Configurar llaves SSH en Git y GitHub

**Criptografía asimétrica**

**¿Para qué necesitamos la criptografía asimétrica?**

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

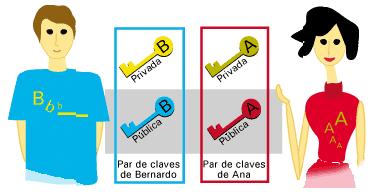


**¿Qué es la criptografía asimétrica?**

La criptografía asimétrica es la forma segura de enviar y recibir un mensaje, ya que incluso si llega a caer en las manos de otra persona, nunca podrá leer el mensaje.

**¿Cómo funciona?**

En este caso, Bernardo 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 Bernardo su llave pública y ella conserva la llave privada, Bernardo manda su mensaje cifrado con la llave pública que le proporcionó Ana, Ana recibe el mensaje de Bernardo y lo desencripta con su llave privada de esta manera Ana y sólo Ana podrá leer el mensaje que le envió Bernardo.



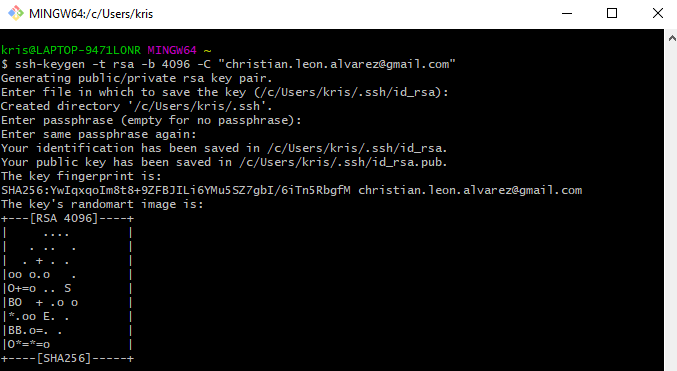
No estamos exentos a estos ataques a la hora de hacer un **push** o un **pull** a GitHub.  
O de igual forma pueden **crackear** el usuario y contraseña y así tener acceso al código de nuestros proyectos, por este motivo debemos tener una conexión segura entre las dos partes.

**Windows/Linux**

En nuestra consola de git bash(Windows) o nuestra consola (Linux) nos movemos a home, porque las llaves pueden ser para todos los proyectos en los que trabajemos.  
Luego generamos las llaves con ***ssh-keygen*** seguidos de los parámetros: ***-t*** aquí ponemos el algoritmo de encriptación que queremos en este caso ocuparemos la más común que es RSA,  ***-b*** aquí vamos a especificar qué tan compleja es la llave, seguido de ***-C*** aquí vamos a poner el email al que se va a configurar la llave.

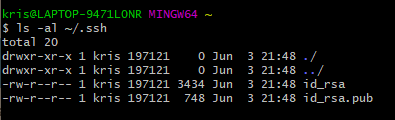
***ssh-keygen -t rsa -b 4096 -C “tu\_email@gmail.com”***

Nos pide la ruta en la que se va a guardar las credenciales, en este caso damos enter, para que las guarde en home: https://krisleon99.github.io/portfolio/img/bg/1.png

Después nos pide una contraseña con espacios (es recomendable ponerla)  
https://krisleon99.github.io/portfolio/img/bg/2.png  
Volvemos a ingresar la contraseña:https://krisleon99.github.io/portfolio/img/bg/3.png  
Nos genera la llave **privada**: **id\_rsa** en una **carpeta oculta** **.ssh** dentro del directorio home y la **llave pública** **id\_rsa.pub**  
https://krisleon99.github.io/portfolio/img/bg/4.png  
Una vez obtenido el resultado sabremos que hemos creado nuestras llaves (pública y privada) SSH satisfactoriamente:  


Para comprobarlo escribimos el comando:

**ls -al ~/.ssh**



Debemos evaluar si está corriendo el servidor de ssh:

**eval $(ssh-agent -s)**

https://krisleon99.github.io/portfolio/img/bg/7.png

Agregar la llave privada a Git (Nos pide el password que le dimos con espacios).

