Configurar llaves SSH para Git y Github

Contenido

- Objetivo de usar ssh
- Generar una nueva llave
- Agregar la nueva llave al servicio de ssh-agent
- Registrar la llave pública a tu cuenta de GitHub

¿Por qué usar SSH?

- Intercepción de la comunicación entre dos sistemas; un tercero en algún lugar de la red entre entidades en comunicación hace una copia de la información que pasa entre ellas. La parte interceptora puede interceptar y conservar la información, o puede modificar la información y luego enviarla al recipiente al cual estaba destinada.
- Personificación de un determinado host: un sistema interceptor finge ser el receptor a quien está destinado un mensaje. Si funciona la estrategia, el cliente no se da cuenta del engaño y continúa la comunicación con el interceptor.

¿Para que necesitamos la criptografía asimétrica?

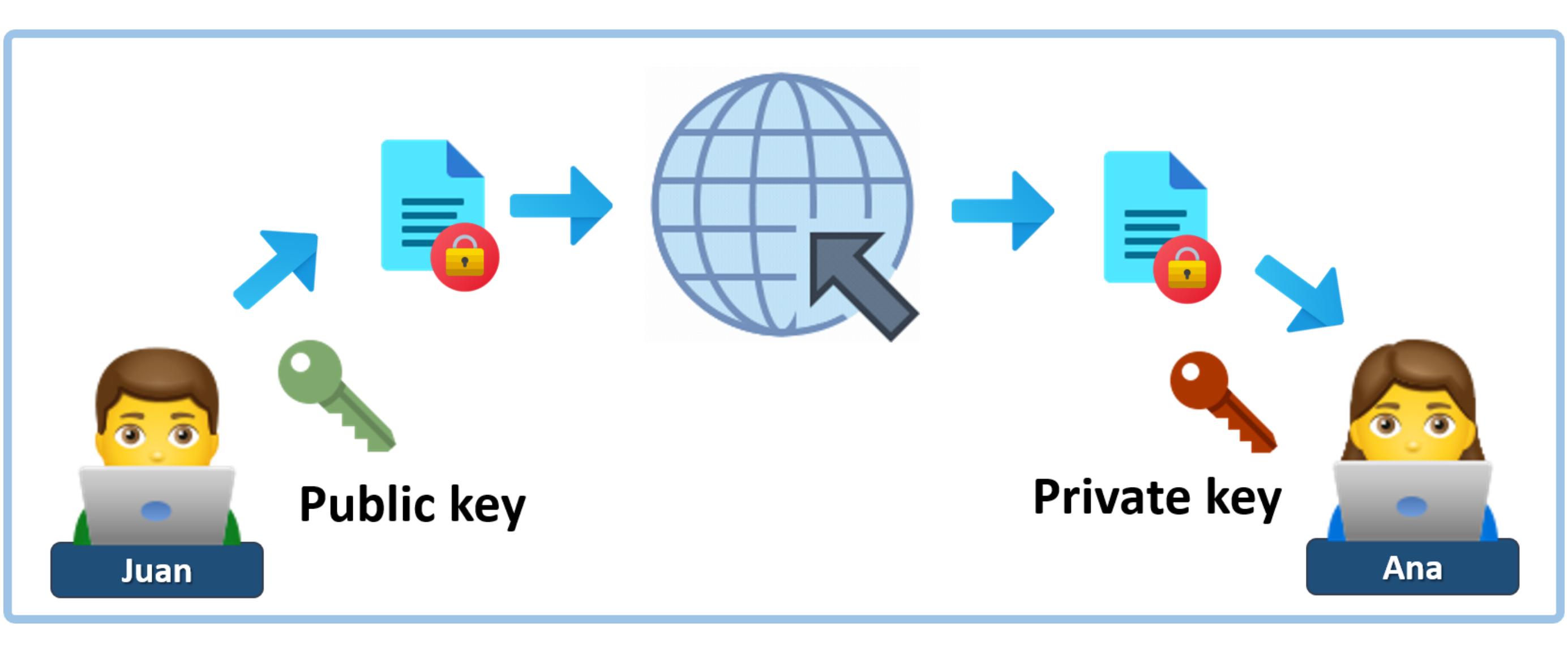
Cuando enviamos datos por internet, ya sea una imagen, un archivo o sólo mensajes, corremos el riesgos de que nos roben nuestra información en el intento, antes de que llegue al receptor.

¿Qué es la criptografía asimétrica?

La criptografía asimétrica es la forma segura de enviar y recibir un mensaje, ya que si incluso llega a ser interceptado, nunca podrá leer el mensaje.

¿Como funciona?

