

Instructivo



GitHub

o  
Aptugo

# Index

<b>¡Hagan Backup!</b> .....	<b>3</b>
<b>¿Qué es Github?</b> .....	<b>3</b>
<b>Instructivo Github</b> .....	<b>4</b>
1 · Ingresar a la página de github...	4
2 · Completar datos de registro.....	5
3 · Verificación de email .....	6
4 · Pre-bienvenida .....	7
5 · Más preguntas .....	8
6 · Seleccionar plan .....	9
7 · Panel de bienvenida.....	10
8 · Instalación Github Desktop (Windows & Linux) .....	10
9. Ingreso a la cuenta y autorización .....	13
10. Configuración de Github Desktop .....	15
11. Repositorios, Commit y uso general .....	16
<b>Conclusiones:</b> .....	<b>21</b>

# ¡Hagan Backup!

A la hora de trabajar en los proyectos, es normal que cometamos errores. A veces sabremos repararlos, otras quizás el error sea tal que no nos quede más remedio que comenzar desde cero, significando una re-inversión de tiempo y concentración.

La idea de hacer una copia de seguridad o backup nos permite evitar esta situación. Ir guardando los diferentes progresos, generando puntos de restauración a los cuales volver en caso de ser necesario. Realizar una copia cada cierto tiempo es una buena práctica que nos ahorra tiempo y unos cuantos dolores de cabeza.

Las técnicas para realizar backup son varias, pero en las siguientes páginas les presentaremos una muy utilizada en el mundo IT: Github.

## ¿Qué es Github?

Github es una plataforma destinada a crear, guardar y compartir código de diferentes proyectos. En principio, es de gestión abierta, lo cual habilita a que cualquier usuario pueda subir y descargar cualquier código disponible, pero también permite la opción de generar repositorios privados. Puede ser usada desde la web, o empleando la versión de escritorio (Desktop)

La ventaja de esta herramienta es su Sistema de Control de Versiones, el cual permite ir generando, como su nombre lo indica, versiones de la aplicación vinculada. Las mismas se actualizan con los cambios, pudiendo ser de diferente naturaleza (agregar tablas, modificar datos, eliminar campos, entre otros). En caso de errores, es posible volver a una versión anterior, restaurar el trabajo y proseguir desde un punto de recuperación.

**¡A continuación, veremos cómo comenzar a usarla!**

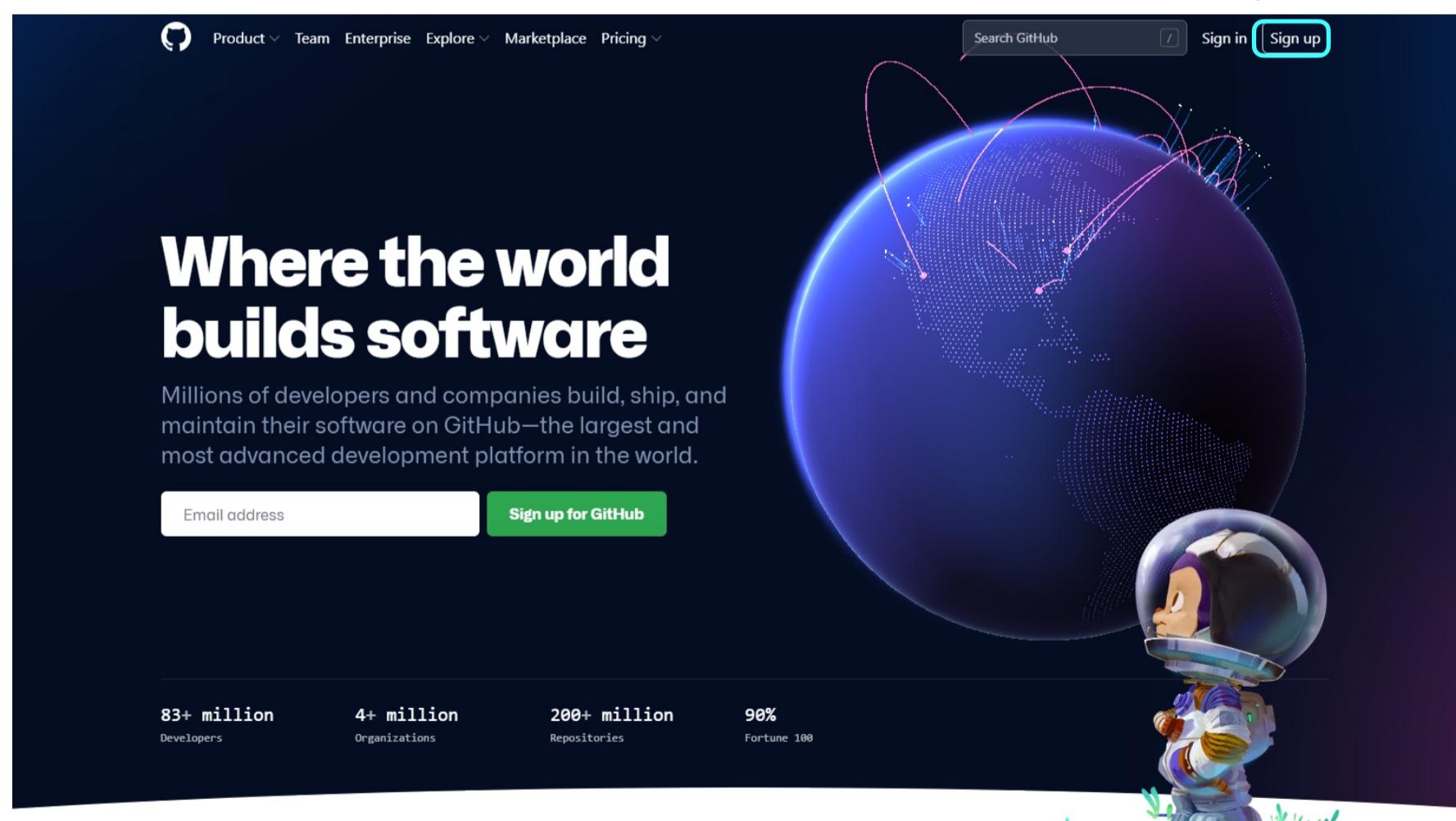
# Instructivo Github

En esta breve guía comenzaremos creando nuestra cuenta en Github. Luego de algunas configuraciones, trabajaremos con Github Desktop y aprenderemos a subir todos nuestros proyectos creados en Aptugo para siempre tener una copia de respaldo.

## 1 · Ingresar a la página de github...

<https://github.com/>

... y hacer click en **Sign Up**.



