

## Configuraciones iniciales de GitHub

Para comenzar a utilizar GitHub, lo primero que necesitas es crear una cuenta.

A continuación te explicamos cómo hacerlo paso a paso:

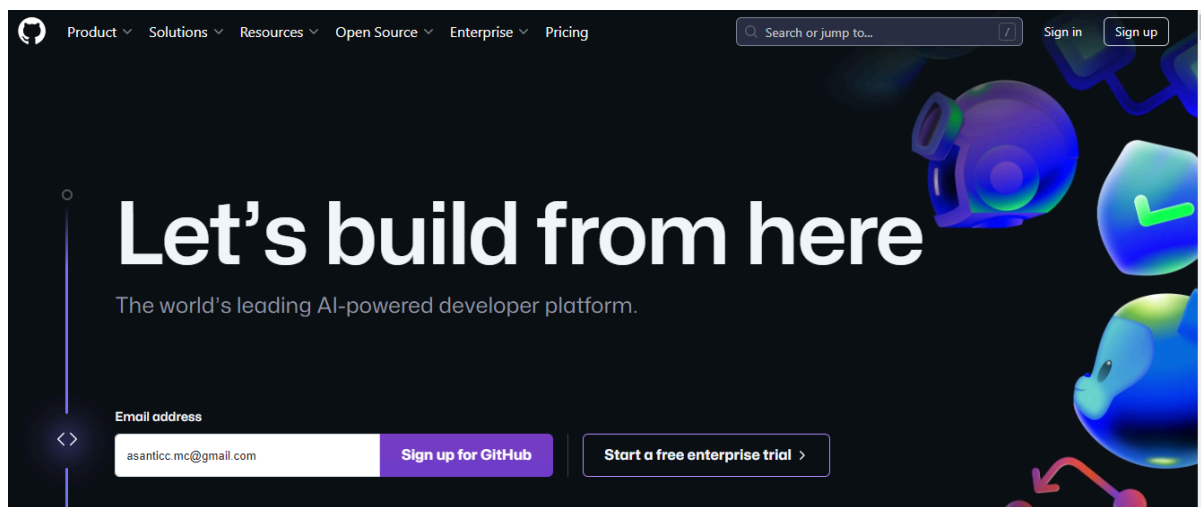
### Registrarse en GitHub

- Visita el sitio web oficial de [GitHub](https://github.com).

Verás una pantalla de registro. Completa los siguientes campos:

1. Email: Ingresa tu dirección de correo electrónico.
2. Username: Crea un nombre de usuario único.
3. Password: Establece una contraseña segura.

Una vez completado, haz clic en el botón **Sign up for GitHub** (Inscribirse) y sigue las instrucciones adicionales para finalizar la creación de tu cuenta.



### Verificación y acceso a tu cuenta de GitHub

Una vez que hayas completado todos los pasos de registro con tus datos personales, GitHub te enviará un correo electrónico de verificación. Sigue estos pasos:

**1. Verificación:**

- Ingresa en tu correo electrónico y haz clic en el botón "Verificar" que encontrarás en el mensaje de GitHub.

**2. Acceso a tu cuenta:**

- Después de la verificación, puedes iniciar sesión en tu cuenta utilizando tu correo y contraseña.

**3. Elección de plan:**

- Al ingresar, si GitHub te pide elegir un plan, selecciona el plan Free (Gratis).

**4. Propósito de uso:**

- Si se te pregunta sobre el propósito de uso, puedes seleccionar Estudiante o Uso personal.

**Establezca una conexión SSH con GitHub**

Para intercambiar datos de forma segura entre tu PC y GitHub, se requiere un procedimiento de autenticación.

El procedimiento de autenticación utiliza el método de autenticación SSH llamado autenticación de clave pública.

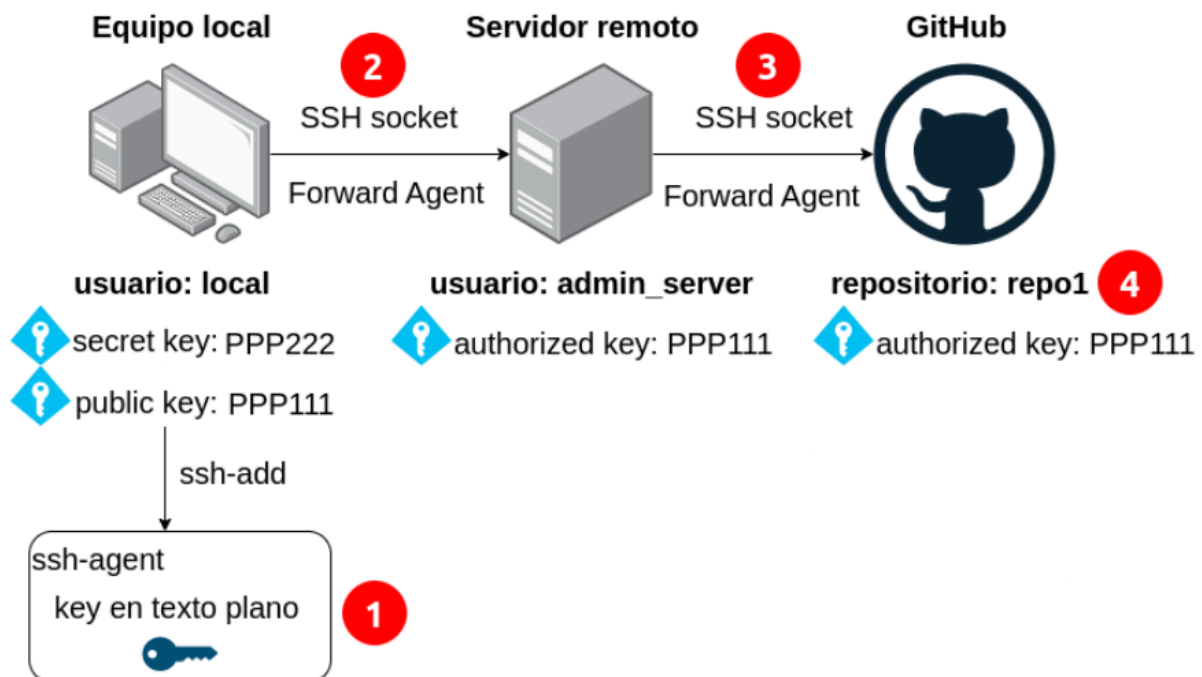
Se omiten las explicaciones detalladas sobre la autenticación mediante clave pública y SSH. Para más información, consulte [Conectarse a GitHub mediante SSH](#) en la documentación oficial.

**Flujo de esquemas de autenticación de clave pública**

El flujo del esquema de autenticación de clave pública es el siguiente.

[Método de autenticación].

1. En el lado del cliente, crea un par de claves, la privada y la pública, y se coloca la clave pública en el lado del servidor.
2. El cliente solicita acceso al servidor
3. El servidor transmite la información cifrada al cliente utilizando la clave pública.
4. El cliente descifra el cifrado recibido con su clave privada y devuelve la información al servidor.
5. El servidor comprueba que la información coincide y se completa la autenticación.



En este caso, el "servidor" corresponde a GitHub.

### Crear pares de claves privadas y públicas.

primero configura tu nombre de usuario y email con los siguientes comandos:

**\$ git config --global user.name "puedes poner como llamaste a tu pc"**

**\$ \$ git config --global user.email "puedes poner el gmail que utilizaste para github"**

Cree las claves privada y pública. Ejecuta el siguiente comando en una terminal.

Para "your\_email@example.com", pon tu dirección de correo electrónico que utilizaste para registrarte en GitHub.

O puede ver el video explicativo de:

[Configura Git y GitHub en WSL Ubuntu con Clave SSH: Guía Completa](#)

```
$ ssh-keygen -t rsa -C "your_email@example.com"
```

Normalmente, ejecutar el comando ssh-keygen crea automáticamente un directorio .ssh.

Ejemplo de cómo se genera:

```
$ ssh-keygen -t rsa -C "your_email@example.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/Users/user/.ssh/id_rsa):
```

Se le preguntará dónde guardar la clave que ha creado. Presione **Enter** en el teclado.

A continuación, se le pedirá que introduzca una frase de contraseña, que debe ser diferente de la contraseña de tu cuenta de GitHub.

Deberás introducirla dos veces para confirmar.

Por motivos de seguridad, no se mostrará en pantalla aunque la introduzcas. **(Aunque no es necesario también esta contraseña)** solo darle a **Enter**, **Enter**.

```
Enter passphrase (empty for no passphrase):
```

```
Enter same passphrase again:
```

En el transcurso de este trabajo, se ha creado un par de claves, privada y pública, dentro del directorio `.ssh` que se encuentra en el directorio de inicio del usuario.

El directorio `.ssh` es un directorio oculto que existe en el directorio de inicio del usuario y está destinado a almacenar archivos relacionados con SSH (Secure Shell). Aquí se guardan las claves SSH (clave privada y clave pública), los archivos de configuración de SSH, los archivos de hosts conocidos, entre otros.

**Verificación del directorio `.ssh`**

Para verificar el directorio `.ssh`, ejecuta el siguiente comando:

```
$ ls -la ~/.ssh
```

Cuando se ejecute, `id_rsa` (clave privada) e `id_rsa.pub` (clave pública) se mostrarán como se muestra a continuación.

```
-rw----- 1 username group 1679 Jun 14 10:00 id_rsa
-rw-r--r-- 1 username group  400 Jun 14 10:00 id_rsa.pub
```

## Registro de clave pública

Registra tu clave pública en GitHub.