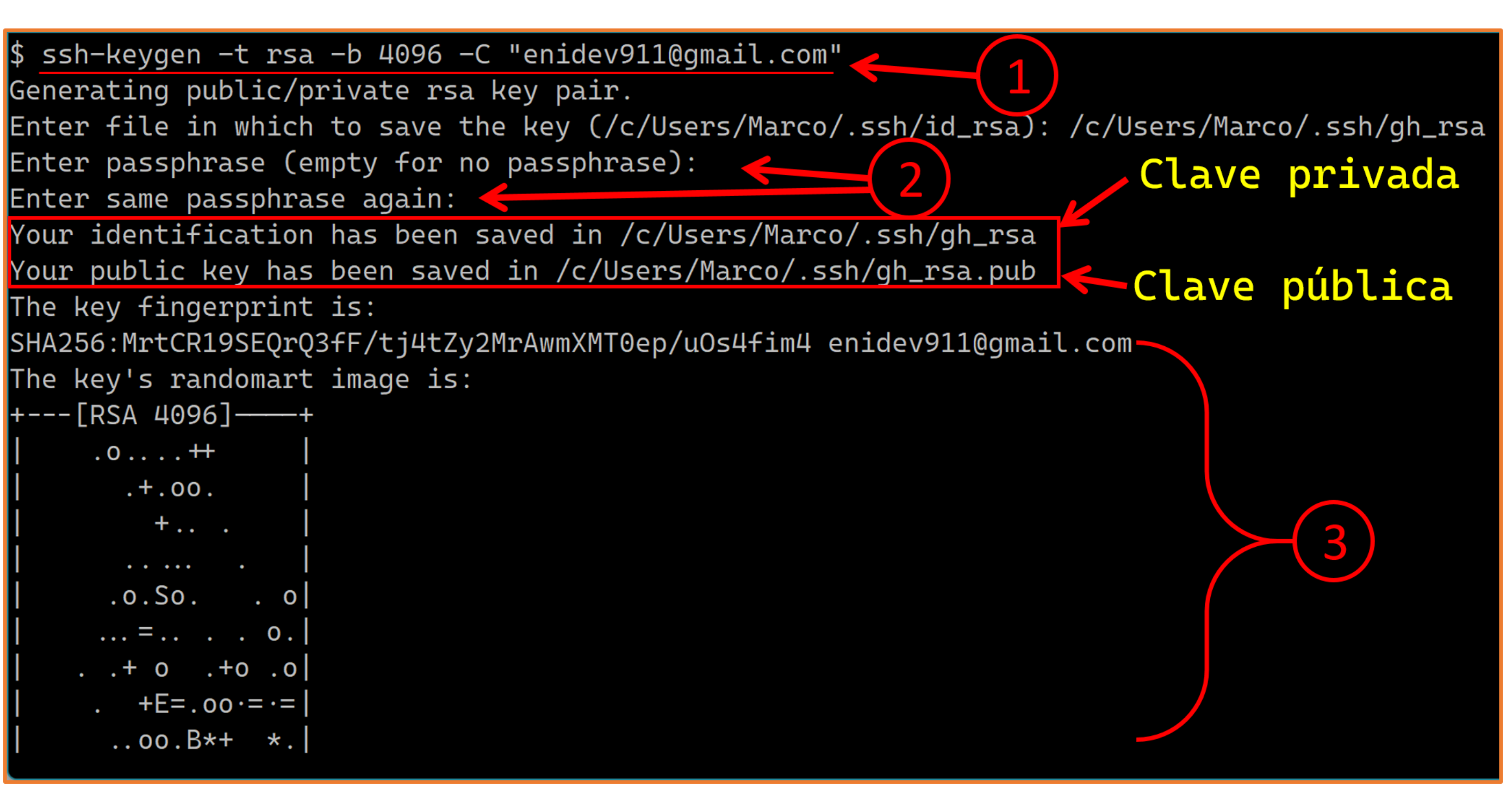
En este caso, Juan le quiere mandar un mensaje a Ana, ambos cuentan con una llave pública y una privada que están ligadas con un algoritmo matemático, Ana le da a Juan su llave pública y Ana conserva su llave privada, Juan manda su mensaje cifrado con la llave pública que le proporciona Ana, Ana recibe el mensaje de Juan y lo desencripta con su llave privada de esta manera Ana y sólo Ana podrá leer el mensaje que le envió Juan.



Generar par de clave pública y privada.

Para generar los pares de claves se empleará la herramienta denominada `ssh-keygen`. Este programa está disponible por medio de la línea de comandos para Linux, MacOSX y en `Git Bash` en Windows, puede generar una clave SSH.

Ejemplo:



```
$ ssh-keygen -t rsa -b 4096 -C "enidev911@gmail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/Marco/.ssh/id_rsa): /c/Users/Marco/.ssh/gh_rsa
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/Marco/.ssh/gh_rsa
Your public key has been saved in /c/Users/Marco/.ssh/gh_rsa.pub
The key fingerprint is:
SHA256:MrtCR19SEQrQ3fF/tj4tZy2MrAwmXMT0ep/uOs4fim4 enidev911@gmail.com
The key's randomart image is:
+---[RSA 4096]----+
|     .o....++    |
|       .+.oo.    |
|         +.. .   |
|        .. ...  .|
|      .o.So.   .o|
|     ...=..  . o.|
|   . .+ o  .+o .o|
|    .  +E=.oo·=·=|
|      ..oo.B*+  *.|
```

Explicación:

1. Comience creando una nueva clave, usando su correo electrónico como etiqueta:

```
ssh-keygen -t rsa -b 4096 -C "user@email.com"
```

- **-t rsa**: Aquí especificamos el algoritmo de encriptación que queremos en este caso el más común es **rsa**
- **-b 4096**: Aquí especificamos que tan compleja es la llave.
- **-C** `"user@email.com"`: Aquí colocamos el email con el que se va a configurar la llave.

Se le pedirá lo siguiente el destino donde se almacenará la nueva llave generada (Si solo presiona `Enter` por lo general se almacena en el directorio del usuario en una subcarpeta oculta):

```
Enter file in which to save the key (/c/Users/user/.ssh/id_rsa):
```

Seleccione una ubicación de archivo o presione `Enter` se creara en la dirección por defecto en mi caso voy a darle un nombre descriptivo como **/c/Users/Marco/.ssh/gh_rsa** (gh: github):

2. Ingresar una frase de contraseña segura creará una capa adicional de seguridad. Evitar que cualquier persona que obtenga acceso a la pc use esa clave sin la frase de contraseña. Sin embargo, será necesario que proporcione la frase de contraseña cada vez que se utilice la clave SSH.