## 2 · Completar datos de registro

Completar la siguiente información:

Welcome to GitHub!  
Let's begin the adventure

Enter your email  
✓ tucorreoelectrónico@gmail.com

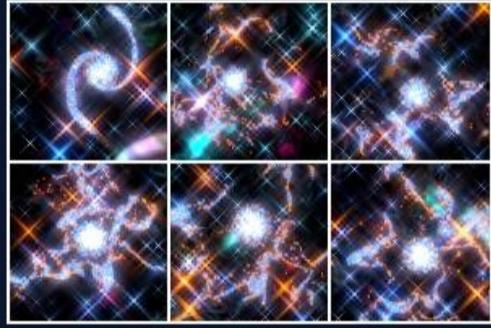
Create a password  
✓ .....\*

Enter a username  
✓ TuUsuarioPersonal

Would you like to receive product updates and announcements via email?  
Type "y" for yes or "n" for no  
✓ n

Verify your account

Seleccione la galaxia en espiral



↻ ↴

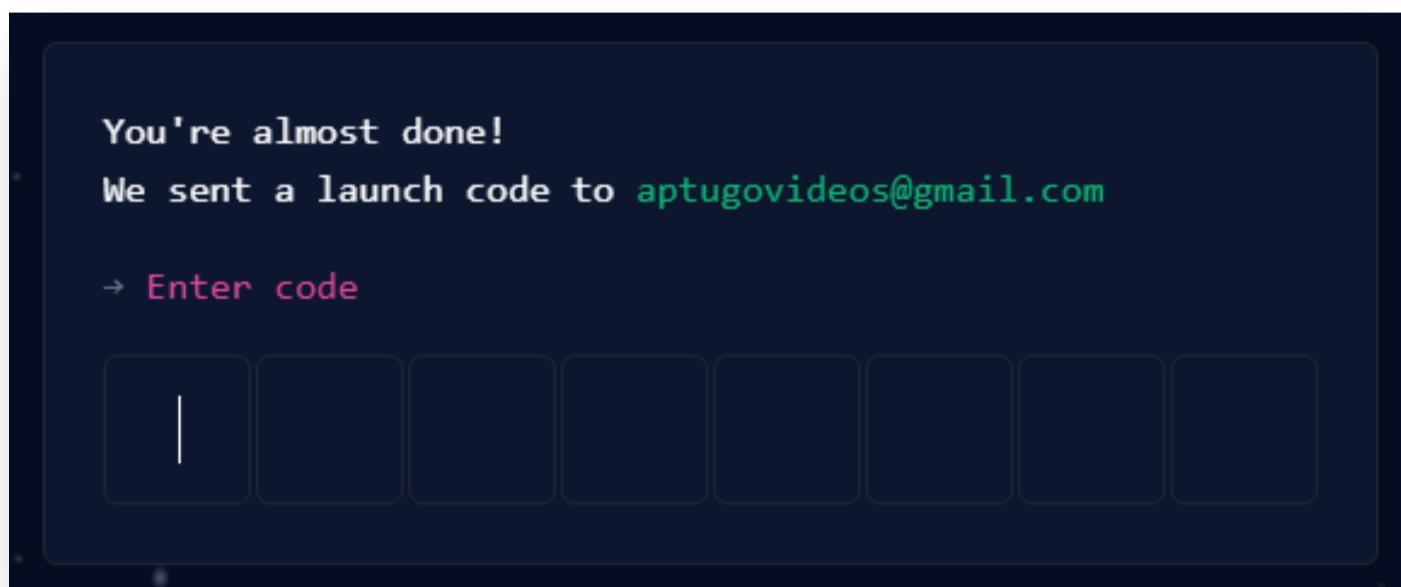
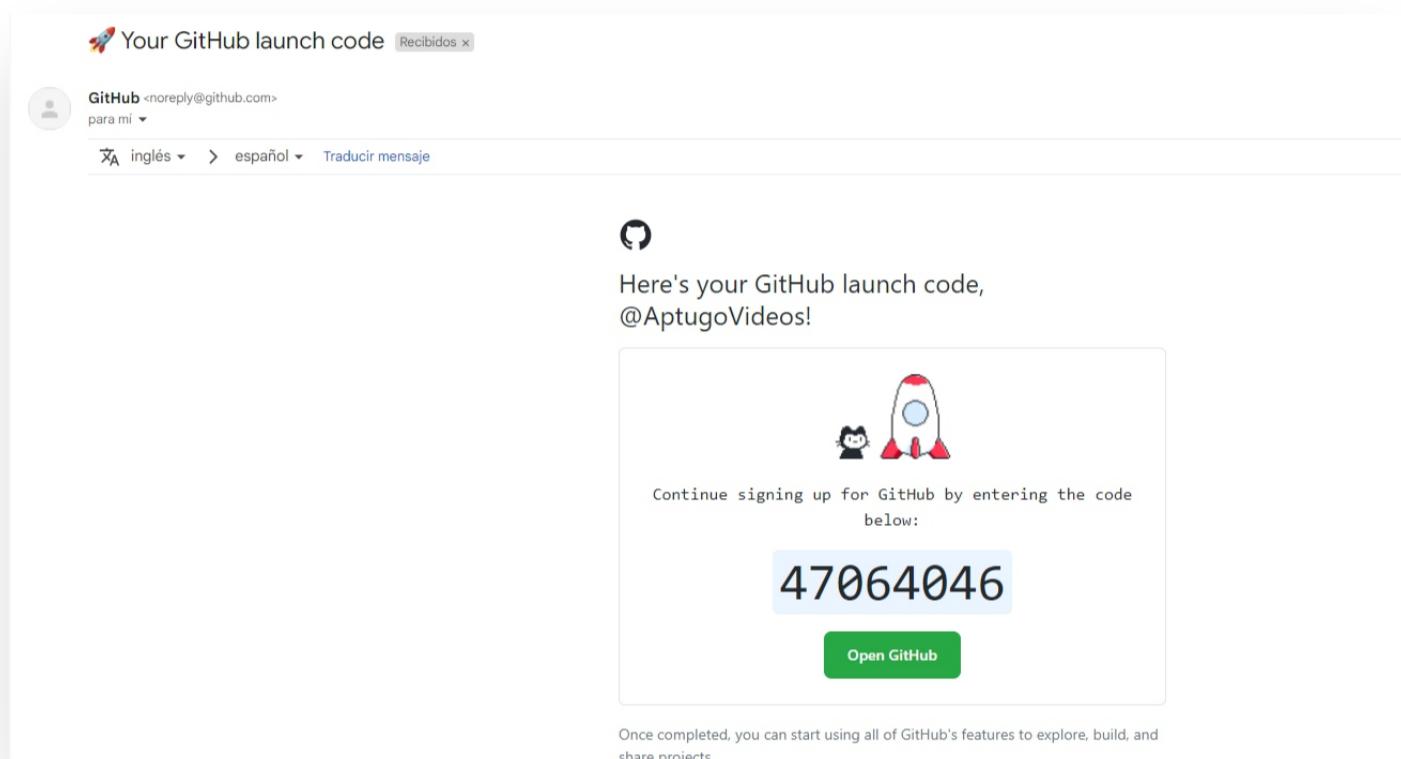
**Create account**

Resolver el captcha y hacer click en **Create account**:



### 3 · Verificación de email

Debes revisar tu bandeja de entrada, donde recibirás un e-mail con el código de 8 dígitos de verificación. Puedes copiar y pegar para asegurarte de que es el correcto.



## 4 · Pre-bienvenida

¡Ya tienes creada tu cuenta!

Ahora, puedes contestar las preguntas o hacer click en “**skip personalization**”.

