



Uso de Git y GitHub

≡ Tags

Lab

Introducción a Git

¿Qué es Git?

Características

Instalar Git en tu computadora

Comandos más comunes de Git

Introducción a GitHub

¿Qué es GitHub?

Cuenta GitHub

INFORMACIÓN IMPORTANTE

Configuración de Credenciales

EJERCICIO PROPUESTO

Eliminando Credenciales

Información Extra

Introducción a Git

¿Qué es Git?

Git es un sistema de control de versiones distribuido que permite a los desarrolladores trabajar con un repositorio completo de manera local y sincronizarlo con el servidor cuando sea necesario.

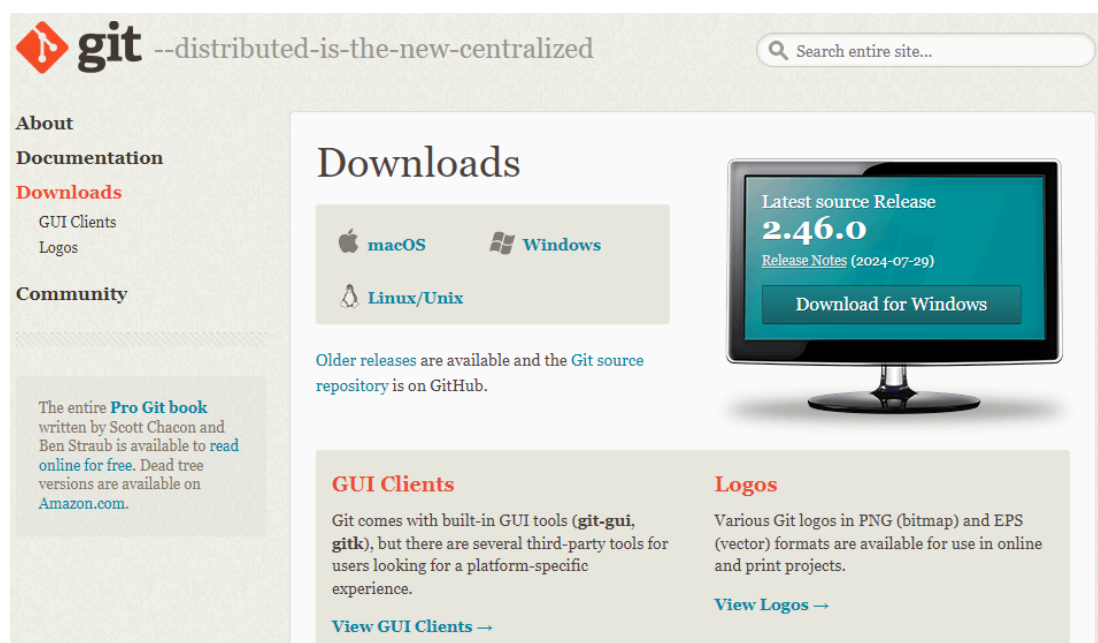
Características

- **Trabajo sin conexión:** A diferencia de los sistemas centralizados, Git permite trabajar localmente sin necesidad de una conexión constante al servidor.
- **Sincronización remota:** Facilita la sincronización entre repositorios locales y remotos, permitiendo una gestión eficiente del código.
- **Flexibilidad y Popularidad:** Su versatilidad y amplia adopción lo convierten en una excelente opción para equipos de desarrollo.
- **Amplia comunidad y recursos:** La gran comunidad de usuarios y la disponibilidad de recursos de capacitación hacen que sea fácil aprender y obtener ayuda.
- **Compatibilidad:** Es compatible con la mayoría de los entornos de desarrollo y sistemas operativos.