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

3. Nos devuelve el hash generado para la nueva llave.

Agregar al SSH-Agent

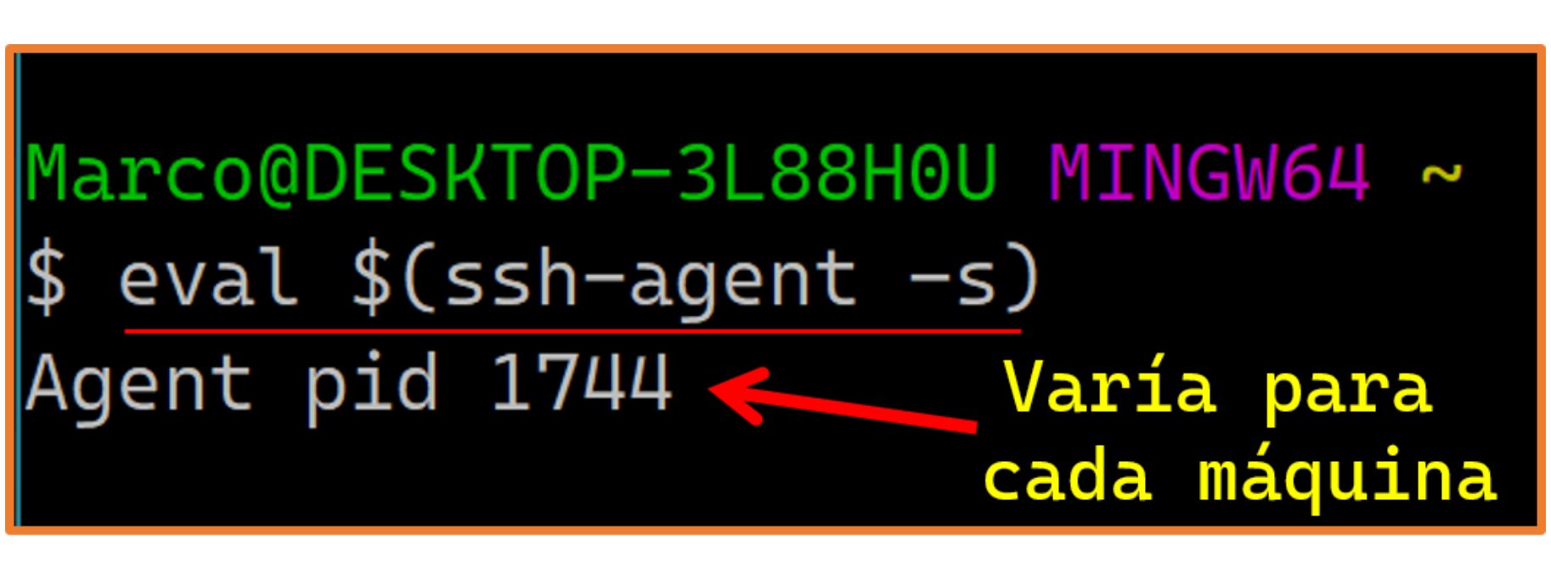
El agente ssh (ssh-agent) es un programa auxiliar que realiza un seguimiento de las claves de identidad del usuario y sus frases de contraseñas. El agente puede usar las claves para iniciar sesión en otros servidores sin que el usuario escriba nuevamente una contraseña o frase de contraseña. En palabras simples podemos decir que se trata de un programa para contener llaves privadas, utilizadas para la autenticación de claves públicas. Antes de agregar una nueva clave SSH al servicio de ssh-Agent para gestionar tus claves, debes haber comprobado las claves SSH existente y **generado una nueva clave SSH** y verificar que se esté ejecutando el servicio.

1. Verifica que el ssh-agent se esté ejecutando.

```
# correr el ssh-agent en segundo plano
eval $(ssh-agent -s)
> Agent pid 1412
```

- **eval**: Ejecuta un comando de shell y efectua una doble evaluación en la línea de comandos.

Ejemplo:

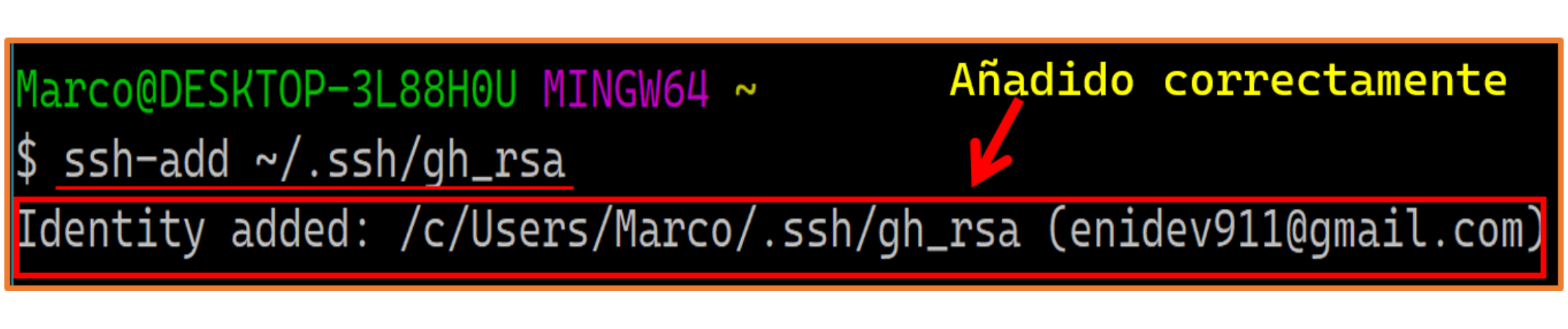


```
Marco@DESKTOP-3L88H0U MINGW64 ~
$ eval $(ssh-agent -s)
Agent pid 1744
```

2. Agrega tu llave privada SSH al ssh-agent. Si creaste tu llave con un nombre distinto, o si estás agregando una llave existente que tiene un nombre distinto, remplaza **gh_rsa** en el comando con el nombre de tu llave.

```
ssh-add /Users/user/.ssh/gh_rsa
```

Ejemplo:



```
Marco@DESKTOP-3L88H0U MINGW64 ~
$ ssh-add ~/.ssh/gh_rsa
Identity added: /c/Users/Marco/.ssh/gh_rsa (enidev911@gmail.com)
```

Ahora el par de claves SSH está listo para usarse.

Agregar clave pública SSH nueva a tu cuenta de GitHub.