En caso de querer completar la información, deberías seleccionar: **Just me** y **Student**.

The screenshot shows the fourth step of the Microsoft 365 Personal account setup wizard. The title is "How many team members will be working with you?". A note below says, "This will help us guide you to the tools that are best suited for your projects." Six options are listed in a grid: "Just me" (selected), "2 - 5", "5 - 10", "10 - 20", "20 - 50", and "50+". Below this is the question "Are you a student or teacher?", with "Student" selected. At the bottom is a blue "Continue" button with a hand cursor icon pointing to it.

How many team members will be working with you?

This will help us guide you to the tools that are best suited for your projects.

Just me

2 - 5

5 - 10

10 - 20

20 - 50

50+

Are you a student or teacher?

Student

Teacher

Continue

## 5 · Más preguntas

Te preguntarán siquieres activar alguna de las características mostradas. Lo más recomendable es ir hasta el final y clickear en **Continue**, ya que no son necesarias, pero si quieres o tienes cierto conocimiento previo, **puedes activarlas**.

### What specific features are you interested in using?

Select all that apply so we can point you to the right GitHub plan.

-  Collaborative coding  
Codespaces, Pull requests, Notifications, Code review, Code review assignments, Code owners, Draft pull requests, Protected branches, and more.
-  Automation and CI/CD  
Actions, Packages, APIs, GitHub Pages, GitHub Marketplace, Webhooks, Hosted runners, Self-hosted runners, Secrets management, and more.
-  Security  
Private repos, 2FA, Required reviews, Required status checks, Code scanning, Secret scanning, Dependency graph, Dependabot alerts, and more.
-  Client Apps  
GitHub Mobile, GitHub CLI, and GitHub Desktop.
-  Project Management  
Projects, Labels, Milestones, Issues, Unified Contribution Graph, Org activity graph, Org dependency insights, Repo insights, Wikis, and GitHub Insights.
-  Team Administration  
Organizations, Invitations, Team sync, Custom roles, Domain verification, Audit Log API, Repo creation restriction, and Notification restriction.
-  Community  
GitHub Marketplace, GitHub Sponsors, GitHub Skills, Electron, and Atom.

**Continue**

## 6 · Seleccionar plan

Debes escoger el plan gratuito: "FREE"

**Learn to ship software like a pro.**

GitHub gives students free access to the best developer tools so they can learn by doing.

**Free**

- Unlimited public/private repositories
- 2,000 CI/CD minutes/month  
Free for public repositories
- 500MB of Packages storage  
Free for public repositories
- Community support

**Get additional student benefits**

**GitHub Pro**

- Protect your branches  
Ensure that collaborators on your repository cannot make irrevocable changes to branches.
- Draft pull requests
- Pages and Wikis
- 3,000 CI/CD minutes/month  
Free for public repositories
- 2GB of Packages storage  
Free for public repositories
- Web-based support

**GitHub Student Developer Pack**

- Free access to the industry's best developer tools  
Hundreds of offers, including Digital Ocean, Microsoft Azure, Heroku, MongoDB, DataDog, Twilio, and Stripe.

**GitHub Campus Expert training**

- Enrich your college technical community  
Learn the skills to build diverse tech communities on campus with training, mentorship, and support from GitHub.

[Continue for free](#)

[Apply for your GitHub student benefits](#)

## 7 · Panel de bienvenida

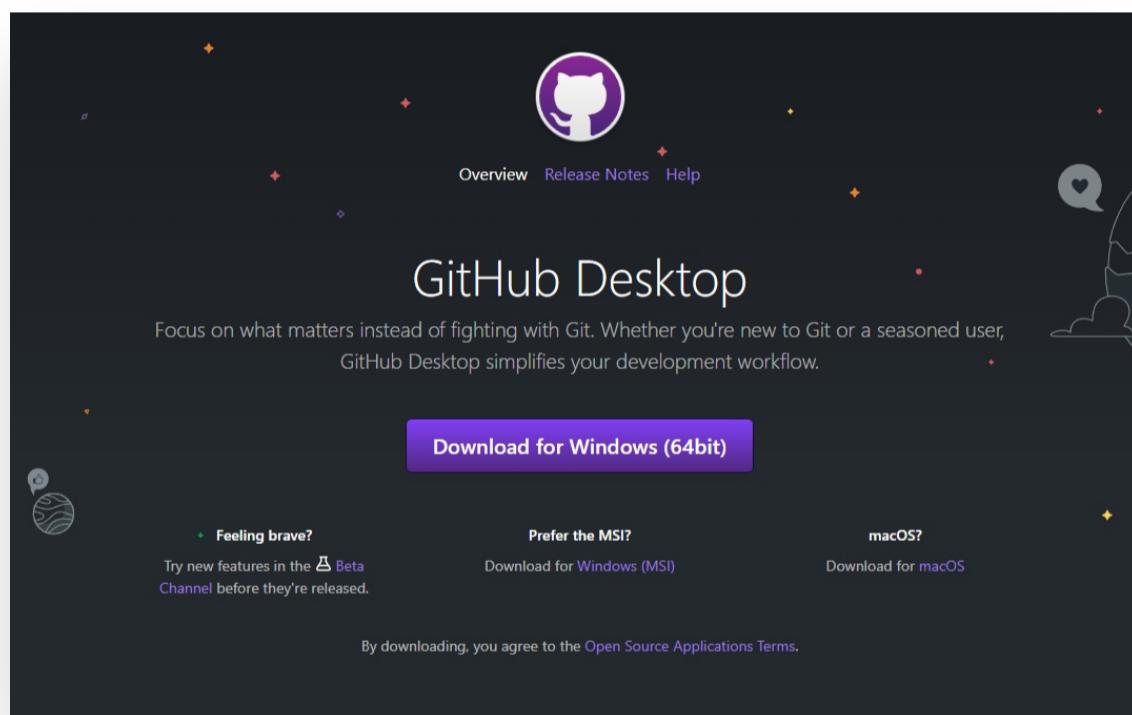
En este paso, ya deberías estar en el panel de bienvenida, lo cual nos permite comenzar a crear repositorios... **Pero antes, descarguemos e instalemos Github Desktop.**

## 8 · Instalación Github Desktop (Windows & Linux)

### 8.1 · Instalación en Windows

Ingresamos a la página:

<https://desktop.github.com/>



Hacemos click en Download for Windows (64bit)

Una vez descargado, hacemos doble click en el instalador para ejecutarlo.

**Cuando la instalación se complete, hacemos click en la opción Sign in to GitHub.com.** Esto nos llevará al navegador, en el cual deberemos ingresar nuestro usuario y contraseña.

## 8.2 · Instalación en Linux



### Advertencia preliminar:

**La siguiente versión de Github Desktop no es oficial.** Al día de la elaboración de la presente guía, no se ha encontrado soporte oficial para el sistema operativo Linux. Sin embargo, existe una versión no oficial, adaptada y funcional, cuya instalación se describirá a continuación. Se debe instalar bajo propia responsabilidad.