Vea y copie el contenido del archivo `id_rsa.pub`, donde se almacena la clave pública.

```
$ cat ~/.ssh/id_rsa.pub
```

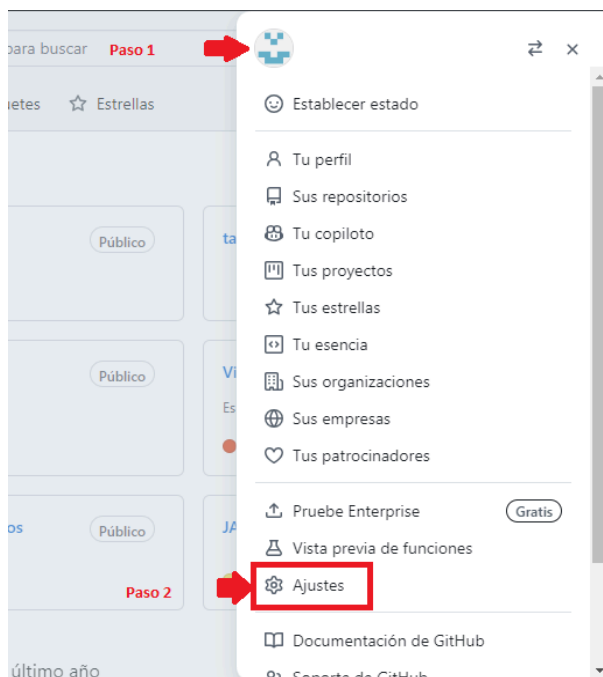
## Ejemplo de cómo se genera

```
$ cat ~/.ssh/id_rsa.pub
```

```
ssh-rsa BBFNDKDKNzaC1yc2•(Omitido porque es demasiado largo.)...
YVjCCDvXOLUfTetp your_email@example.com
```

Copia lo anterior de `ssh-rsa` al final de tu dirección de correo electrónico.

Configure la clave pública copiada en GitHub. **Haga clic en su imagen en la esquina superior derecha del encabezado de GitHub**, luego haga clic en **"Ajustes (Configuración)"**





Haz clic en '**Claves SSH y GPG**', y luego haz clic en '**Nueva Clave SSH**' en el lado derecho.





Su cuenta personal

 Perfil público


 Cuenta

 Apariencia


 Accesibilidad


 Notificaciones


Acceso


 Facturación y planes




 Correos electrónicos

 Contraseña y autenticación

 Sesiones

 Claves SSH y GPG

 Organizaciones

En el campo '**Título**' puedes escribir lo que desees, como el nombre de tu dispositivo, en **Tipo de clave**

Pon **Clave de autenticación**.

Pero en el campo '**Llave**' debes pegar lo que copiaste previamente.

Agregar nueva clave SSH

Título

Tipo de clave

Clave de autenticación

Llave

```
ssh-rsa AAAAB3NzaG1zb2EAAAADAQABAAQDAK1Sm1SeloYDS9PUA...a55AMuZgloF8m1ywlGm
iARCndHfPM0dQxAHV7Z7
98mLFafrFKS75Gjbcjq4so2
FGwkv2ma39o3/tMlaBoL
IPYTKWW1+VAtHfjPd2b2
i+Lqc9TUWK4Ympo3xZY8
@gmail.com
```

Agregar clave SSH

### Atención

Asegúrate de que la clave pública que pegaste empieza por `ssh-rsa` y termina con tu dirección de correo electrónico. Si la pegas incorrectamente, no podrás conectarte entre tu PC y GitHub.

Haga clic en el botón verde **"Agregar clave SSH"** para completar el registro.

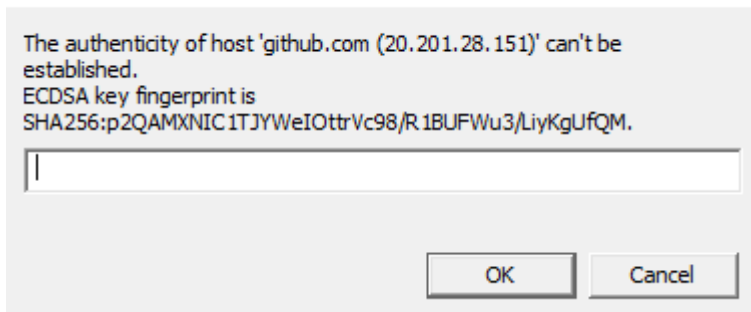
### Comprobación de la conexión entre tu PC y GitHub.

Ejecuta el comando de comprobación de la conexión.

```
$ ssh -T git@github.com
```

En caso que le aparezca la ventana emergente

Git for Windows



Añada el código que comienza con "SHA256:" y presione "OK":

```
SHA256:p2QAMXNIC1TJYWeIOttrVc98/R1BUFWu3/LiyKgUfQM
```

Es posible que a continuación se le pida que introduzca (yes) Sí o No, en cuyo caso introduzca Sí y pulse Enter.

Please type 'yes', 'no' or the fingerprint:

Cuando se te pida una contraseña(SI ES QUE PUSO), introduzca la que estableció al crear el par de claves privada/pública y pulse Enter.

Enter passphrase for key '/Users/Nombre de usuario/.ssh/id\_rsa':

Si la conexión es correcta se muestra el siguiente mensaje.

Hi Username! You've successfully authenticated, but GitHub does not provide shell access.

### **Configuración concluida**

## **Resumen**

- Utiliza el método de autenticación de clave pública para las conexiones SSH a GitHub.
- Se puede utilizar una cuenta de GitHub y un par de claves privada/pública para acceder a los repositorios de GitHub.

### **Documento Oficial**

[1. Conéctate a GitHub mediante SSH](#)