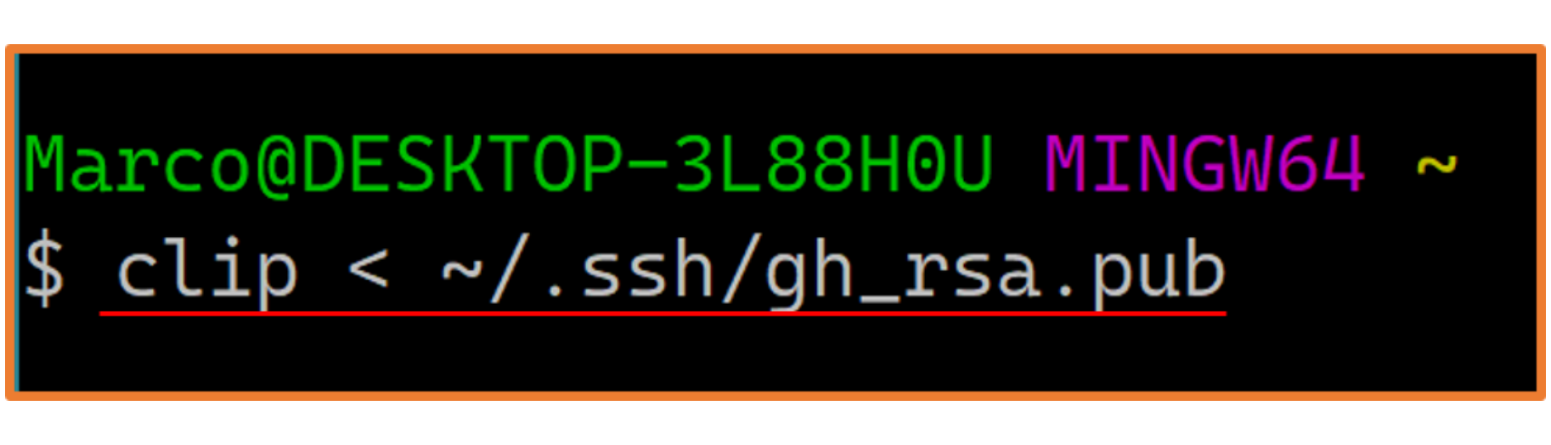
1. Copia la clave SSH a tu portapapeles.

Si tu archivo de clave SSH tiene un nombre diferente al código del ejemplo, modifica el nombre de archivo. Al copiar tu clave, no agregues líneas nuevas o espacios en blanco.

```
# copiar el contenido de la clave pública al portapapeles.
clip < ~/.ssh/gh_rsa.pub
```

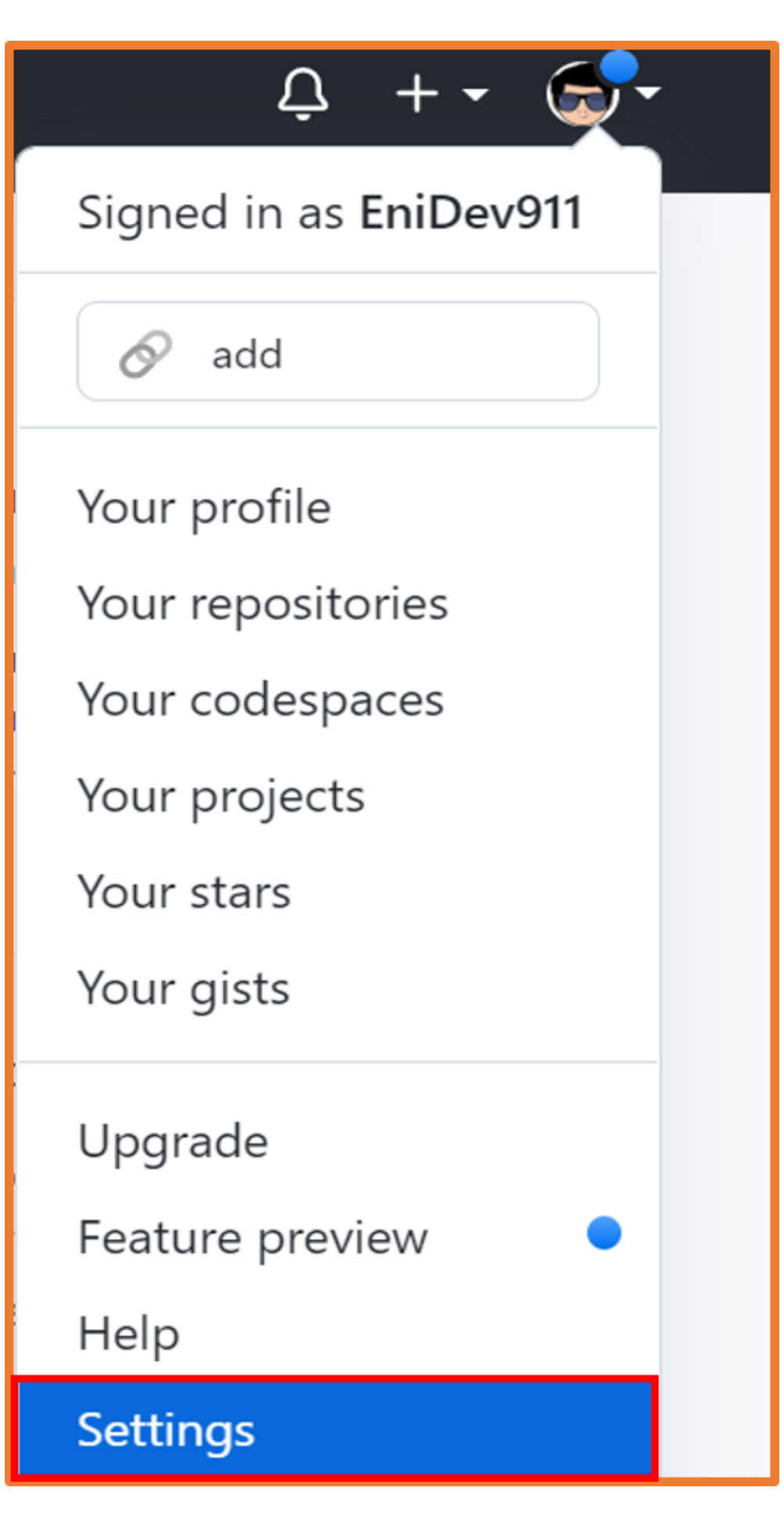
- ~: es un símbolo llamado virgulilla que en los sistemas operativos UNIX se refiere al valor de la variable $HOME, esto es, el directorio del usuario que está logueado.
- **gh_rsa.pub**: Este archivo es el que almacena el contenido de la llave pública, por ende, debemos tener mucho cuidado en verificar que estemos copiando el contenido de este archivo y no de otro si en tu caso le dejaste el nombre predeterminado deberias usar `clip < ~/.ssh/id_rsa.pub`.