Instalar Git en tu computadora

▼ Guía de instalación

1. Accede al siguiente enlace [descargar Git](#).
2. Elige el sistema operativo que estes utilizando.

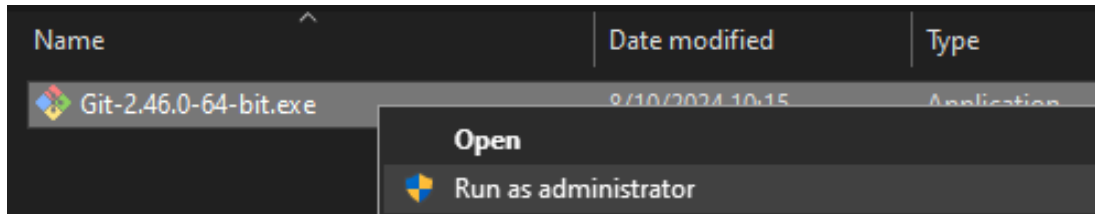


3. Descarga la última versión disponible.

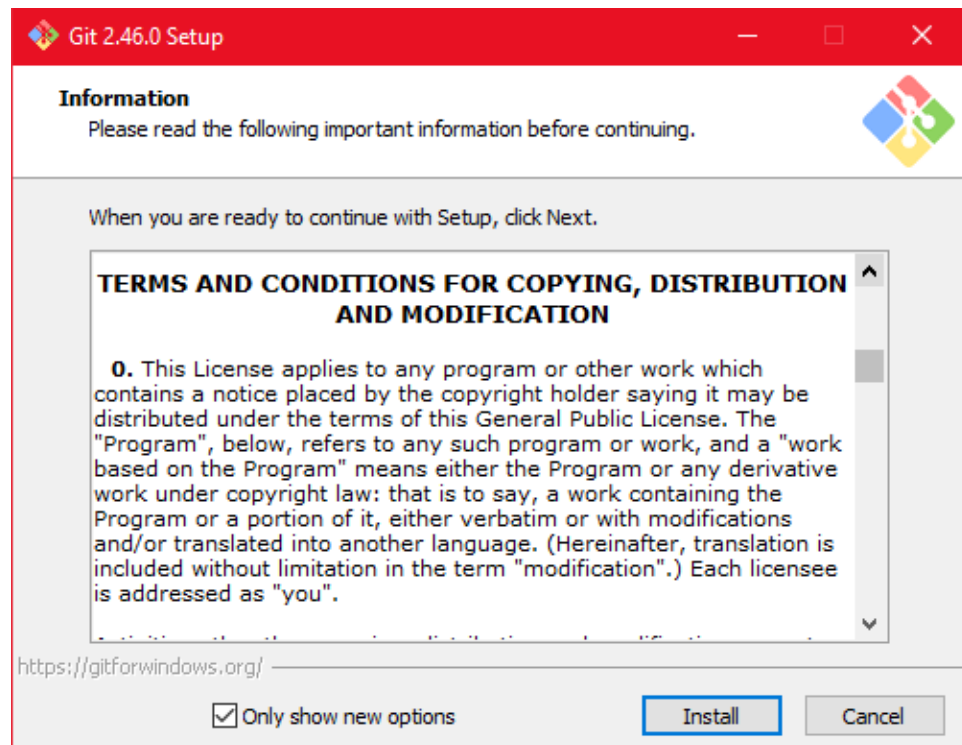
Download for Windows

[Click here to download](#) the latest (2.46.0) 64-bit version of Git for Windows. This is the most recent [maintained build](#). It was released **12 days ago**, on 2024-07-29.

