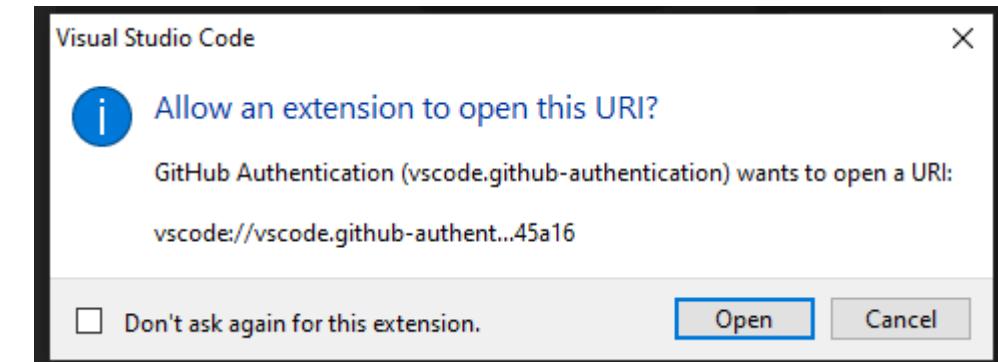
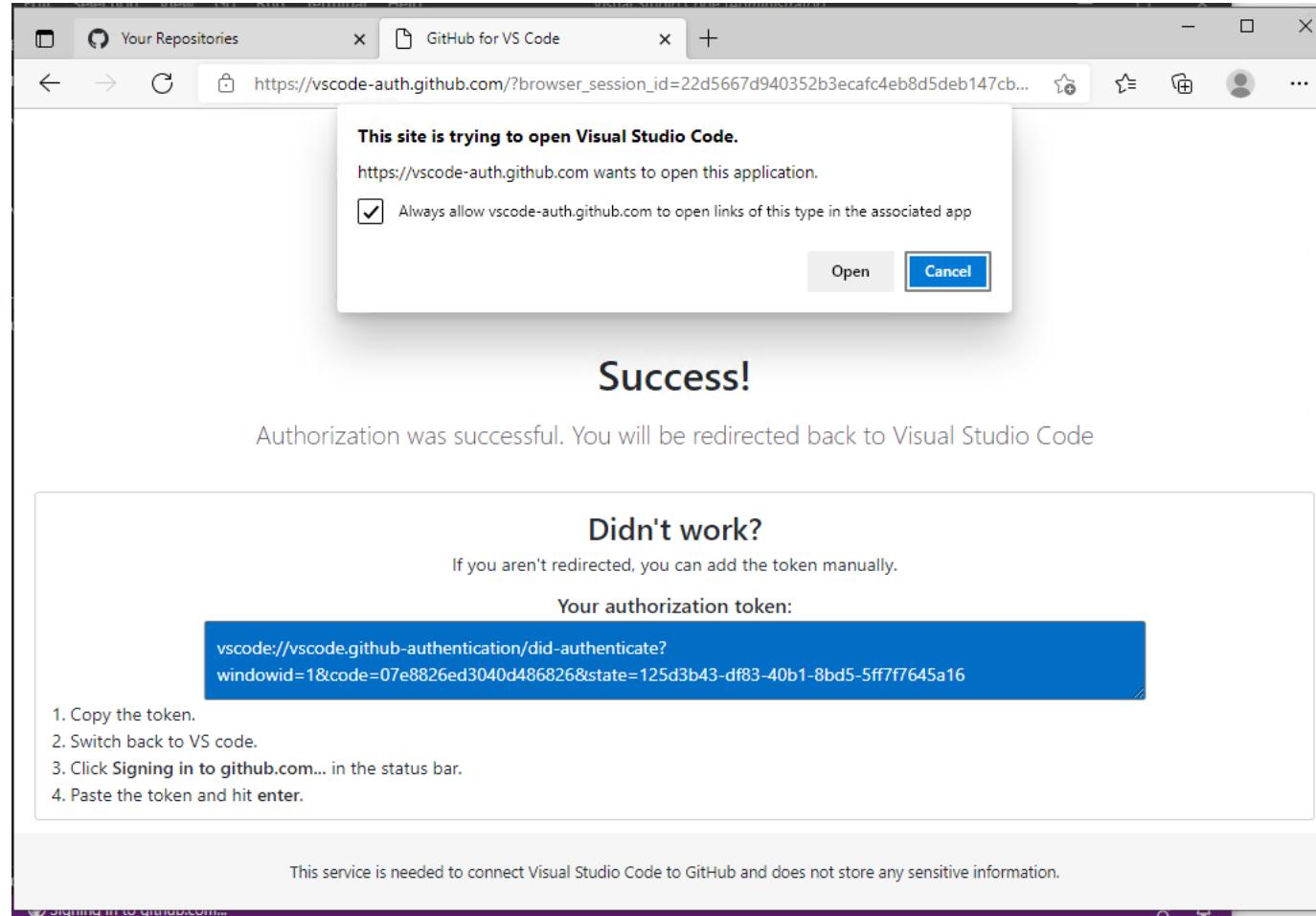
# ¿Cómo configurar mi usuario y mi contraseña?



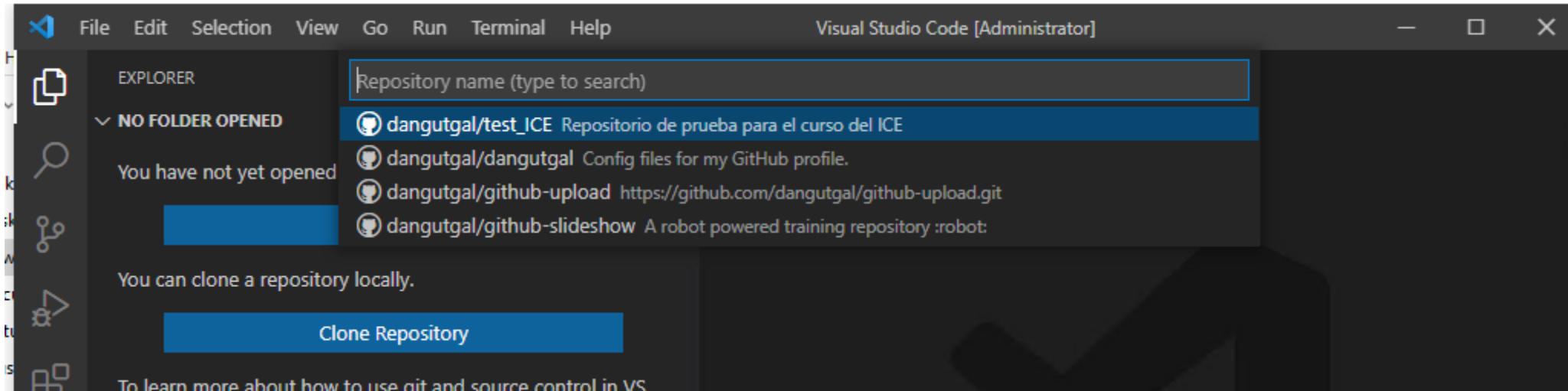
# ¿Cómo configurar mi usuario y mi contraseña?



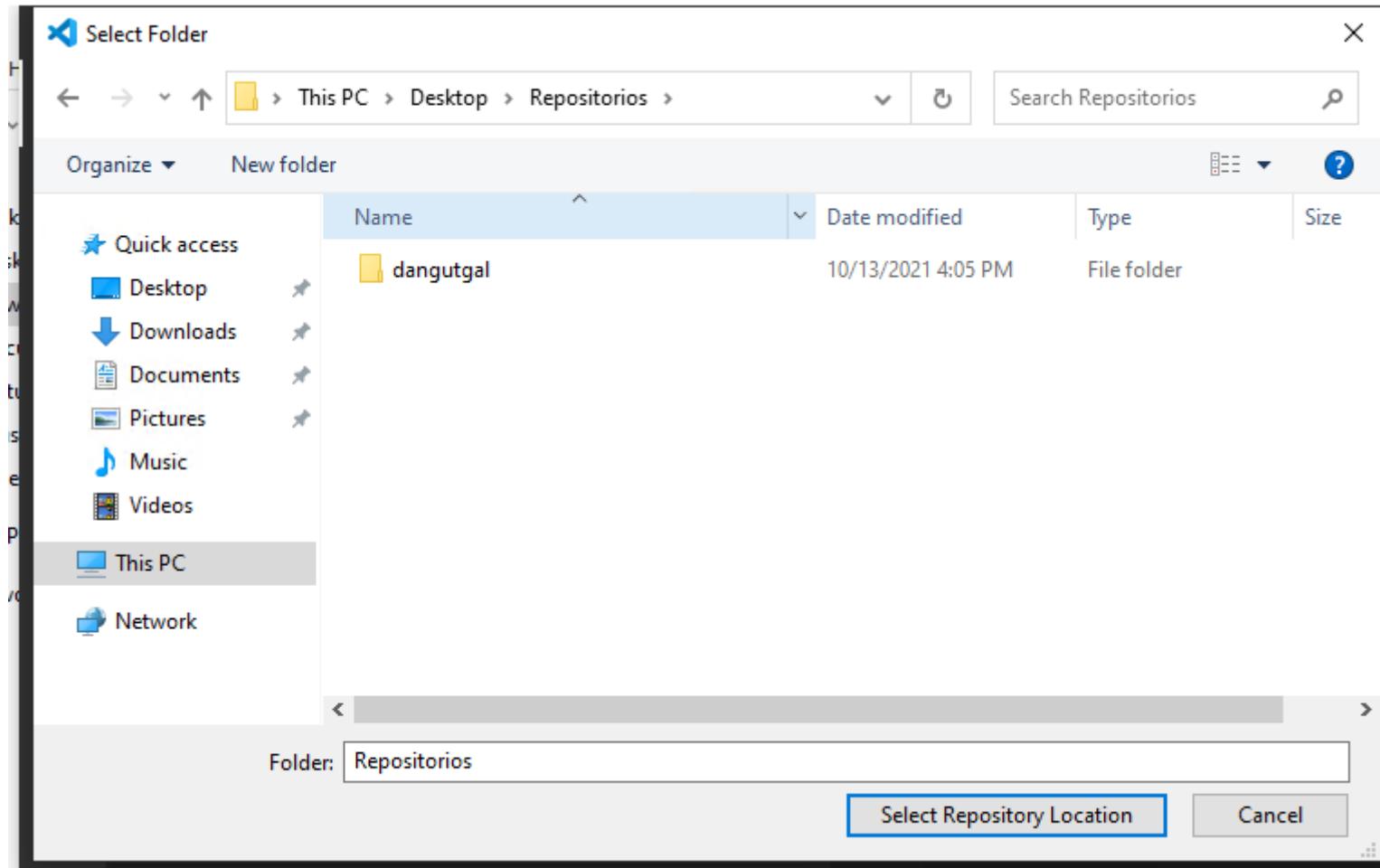
# ¿Cómo configurar mi usuario y mi contraseña?