Ejemplo:

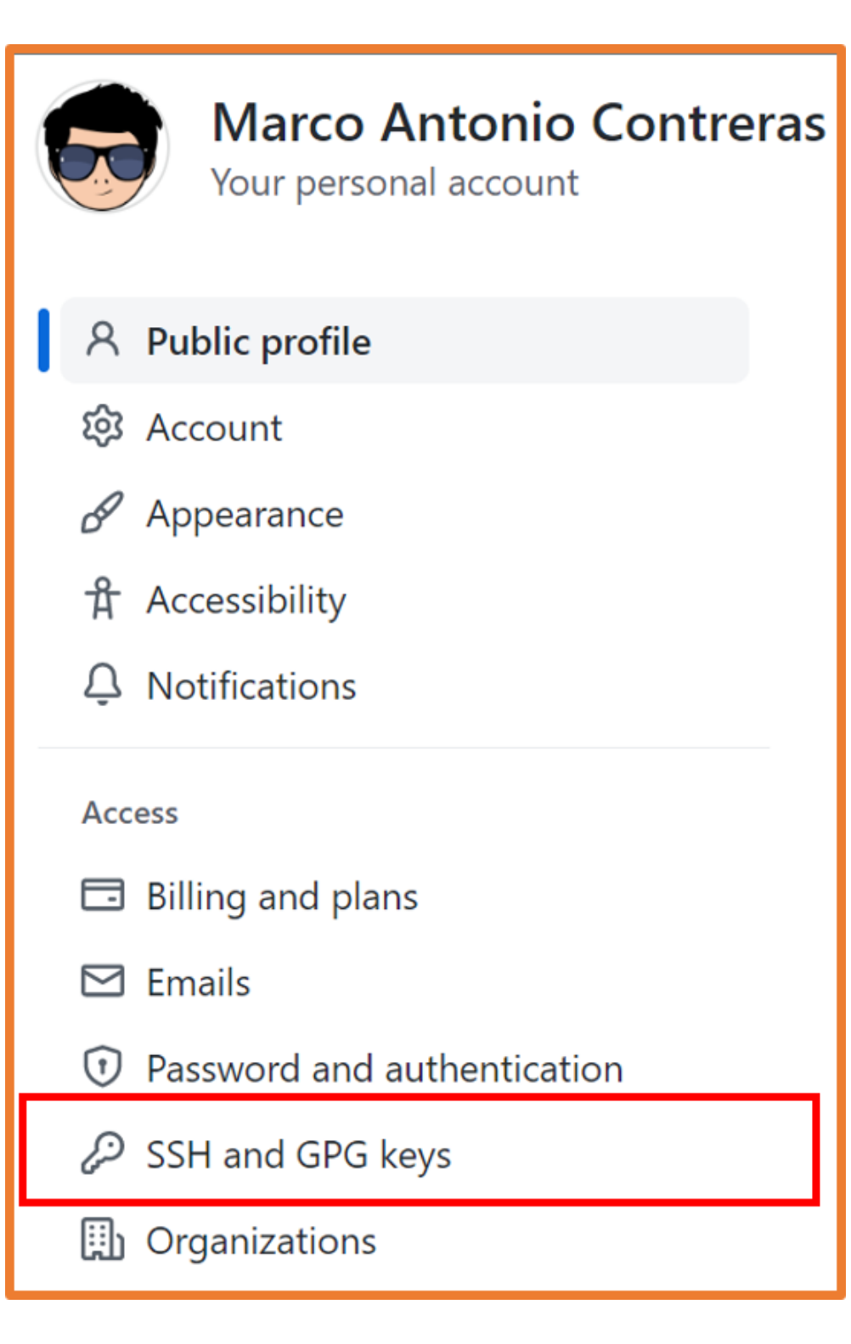


```
Marco@DESKTOP-3L88H0U MINGW64 ~
$ clip < ~/.ssh/gh_rsa.pub
```

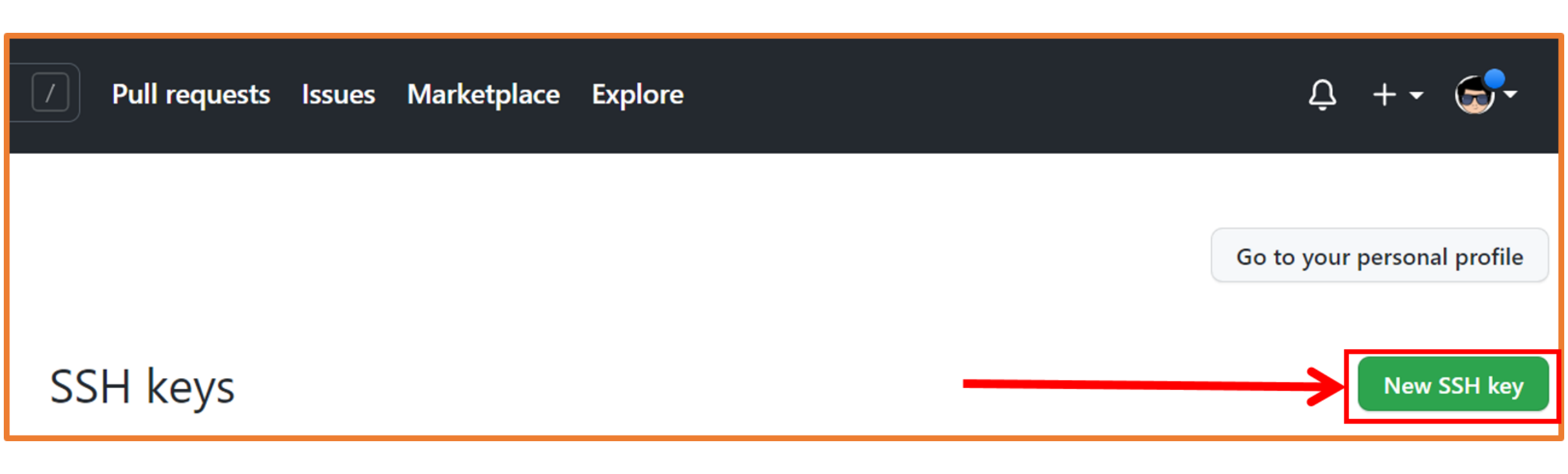
2. Ahora que tenemos copiado en el portapapeles el contenido de la llave publica, nos vamos a nuestra cuenta de GitHub e iniciamos sesión. En la esquina superior derecha de nuestra página en GitHub, das clic en tu foto de perfil y despues da clic en configuración.