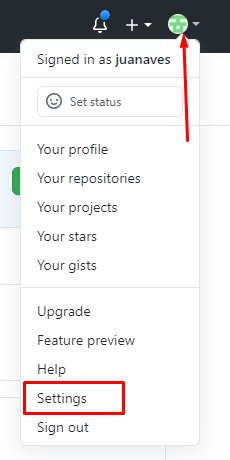
**ssh-add ~/.ssh/id\_rsa**

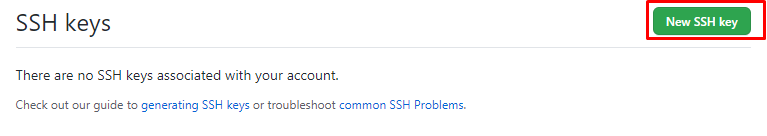
https://krisleon99.github.io/portfolio/img/bg/8.png

Nos aseguramos de copiar el contenido de nuestra llave SSH pública (**id\_rsa.pub**)  
lo podemos hacer con el uso de vi o nano:

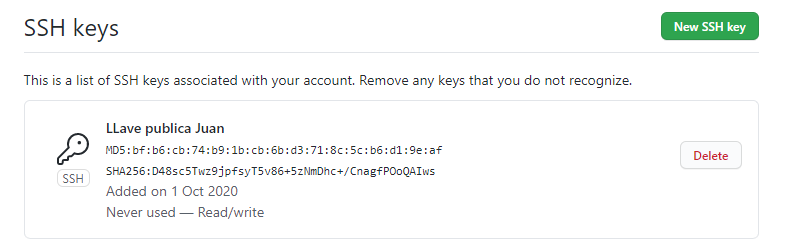
**vi ~/.ssh/id\_rsa.pub**

Copiamos el contenido. (Salimos con esc+shift+zz)  
En estos momentos debemos de ir a nuestro GitHub → settings → SSH and GPG Keys → New SSH Key

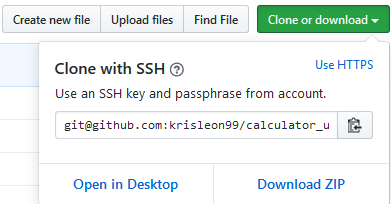




En title ponemos algo que nos identifique, en key pegamos lo que copiamos de nuestra llave pública:

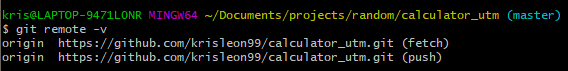
Damos click a **Add SSH Key**, podemos ver que nos agregó una nueva llave con el nombre que le dimos, pero aún no se ha usado.  


Copiamos la llave pública de cualquiera de nuestros repositorios de GitHub (Clone with SSH)



Tenemos que posicionarnos en el proyecto que copiamos la llave SSH pública de GitHub y con el siguiente comando podemos ver a qué repositorio apunta nuestro proyecto:

**git remote –v**



Seteamos la ruta

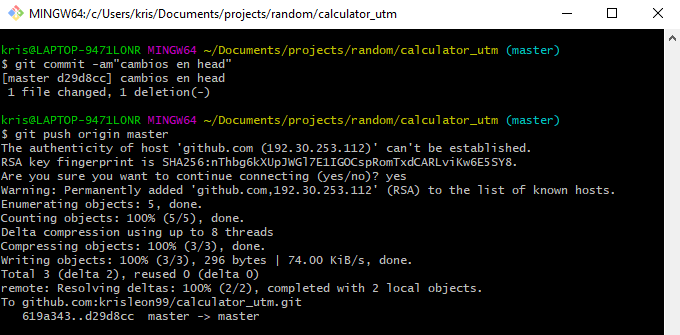
git remote set-url origin [git@github.com:krisleon99/calculator\_utm.git](mailto:git@github.com:krisleon99/calculator_utm.git)

https://krisleon99.github.io/portfolio/img/bg/14.png

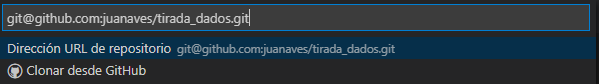
En este caso no nos dice nada la consola, pero podemos verificarlo ingresando de nuevo el comando (podemos notar que cambió el origin):

**git remote –v**

https://krisleon99.github.io/portfolio/img/bg/15.png

Y listo, podemos comprobarlo haciendo un push al repositorio y no nos pedirá el usuario ni la contraseña (La primera vez nos pregunta que si estamos seguros de continuar la coneccion con una llave RSA, le damos yes).  


Tenemos que instalar la extensión



## Agregar su clave SSH al ssh-agent (opcional)

Puede agregar su clave SSH al ssh-agent, si no desea volver a ingresar su frase de contraseña cada vez que use su clave SSH. Para hacerlo:

Inicie el ssh-agent en segundo plano, usando el comando

eval $(ssh-agent -s)



Borra todo lo que existe

ssh-add -D



Luego agregue la clave generada al agente

ssh-add ~/.ssh/id\_rsa



Para enumerar las claves agregadas:

ssh-add -l