# ¿Cómo configurar mi usuario y mi contraseña?

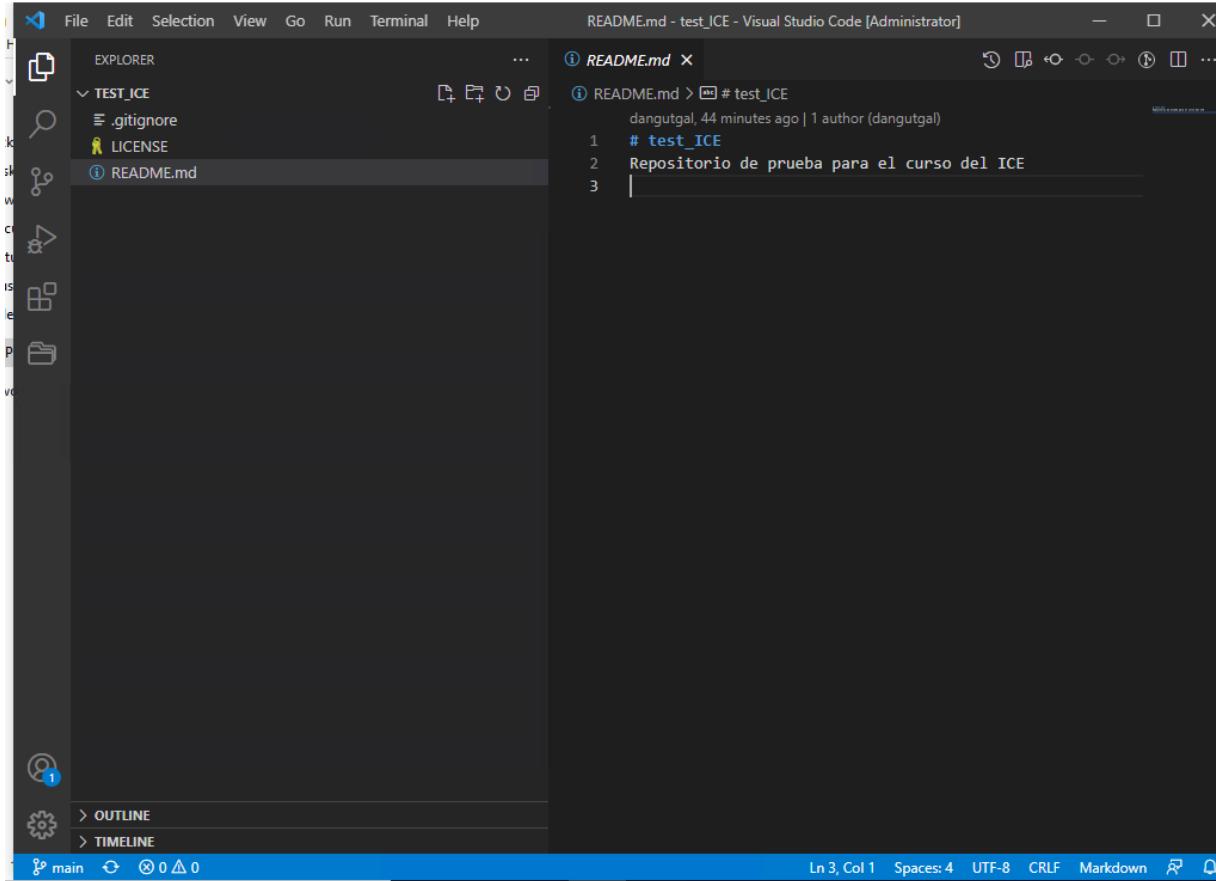


# ¿Cómo configurar mi usuario y mi contraseña?

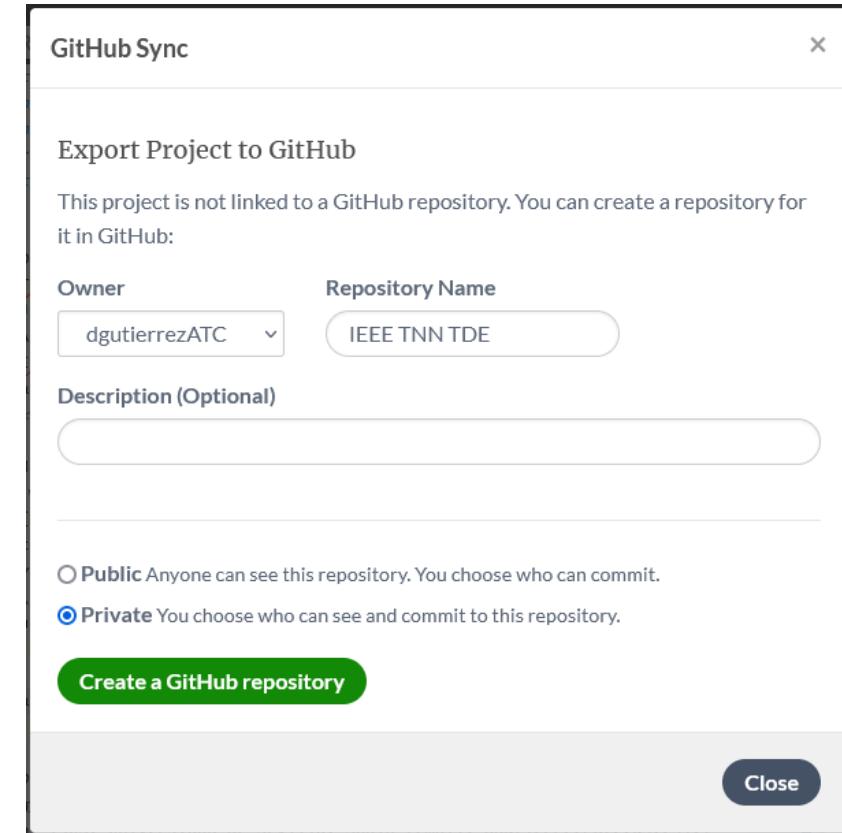
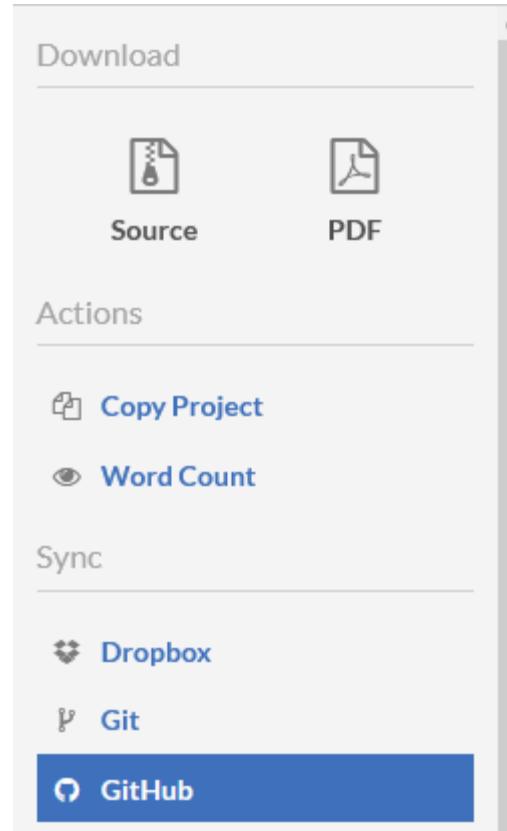
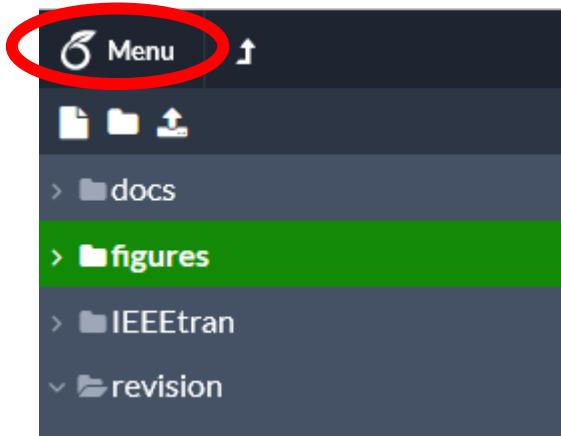


# ¿Cómo configurar mi usuario y mi contraseña?

## ¡OJO! Este repositorio no aparece en GitHub Desktop



# Ejemplos de uso en Overleaf



# Dudas, preguntas, sugerencias...

- ¡Muchas gracias por su atención y su tiempo!

- ¿Preguntas?

- Referencias:

- [1] <https://bitbucket.org/product/es/version-control-software>
- [2] <https://www.atlassian.com/es/git/tutorials/what-is-git>
- [3] <https://docs.github.com/en/get-started>
- [4] <https://www.diegocmartin.com/tutorial-git/>
- [5] <http://cessor.blogspot.com/2018/03/git-y-github-conceptos-basicos.html>
- [6] <https://www.freecodecamp.org/news/git-and-github-crash-course/>