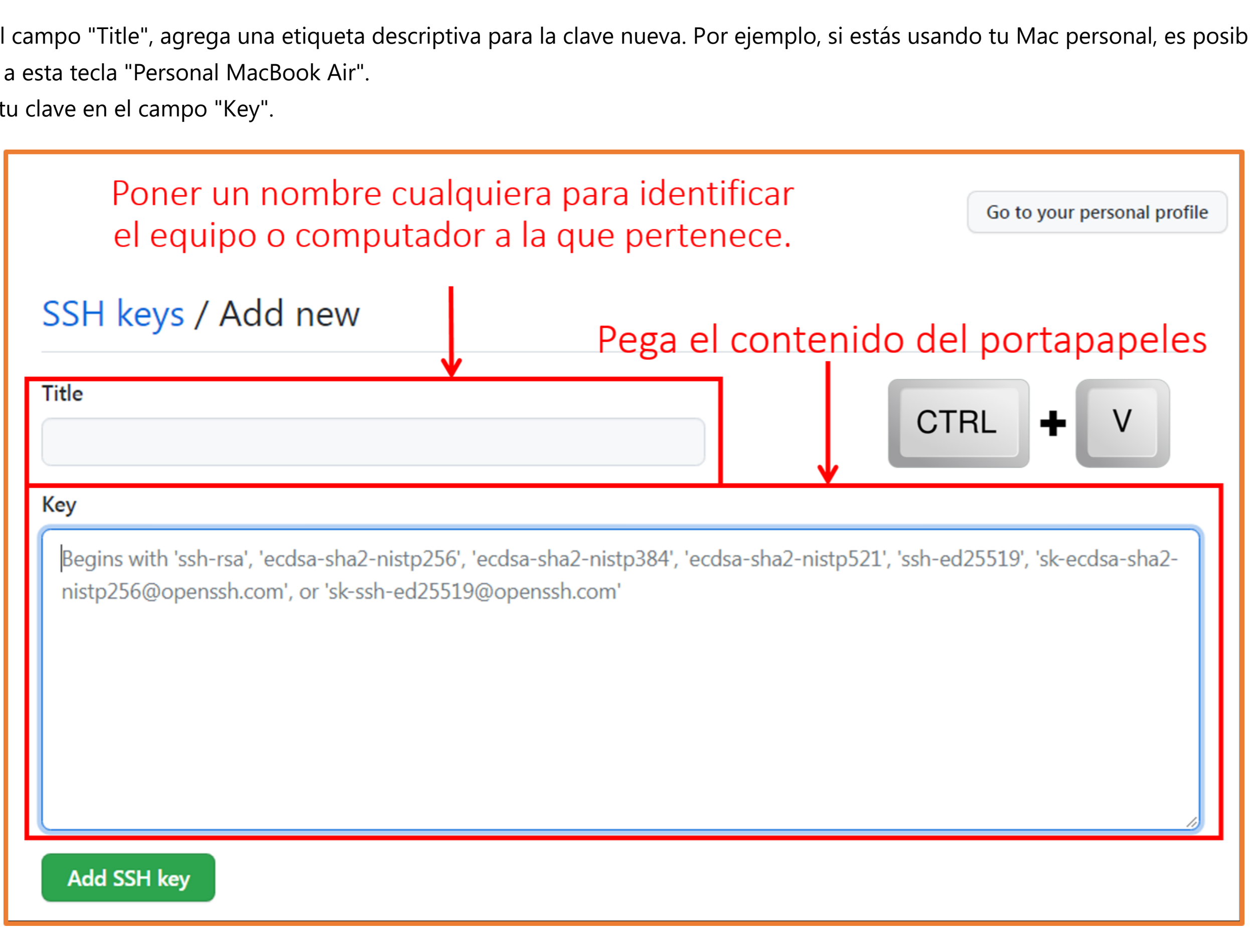
3. En la barra lateral de configuración de usuario, da clic en llaves SSH y GPG.