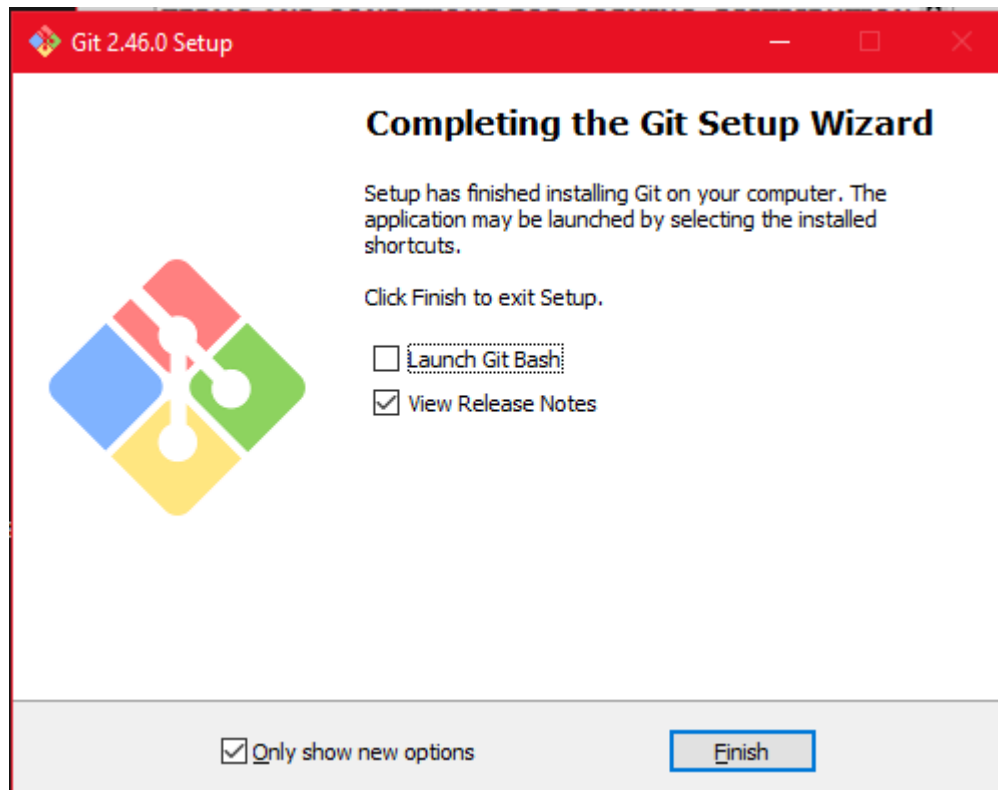
4. Una vez descargado, ejecuta el instalador como Administrador.



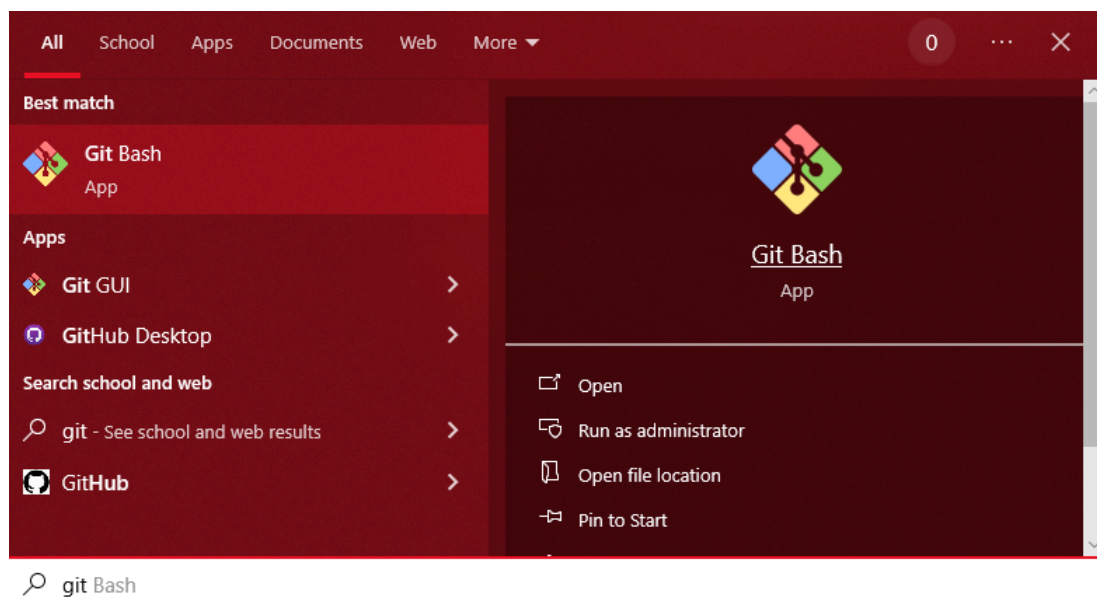
5. Instalas Git.