**Primero, en una terminal insertamos el siguiente comando, y añadiremos la contraseña solicitada:**

```
usuario@user-ubuntu:~$ sudo wget https://github.com/shiftkey/desktop/releases/download/release-2.9.3-linux3/GitHubDesktop-linux-2.9.3-linux3.deb
```

The screenshot shows a terminal window with a dark background. The title bar says "anee@anee-ubuntu: ~". The command "sudo wget https://github.com/shiftkey/desktop/releases/download/release-2.9.3-linux3/GitHubDesktop-linux-2.9.3-linux3.deb" is typed in, followed by "[sudo] contraseña para anee: [redacted]" where the password has been redacted.

**Luego, si no tenemos instalado gdebi-core, debemos hacerlo del siguiente modo (de pedir contraseña, por favor ingresarla):**

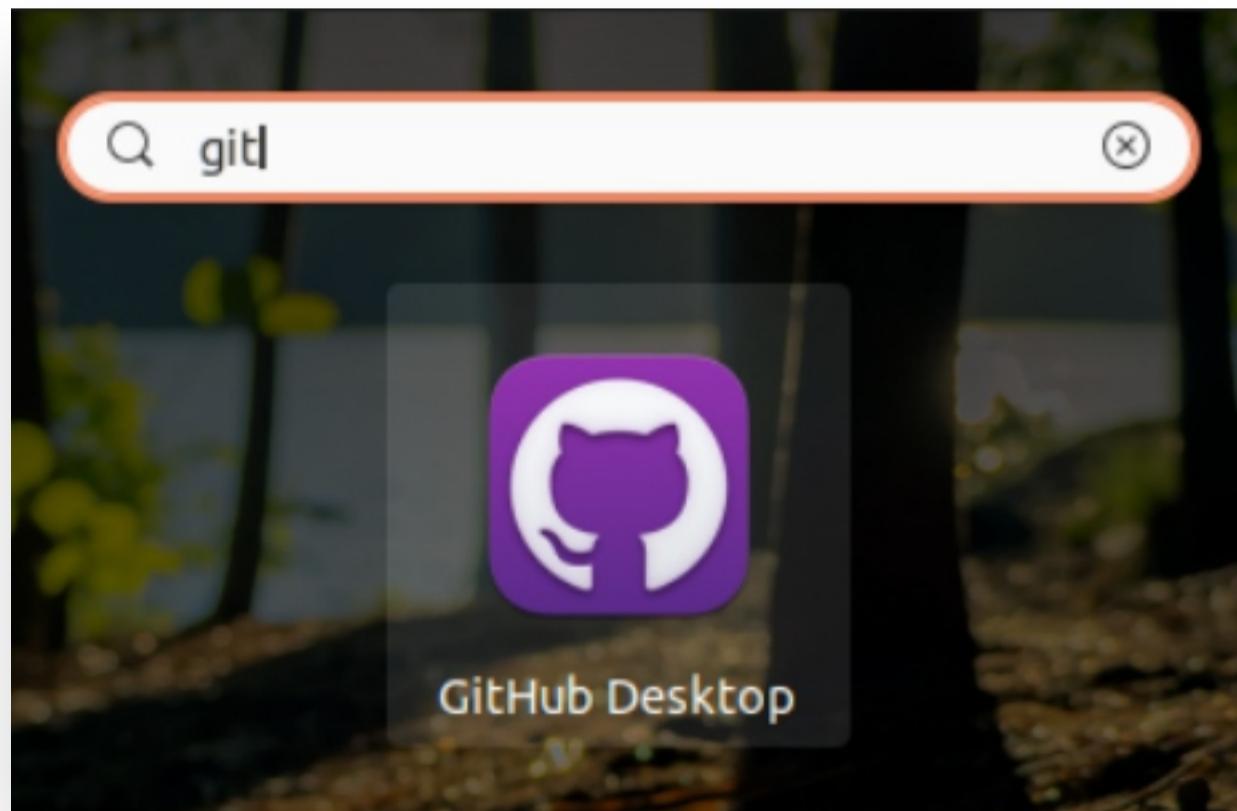
```
usuario@user-ubuntu:~$ sudo apt-get install gdebi-core
```

Nos preguntará si queremos instalarlo, **escribimos “s” (sin comillas) en la terminal y apretamos Enter.**

```
anee@anee-ubuntu:~$ sudo gdebi GitHubDesktop-linux-2.9.3-linux3.deb
Reading package lists... Done
Building dependency tree
Reading state information... Done
Reading state information... Done

Simple collaboration from your desktop
This is the unofficial port of GitHub Desktop for Linux distributions
¿Quiere instalar el paquete de software? [s/N]:s
```

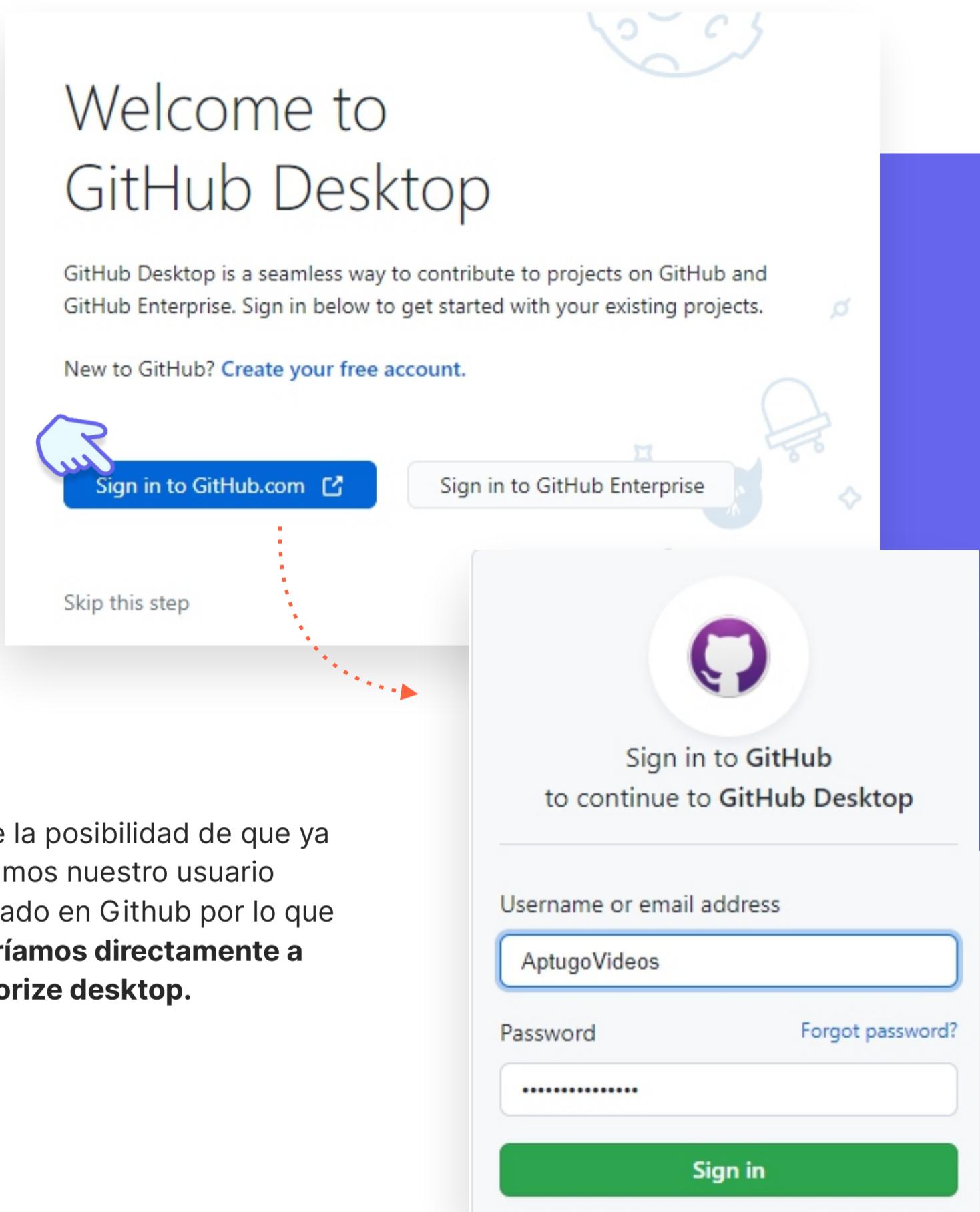
**Una vez instalado, encontraremos el icono de GitHub Desktop en nuestro cajón de aplicaciones:**

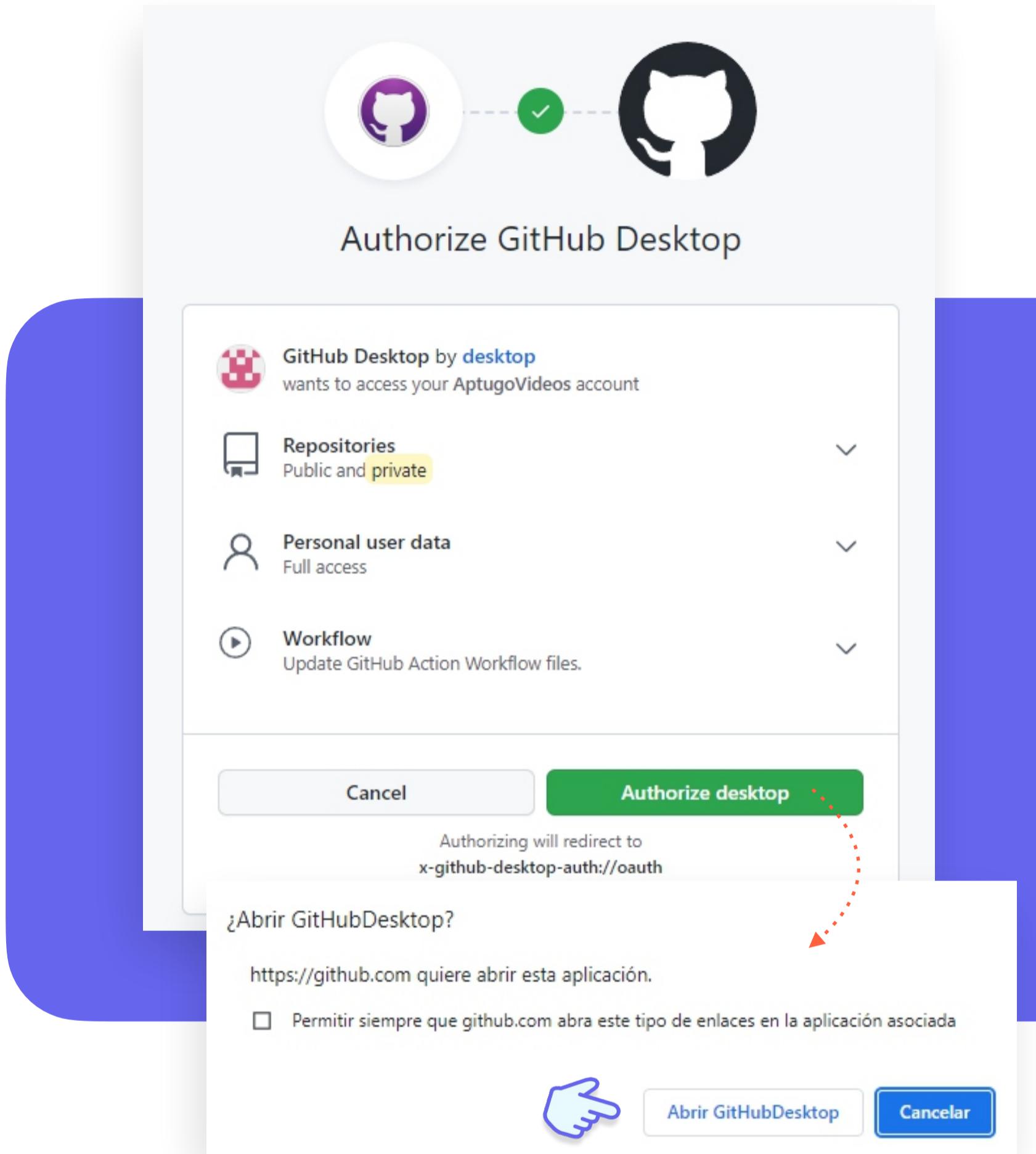


**Con ésto, hemos finalizado de instalar GitHub en Linux. 👍**

Los siguientes pasos son comunes a todos los sistemas operativos abordados en la guía.

## 9. Ingreso a la cuenta y autorización





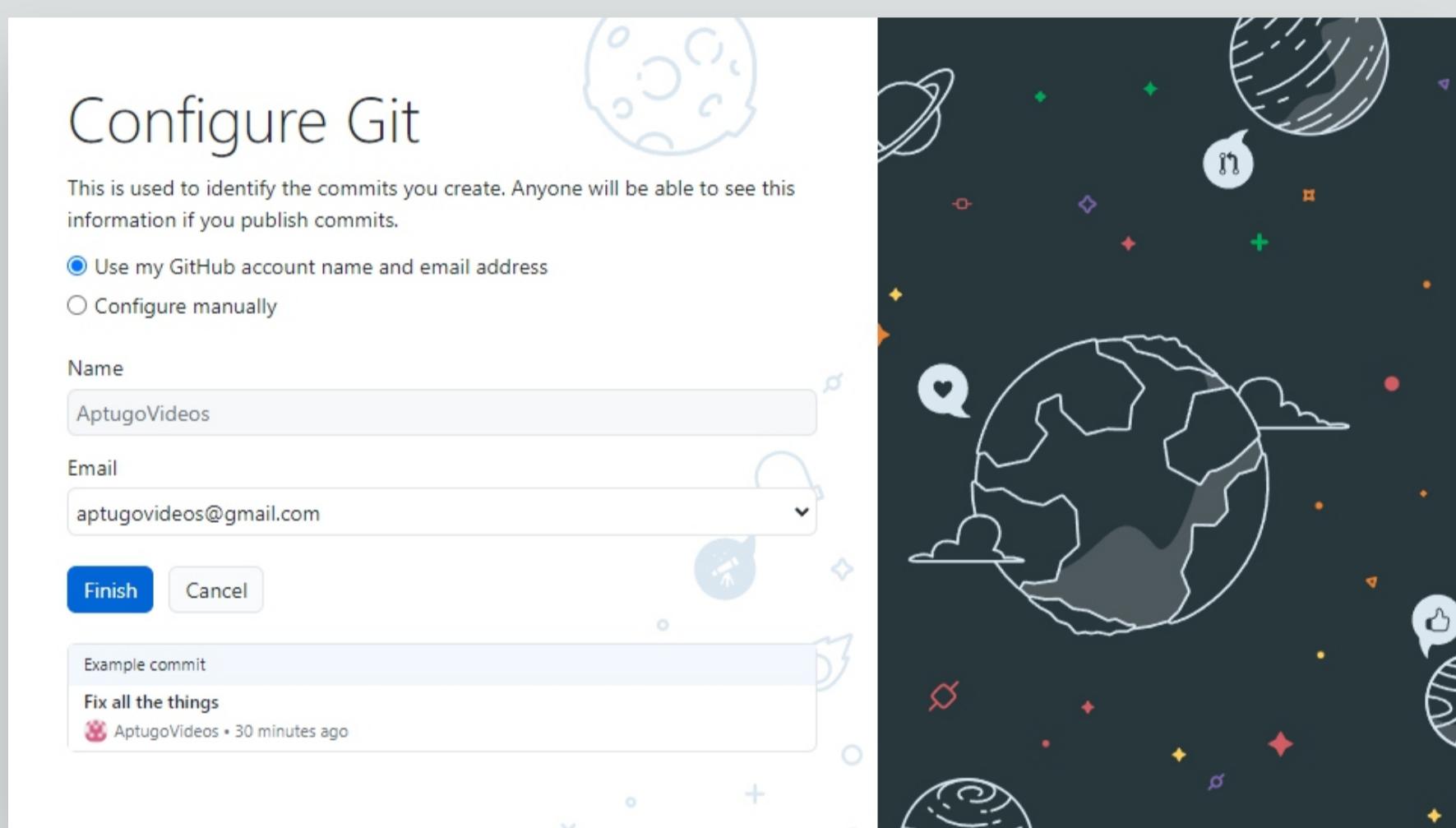
Para confirmar este paso **se nos abrirá un PopUp** al que debemos **hacer click en Abrir GitHub Desktop**.

## 10. Configuración de Github Desktop

Una vez que estamos dentro de Github Desktop, debemos configurar algunas características.

**Primero seleccionamos la opción Use My GitHub account name and email address.**

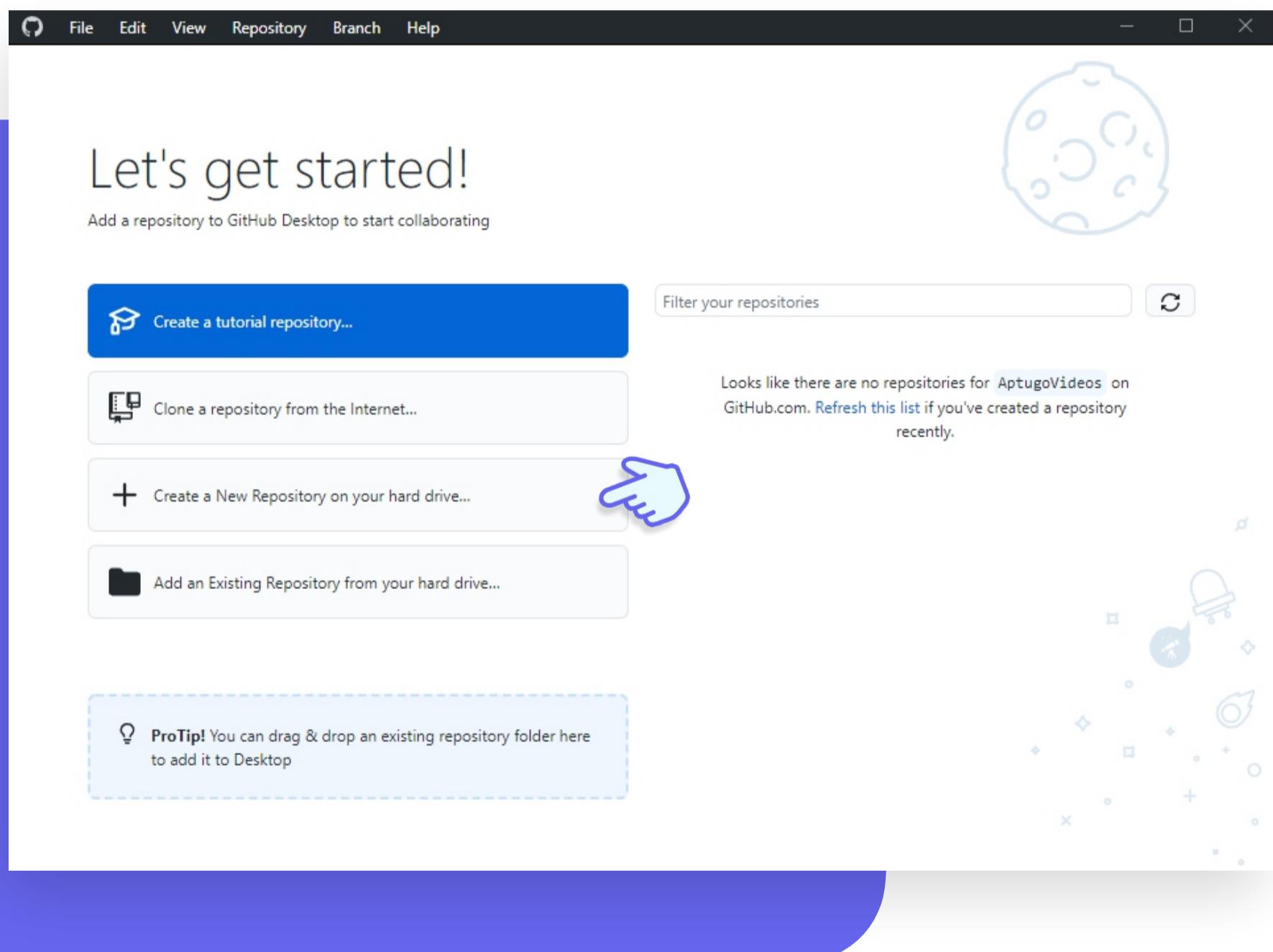
El nombre no admite modificaciones y debemos verificar que el e-mail asociado sea el correcto. **Una vez listo, continuamos haciendo click en Finish.**



## 11. Repositorios, Commit y uso general

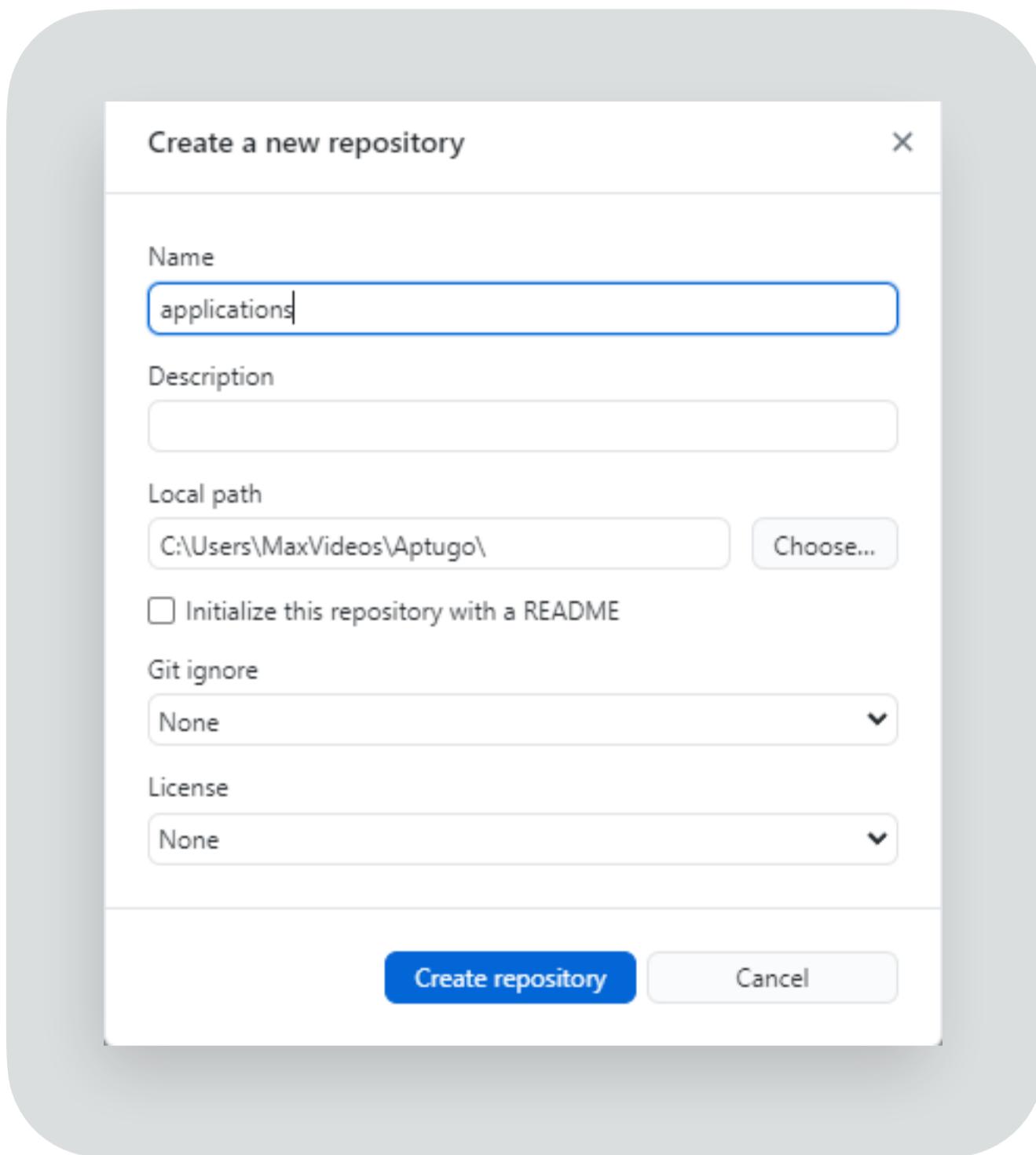
A continuación, veremos la **interfaz para crear nuestro primer repositorio dentro de Github Desktop**.

Hacemos click en la 3era opción  
**“Create a New Repository on your hard drive..”**



Colocamos como nombre el de nuestra carpeta de applications (o donde tengamos guardados nuestros proyectos de Aptugo, no la carpeta builds).

Una vez hecho eso hacemos click en el botón “**Create repository**”



## Crear un Commit:

Empecemos por lo básico, **¿que es un commit?**



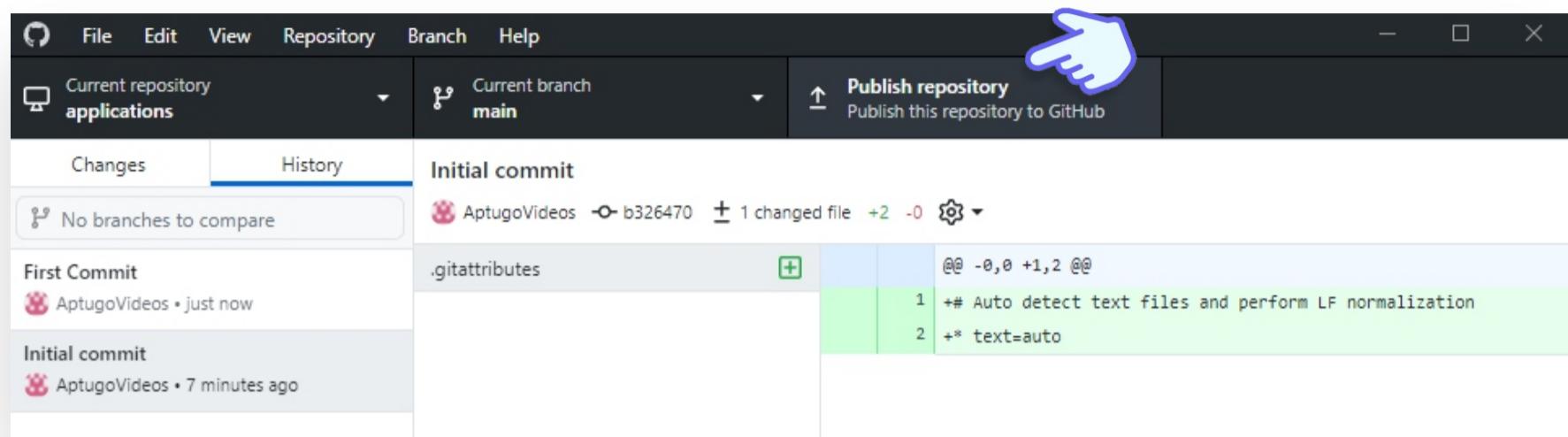
Un **commit** es el conjunto de código de la aplicación vinculada en un punto específico. Se trata de todo lo que la constituye hasta el momento en el que realizamos la copia de respaldo. Siguiendo la línea del control de versiones, es una versión del proyecto que se corresponde con el estado de la aplicación al momento de hacerlo.

Cada modificación que creamos dentro de nuestro proyectos aparecerá en la solapa “**changes**”.

Podemos ir guardando las modificaciones en commits, que funcionan como distintas versiones de nuestro proyecto.

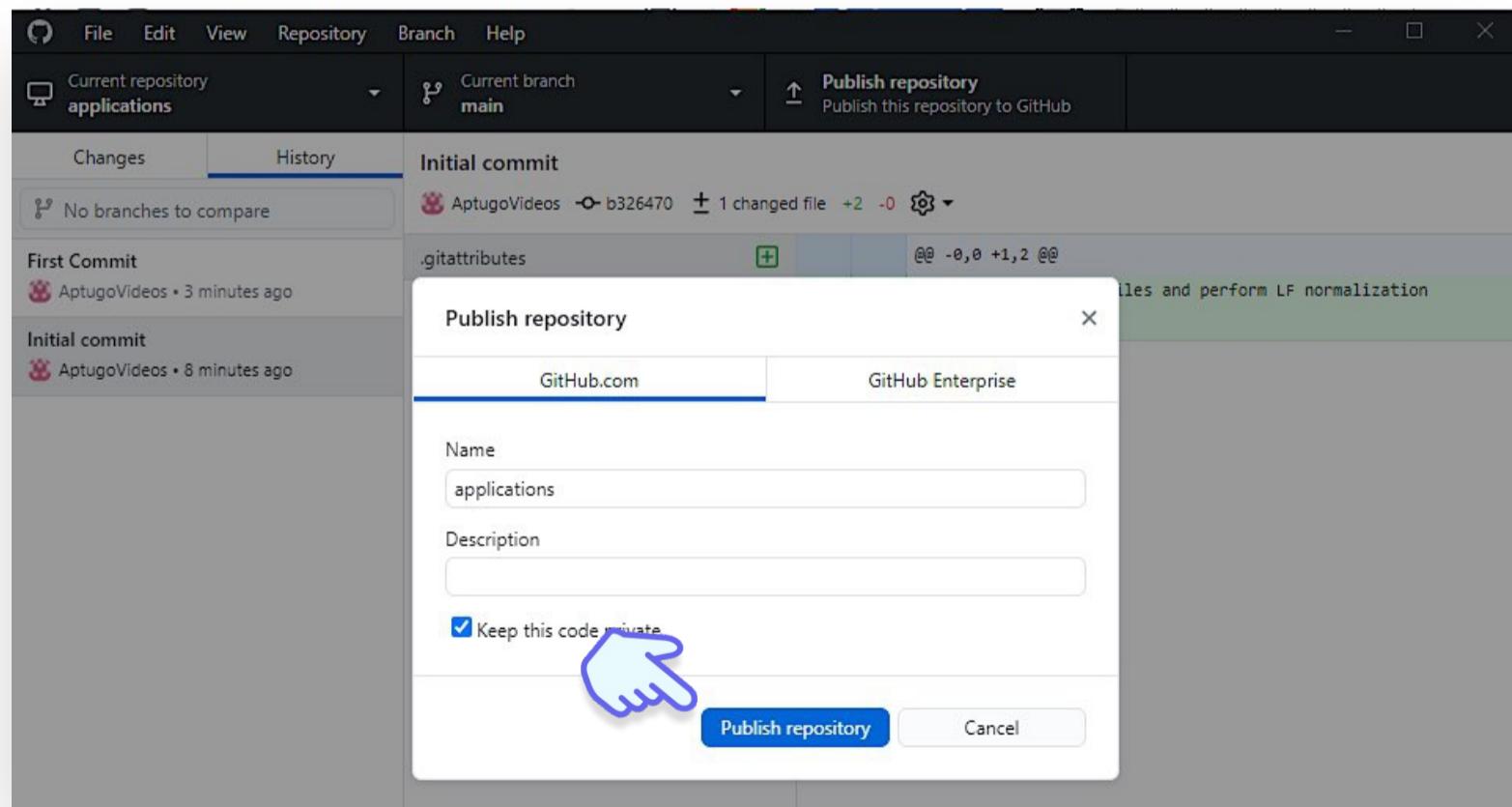
The screenshot shows the GitHub desktop application interface. At the top, there's a toolbar with File, Edit, View, Repository, Branch, and Help. Below that, it says "Current repository: applications" and "Current branch: main". On the right, there's a "Publish repository" button. The main area has tabs for "Changes" (32) and "History". Under "Changes", a blue hand icon points to the "Summary (required)" section where "HolaMundo\Pages\9JmqOsZA.json" and "HolaMundo\Pages\CDzKniA1.json" are selected. A red arrow points from the "Summary" section down to the "Commit to main" button. Below this, it says "Committed 3 minutes ago" and "Initial commit". To the right, under "First Commit", there's a text box containing "Primer commit (Descripción de las actualizaciones que se hicieron)". Another "Commit to main" button is at the bottom. It also says "Committed 4 minutes ago" and "Initial commit".

Una vez guardadas estas versiones, aparecerán en la **solapa History**, dejándonos lugar a poder realizar nuevas modificaciones.



¡Ahora es tiempo de publicar nuestros repositorios! 

Al hacer click en Publish repository, todos los commits que hayamos realizado se **subirán a nuestro Github**, primero debemos **colocarle un nombre al repositorio** y hacer click en el botón **Publish repository**:



Tardará apenas unos segundos en subirlo (dependiendo de cuantos proyectos tengamos y que tamaño tengan cada uno). **Una vez logrado esto, en el navegador podremos visualizar nuestros repositorios.**

The screenshot shows a GitHub profile page for a user named 'AptugoVideos'. At the top, there's a search bar and navigation links for 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. Below the search bar, there are tabs for 'Overview', 'Repositories 1', 'Projects', 'Packages', and 'Stars'. A large circular profile picture is displayed, followed by a repository card for 'applications' (Private). The repository was updated 25 seconds ago. There are buttons for 'Star' and 'New'. On the left, there's a sidebar with the user's name and a 'Edit profile' button.

En este caso, **si hacemos click en el repositorio applications** veremos su contenido el cual podremos descargar nuevamente si lo necesitamos por algún motivo.

The screenshot shows the detailed view of the 'applications' repository. At the top, there's a header with the repository name 'AptugoVideos / applications' (Private), and buttons for 'Unwatch', 'Fork', and 'Star'. Below the header, there are navigation links for 'Code', 'Issues', 'Pull requests', 'Actions', 'Projects', 'Security', 'Insights', and 'Settings'. The 'Code' tab is selected. On the left, there's a sidebar with 'main' branch information, '1 branch', '0 tags', and a 'Code' dropdown. The main area displays three commits: 'AptugoVideos First Commit' (922cebf, 5 minutes ago, 2 commits), 'HolaMundo' (First Commit, 5 minutes ago), and '.gitattributes' (Initial commit, 11 minutes ago). To the right, there's an 'About' section with a note 'No description, website, or topics provided.', and statistics: '0 stars', '1 watching', and '0 forks'. At the bottom, there's a section to 'Add a README' and a 'Releases' section indicating 'No releases published'.

# Conclusiones:

**Github es una de las múltiples herramientas que tenemos a disposición para respaldar nuestros proyectos.** Es de acceso libre, fácil de usar y altamente funcional. Es una buena práctica el generar versiones de los proyectos que realicemos, nos ahorrará tiempo y preocupaciones ante cualquier inconveniente.

**Recalcamos la importancia de reservar especialmente la carpeta del proyecto que se encuentra en Applications.** Aptugo se basa en ella para generar las compilaciones, que luego se reflejarán en la aplicación funcional.

**El sentido es unidireccional: desde applications es posible hacer builds, pero no viceversa.** Cualquier daño en dicha carpeta implica la imposibilidad de realizar cambios o volver a generar la aplicación. Podremos iniciar el proyecto, más no intervenirlo, lo cual significa una gran pérdida.



¡Prevenir antes que lamentar! **Tomemos el hábito de realizar copias de resguardo mediante estas plataformas.**