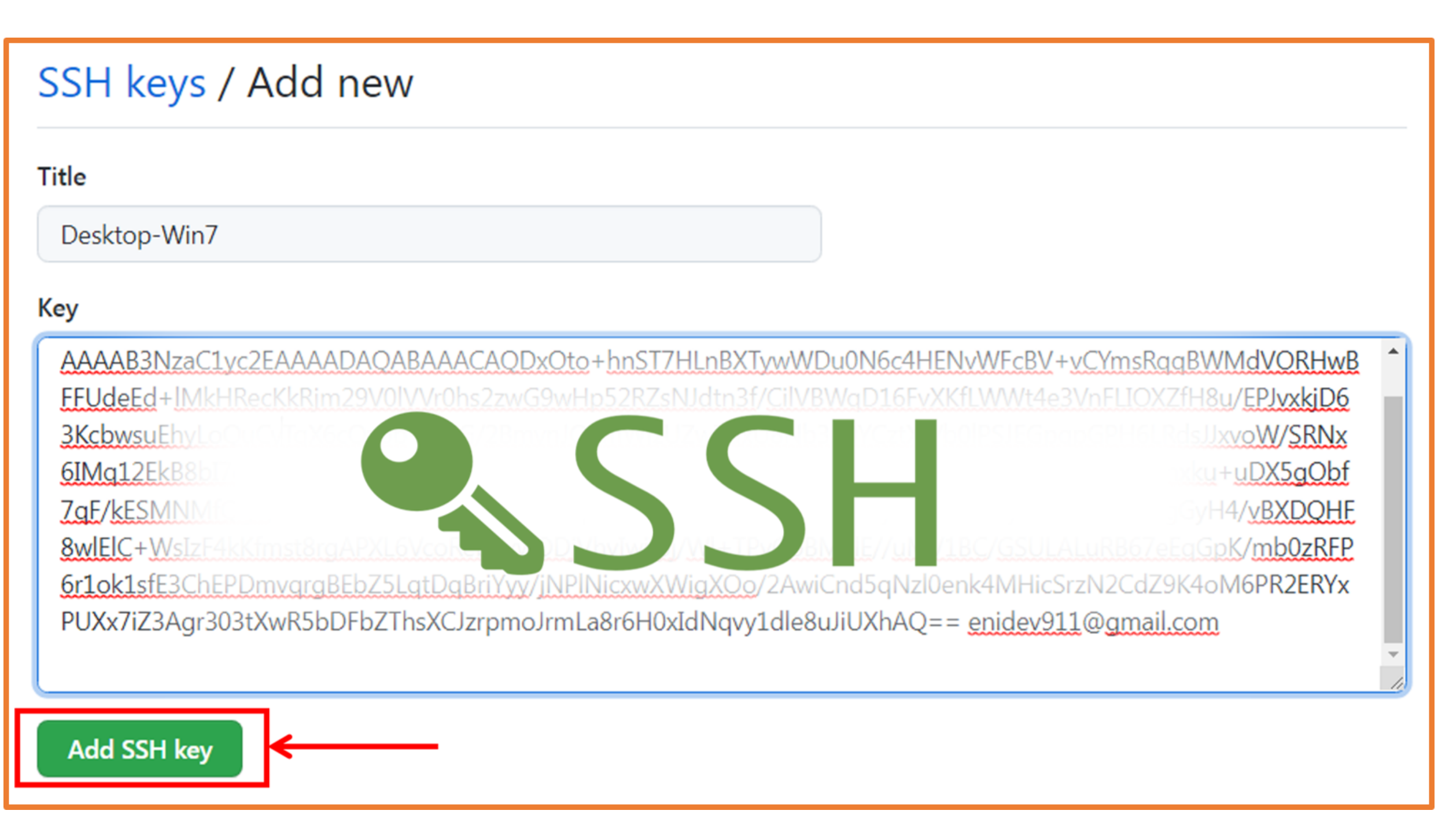


4. Haz clic en **New SSH key** o **Add SSH key**.



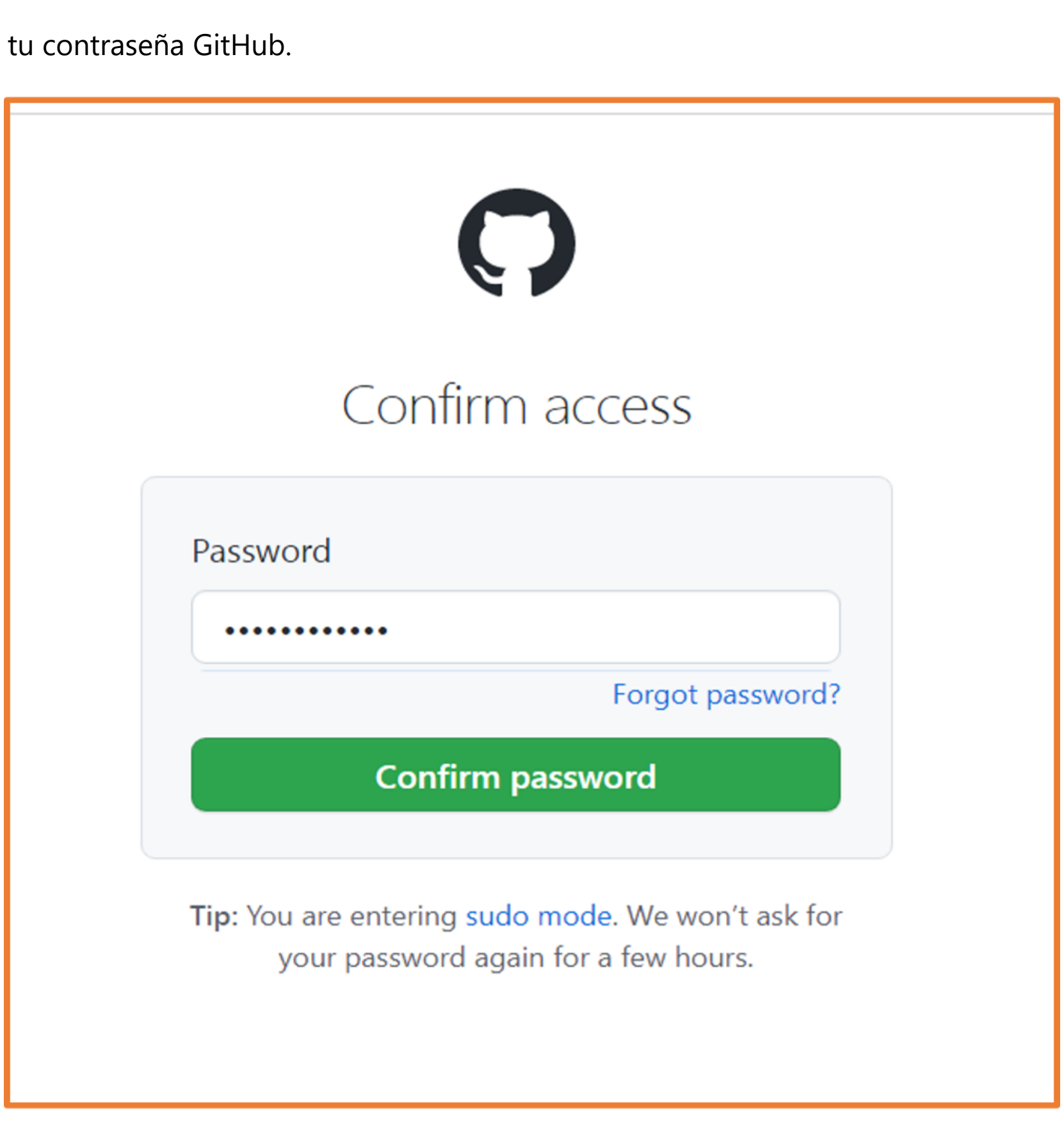
5. En el campo "Title", agrega una etiqueta descriptiva para la clave nueva. Por ejemplo, si estás usando tu Mac personal, es posible que llames a esta tecla "Personal MacBook Air".
Copia tu clave en el campo "Key".