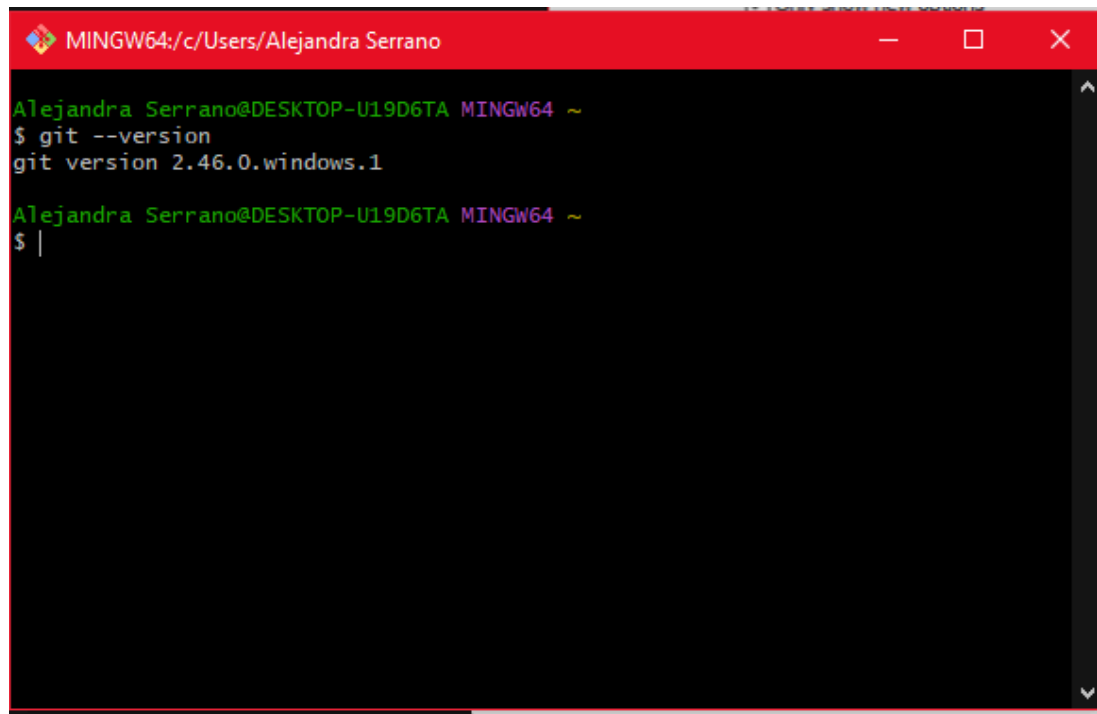
6. Le das al botón que dice Finish.



7. Dirígete a la barra de tareas, busca Git bash y ejecútala.



8. Realiza el comando **git -version** y así ya tienes instalado Git correctamente.

A screenshot of a Windows terminal window with a red title bar. The title bar text is "MINGW64:/c/Users/Alejandra Serrano". The terminal shows the command prompt "Alejandra Serrano@DESKTOP-U19D6TA MINGW64 ~" followed by the command "\$ git --version" and its output "git version 2.46.0.windows.1". Below that, the prompt is shown again with a cursor: "Alejandra Serrano@DESKTOP-U19D6TA MINGW64 ~ \$ |".

```
MINGW64:/c/Users/Alejandra Serrano

Alejandra Serrano@DESKTOP-U19D6TA MINGW64 ~
$ git --version
git version 2.46.0.windows.1

Alejandra Serrano@DESKTOP-U19D6TA MINGW64 ~
$ |
```

Comandos más comunes de Git

Comando	Descripción
<code>git clone <url></code>	Clona un repositorio remoto en tu máquina local. Copia todos los archivos del repositorio, así como su historial de commits.
<code>git add <archivo></code>	Añade un archivo o cambios específicos al área de preparación (stage), preparándolo para ser confirmado (commit).
<code>git commit -m "mensaje"</code>	Confirma los cambios en el área de preparación y guarda una instantánea de ellos en el historial del repositorio. El mensaje describe los cambios realizados.
<code>git push origin <rama></code>	Envía tus commits locales a la rama especificada del repositorio remoto. Por ejemplo, <code>git push origin main</code> enviará tus cambios a la rama <code>main</code> en el remoto.
<code>git merge <rama></code>	Combina la rama especificada con la rama actual. Integra los cambios de otra rama en la rama en la que te encuentras.
<code>git fetch</code>	Descarga los cambios más recientes desde el repositorio remoto al repositorio local sin aplicar esos cambios a la rama actual.

Introducción a GitHub

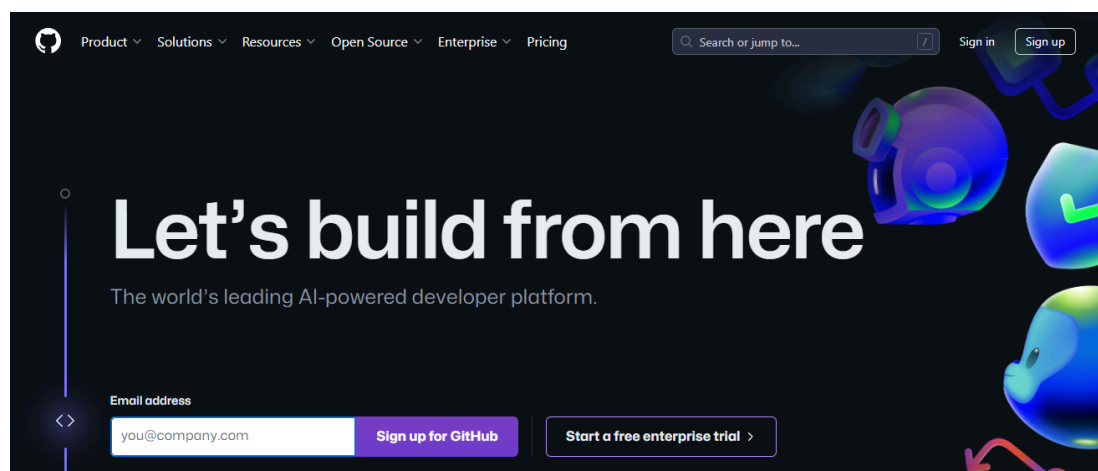
¿Qué es GitHub?

GitHub es una plataforma en línea que usa el sistema de control de versiones Git para facilitar el alojamiento, colaboración y seguimiento de proyectos de software. Permite a los desarrolladores almacenar, compartir y gestionar su código, así como trabajar en equipo. Los proyectos se organizan en repositorios, que pueden ser públicos o privados según la preferencia del propietario.

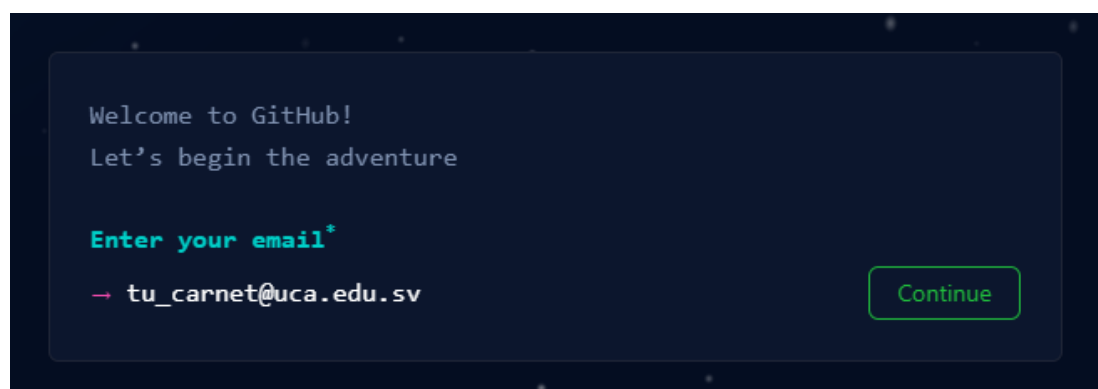
Cuenta GitHub

▼ Creación cuenta de GitHub

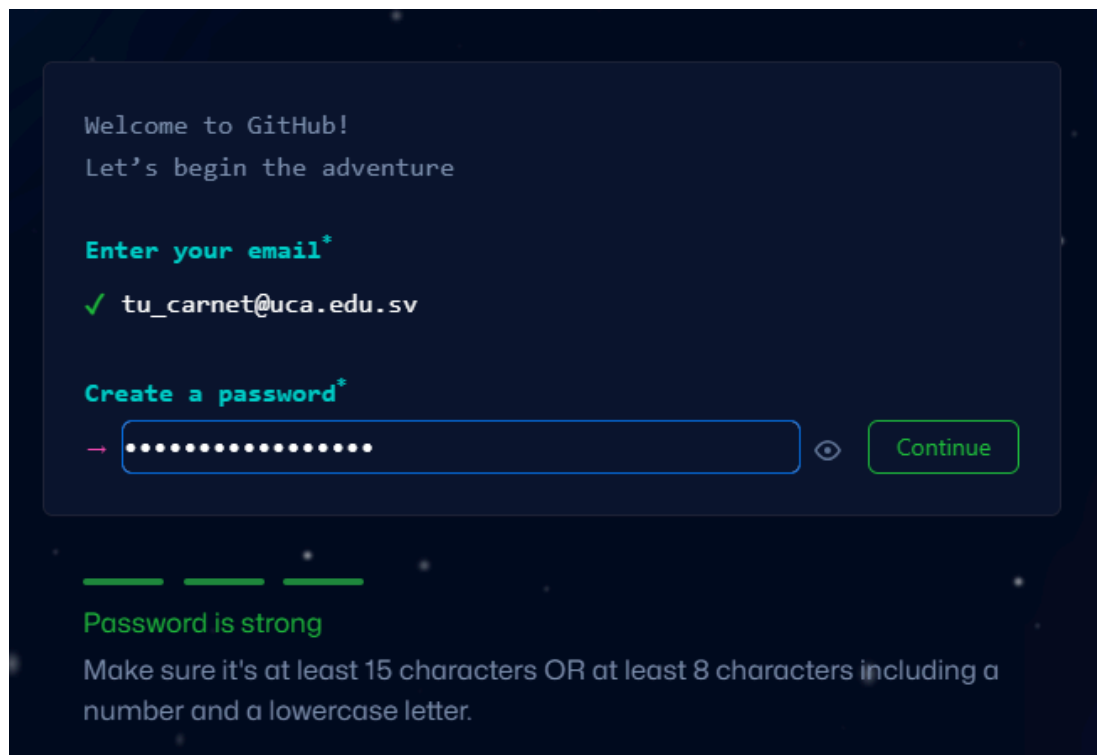
1. Accede a [github](https://github.com).
2. Dar clic en la parte superior derecha donde dice **Sign Up**.



3. Coloca tu dirección de correo electrónico institucional.



4. Agrega una contraseña que sea segura (8 caracteres, una mayúscula, número y un carácter especial) y que no se te olvide.



Welcome to GitHub!
Let's begin the adventure

Enter your email*

✓ tu_carnet@uca.edu.sv

Create a password*

→ ☐ Continue

Password is strong

Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter.

5. Elige un nombre de usuario que te identifique tomando en cuenta que esta cuenta la usarás en toda tu carrera.



Welcome to GitHub!
Let's begin the adventure

Enter your email*

✓ tu_carnet@uca.edu.sv

Create a password*

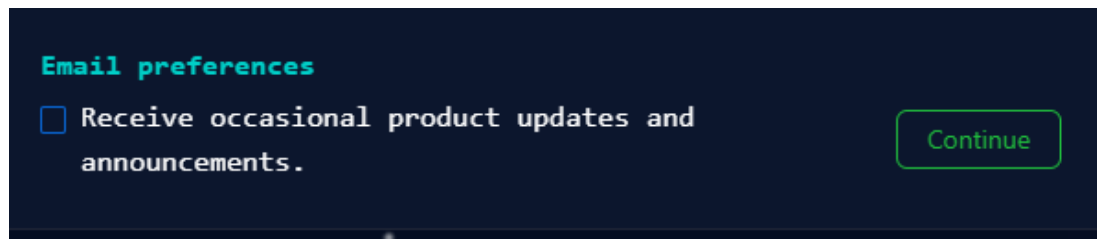
✓

Enter a username*

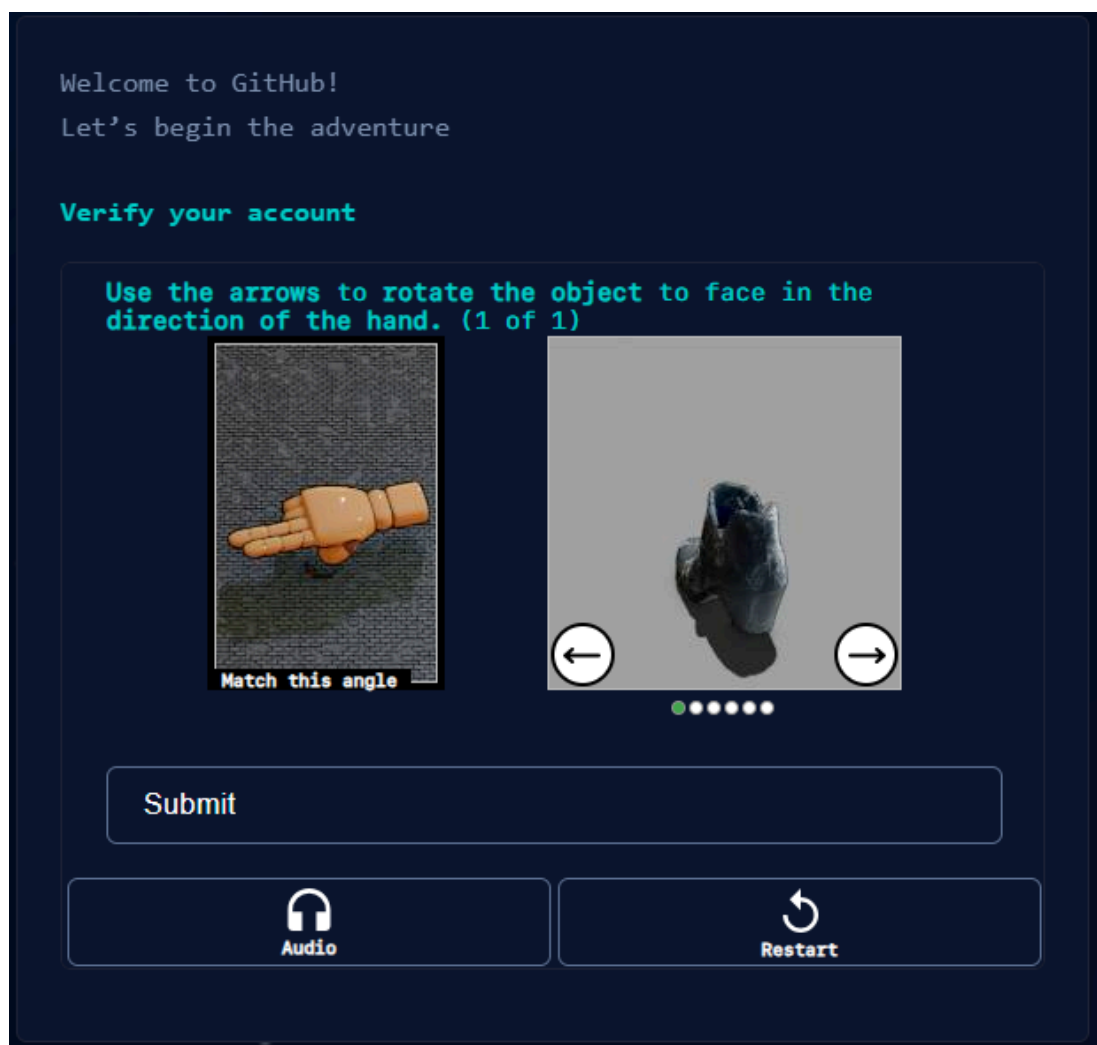
→ MRivas43 Continue

MRivas43 is available.

6. Realmente no es necesario marcar este cuadro, pueden dar a continuar.



7. Resolver el Captcha si es necesario, caso contrario pueden pasar por alto este paso



8. Verificar el correo electrónico con el código de verificación que Github envía.



9. Elegir la siguiente configuración y darle al botón continuar.
10. En este punto pueden no seleccionar nada pero se sugiere seleccionar "Collaborate Coding" y dar clic en continuar.
11. Elegir la versión gratis. (en futuras materias se les explicará como usar el Student Pack)
12. Si seguiste correctamente los pasos, felicidades ya tienes tu cuenta de Github.

INFORMACIÓN IMPORTANTE

Para las distintas evaluaciones del ciclo se necesita lo siguiente en su cuenta de GitHub:

- Cambiar foto de perfil (*Una de ustedes, nada de juegos, animes, películas, series, etc.*)
- Agregar nombre completo (*Este nombre es aparte de su **Username**, lo podrán encontrar en **Settings** → **Public** profile → **Name**)*
- Agregar una biografía breve



El incumplimiento de cualquiera de estas indicaciones podrá afectar su nota en futuras evaluaciones

Configuración de Credenciales



Para la realización del **EJERCICIO PROPUESTO** necesitamos configurar nuestras credenciales de la siguiente manera:

1. Abre la consola de git dando clic derecho en el **Escritorio** y luego seleccionamos la opción **Open Git Bash Here** .
2. Copia y pega el siguiente comando `git config user.name "Nombre de usuario"` *(Por ejemplo: `git config user.name "TúNombreDeUsuario"`)*
3. Luego copia y pega el siguiente comando `git config user.email carnet@uca.edu.sv`

EJERCICIO PROPUESTO



Realizar los siguiente pasos:

1. Crear un repositorio público o privado llamado **MiPrimerRepo_NúmeroCarnet** *(Por ejemplo `MiPrimerRepo_00083522`)*. **Al momento de crear su repositorio, verificar que la opción `Add README`, esté marcada.**
2. Clonar repositorio en el Escritorio *(para este punto y los siguientes, hacer uso de comando de Git, revisar cuadro de comandos)*
3. Agregar un archivo de texto con nombre **Mi_Texto.txt**
4. Realizar un commit *(el mensaje de preferencia con un emoji por ejemplo: `":memo: Add: MiPrimerCommit"`)*
5. Hacer push
6. Agregar una imagen cualquiera con nombre **Mi_Imagen.jpg**
7. Realizar un commit *(el mensaje de preferencia con un emoji por ejemplo: `":memo: Add: MiSegundoCommit"`)*
8. Hacer push

Eliminando Credenciales



Una vez, terminada la práctica, habrá que eliminar las credenciales. Para eliminar las credenciales de Git que configuramos previamente, sigue los siguientes pasos:

1. **Abre la consola de Git** de la misma manera en que lo hiciste para configurar las credenciales, es decir, haciendo clic derecho en el **Escritorio** y seleccionando la opción **Open Git Bash Here**.

2. **Elimina el nombre de usuario** utilizando el siguiente comando:

```
git config --unset user.name
```

3. **Elimina el correo electrónico** utilizando el siguiente comando:

```
git config --unset user.email
```

4. **Verifica que las credenciales han sido eliminadas** utilizando los comandos:

```
git config user.name
```

```
git config user.email
```



Si ambos comandos no devuelven ningún resultado, significa que las credenciales han sido eliminadas correctamente.

Información Extra

[Página para practicar comandos de GIT.](#)

[Gitmoji.](#)