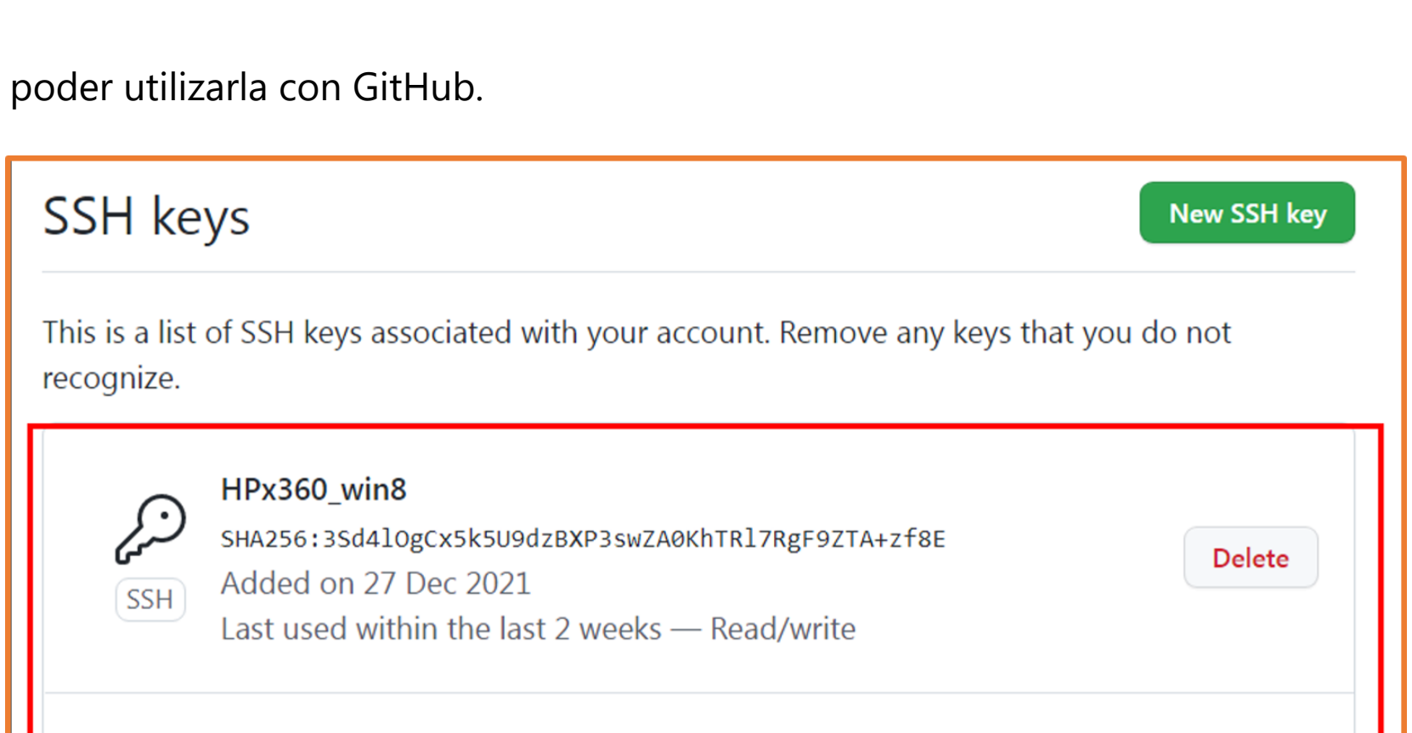


Title: Desktop-Win7

6. Si se te solicita, confirma tu contraseña GitHub.



Está lista nuestra llave para poder utilizarla con GitHub.



This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

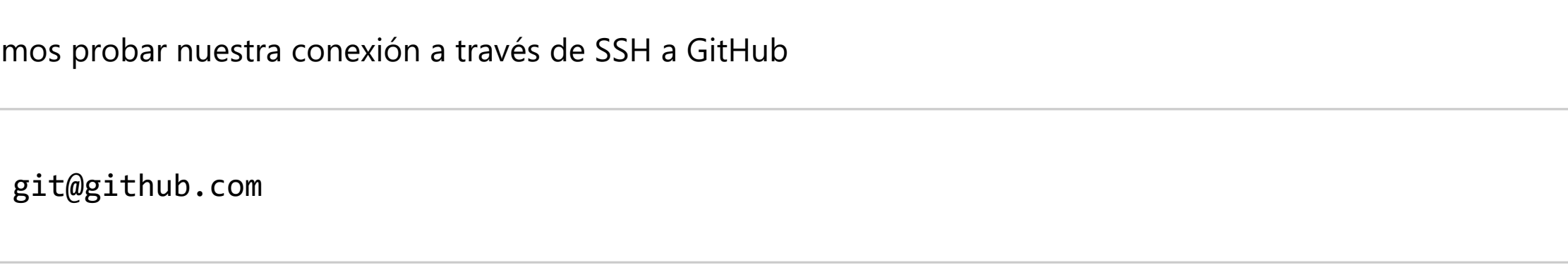
HPx360_win8
SHA256:3Sd4lOgCx5k5U9dzBXP3swZA0KhTRl7RgF9ZTA+zf8E
Added on 27 Dec 2021
Last used within the last 2 weeks — Read/write

Desktop-Win7
SHA256:EUqg+9hLV9VHJY7fVWOknQAp+VE4WRVvs+jZTngrNVg
Added on 15 Jan 2022
Never used — Read/write

Probar la conexión SSH a GitHub

Ahora podemos probar nuestra conexión a través de SSH a GitHub

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



```
Marco@DESKTOP-3L88H0U MINGW64 ~
$ ssh -T git@github.com
Hi EniDev911! You've successfully authenticated, but GitHub does not provide shell access.
```

Si la última línea contiene su nombre de usuario en GitHub, ¡está autenticado correctamente!.

Si tienes alguna sugerencia escribeme: