

**Guía: Preparación de ambiente Git Bash, conectando con la cuenta de GitHub.**

1. **Contenido**

[II. Introducción. 3](#_heading=h.gjdgxs)

[A. Arquitectura de conexión aplicativos. 3](#_heading=h.30j0zll)

[B. Arquitectura de funcionamiento Desarrollador utilizando Git Bash 3](#_heading=h.1fob9te)

[C. Arquitectura de funcionamiento Desarrollador utilizando GitHub Desktop 4](#_heading=h.3znysh7)

[III. Configuración Git Bash con GitHub. 4](#_heading=h.2et92p0)

[A. II.1 Descarga e instalación de Git 4](#_heading=h.tyjcwt)

[B. II.2 Creación de llave SSH 7](#_heading=h.3dy6vkm)

[C. II.3 Creación de proyecto a subir a Github 11](#_heading=h.1t3h5sf)

[D. II.4 Creación repositorio en GitHub 16](#_heading=h.4d34og8)

[E. II.5 Envío de repositorio local a repositorio remoto 17](#_heading=h.2s8eyo1)

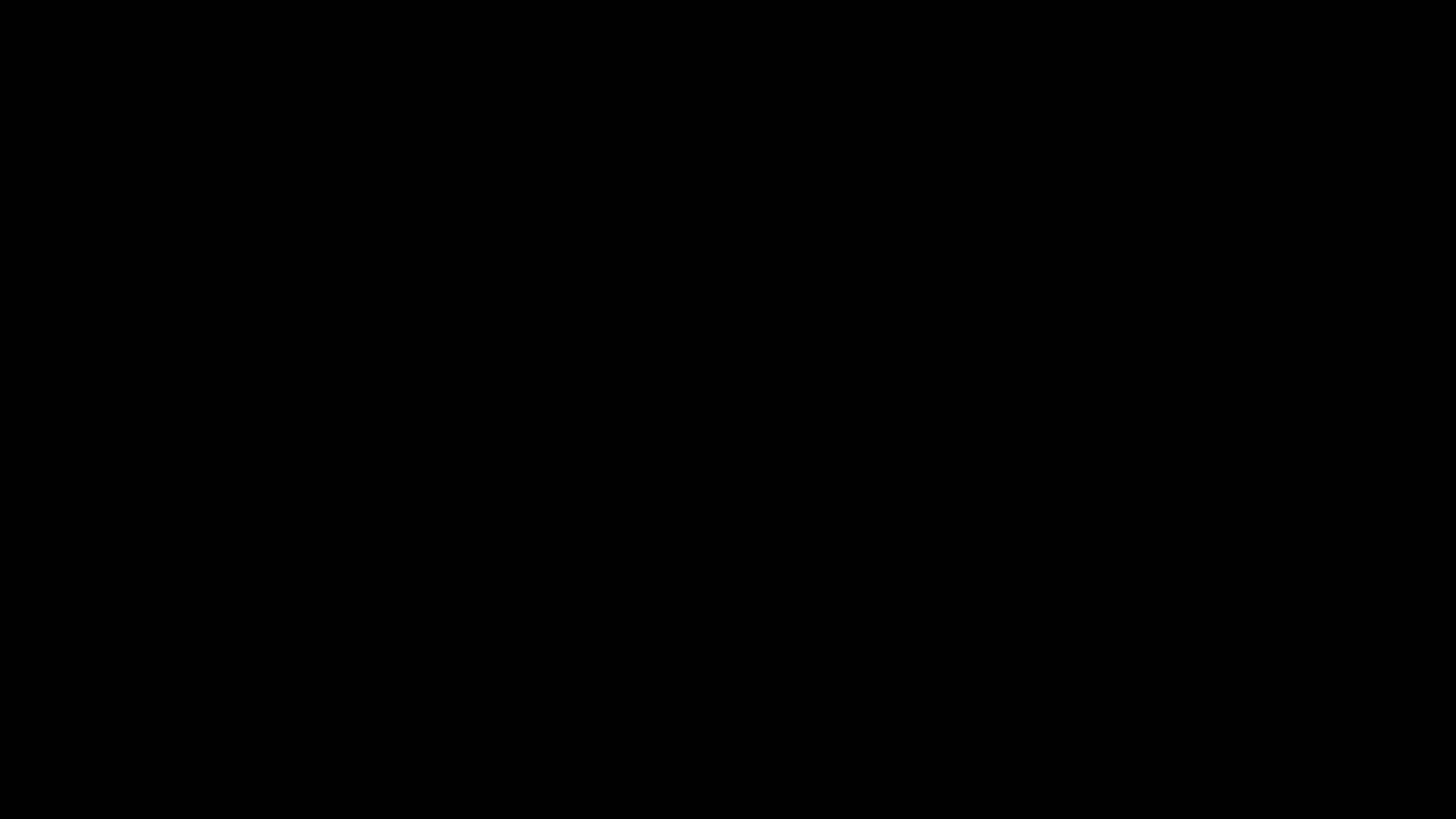
1. **Introducción.**

En la presente guía, revisaremos los procesos para conectar las herramientas Git Bash, y GitHub Desktop, con la nube de GitHub.

* 1. **Arquitectura de conexión aplicativos.**



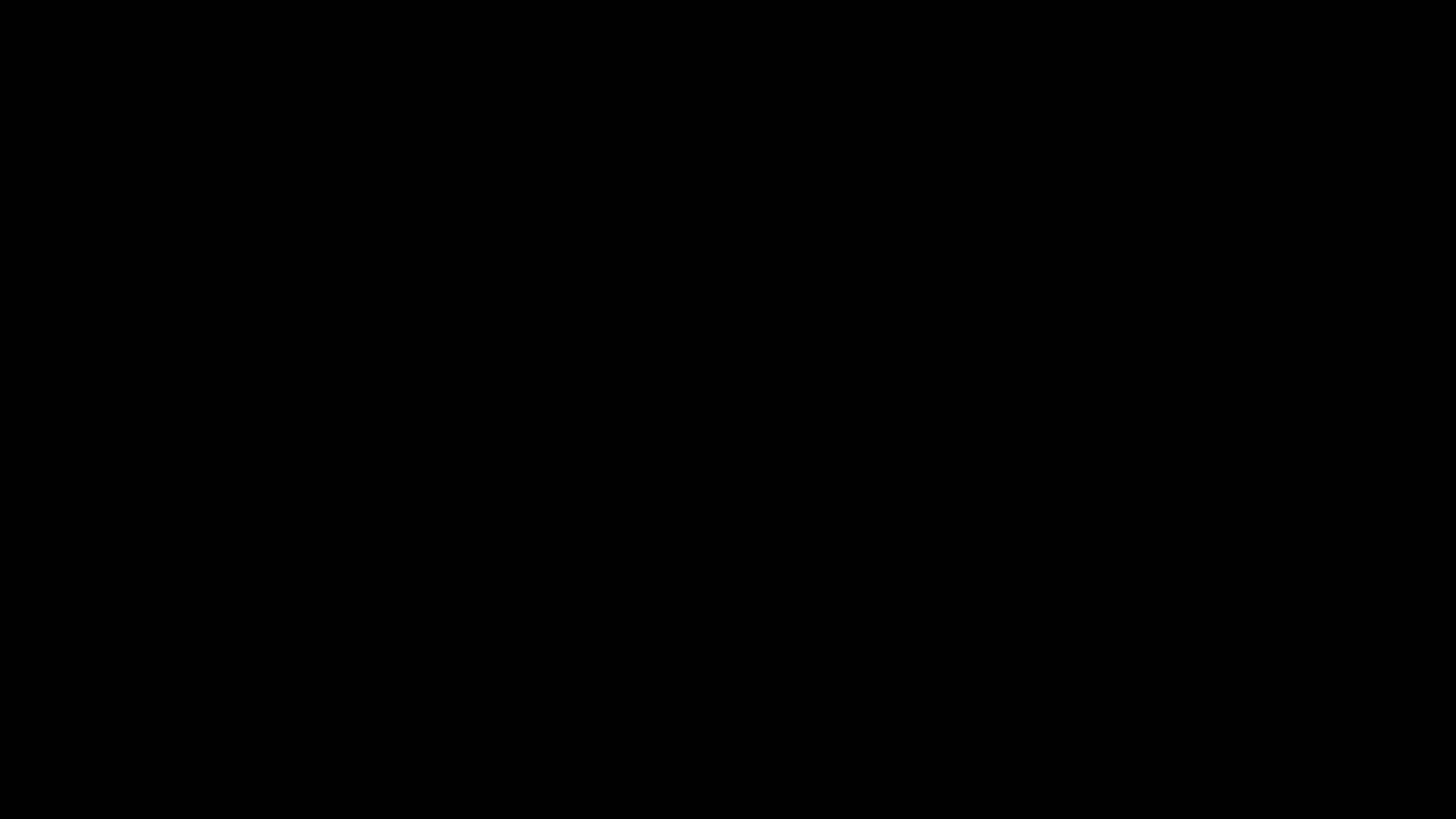




* 1. **Arquitectura de funcionamiento Desarrollador utilizando Git Bash**

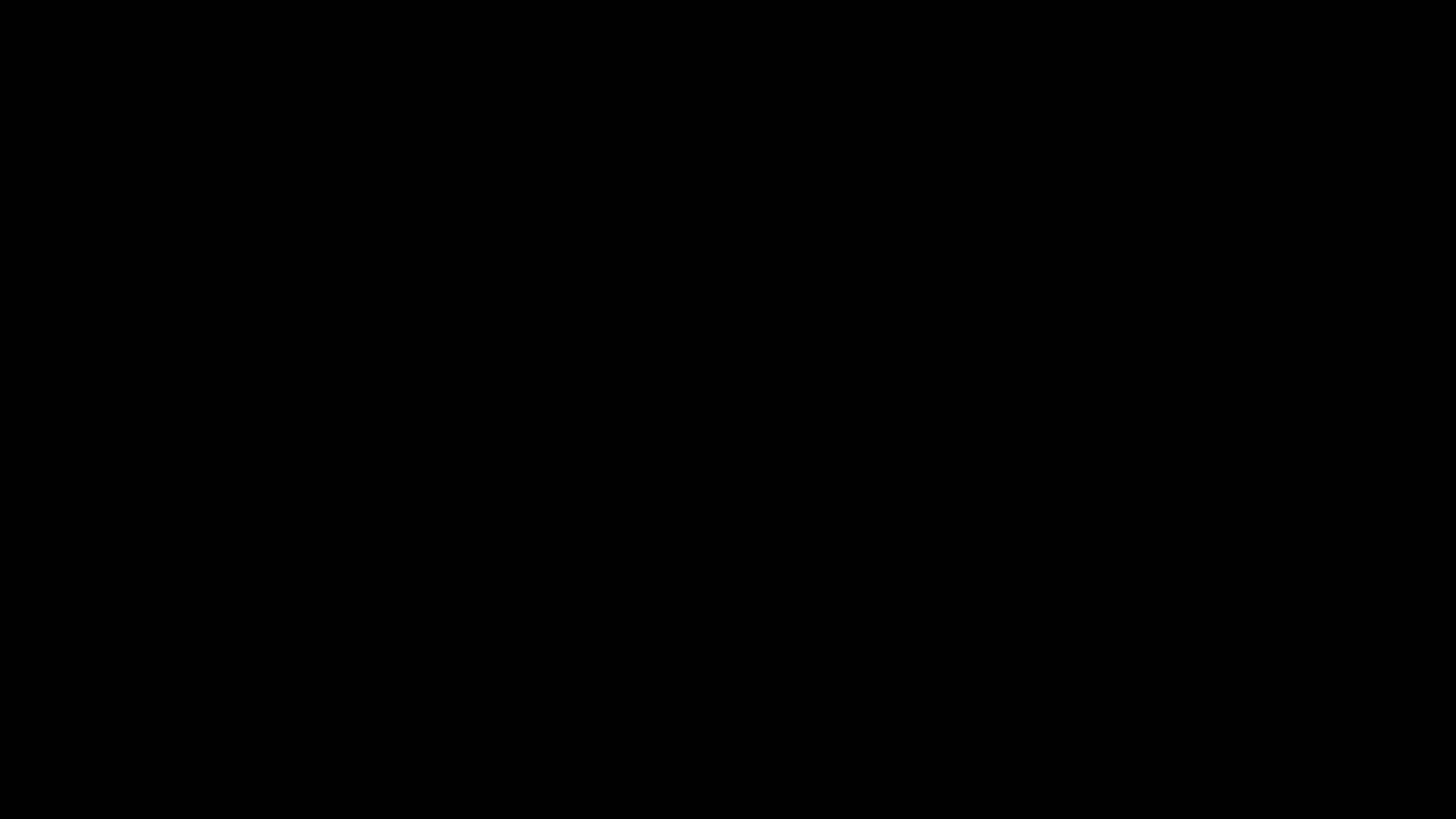






* 1. **Arquitectura de funcionamiento Desarrollador utilizando GitHub Desktop**





**Aclaración**: el desarrollador puede trabajar con Git Bash y GitHub sin utilizar GitHub Desktop, o viceversa, esto es debido a que Git Bash y GitHub Desktop realizan el mismo proceso, la gran diferencia entre ambas, es que la primera realiza los procesos por medio de una interfaz, y la otra por una línea de comandos

1. **Configuración Git Bash con GitHub.**
   1. **II.1 Descarga e instalación de Git**
2. Descargar el producto desde plataforma AVA, llamado 3.1.3 Git Bash.7z, o desde internet, ingresando al siguiente link <https://git-scm.com/download/win>
3. Descomprimir e iniciar el proceso de instalación.



1. Dar click en “**next**”

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. Dar click en “**next**” nuevamente y dejar preseleccionado todas la opciones que Git nos marca, estas opciones preseleccionadas, son las más adecuadas para un entorno de aprendizaje y primeros pasos.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

1. Presionar “**Finish**”

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. Debería abrirse el aplicativo Git Bash

Interfaz de usuario gráfica, Texto

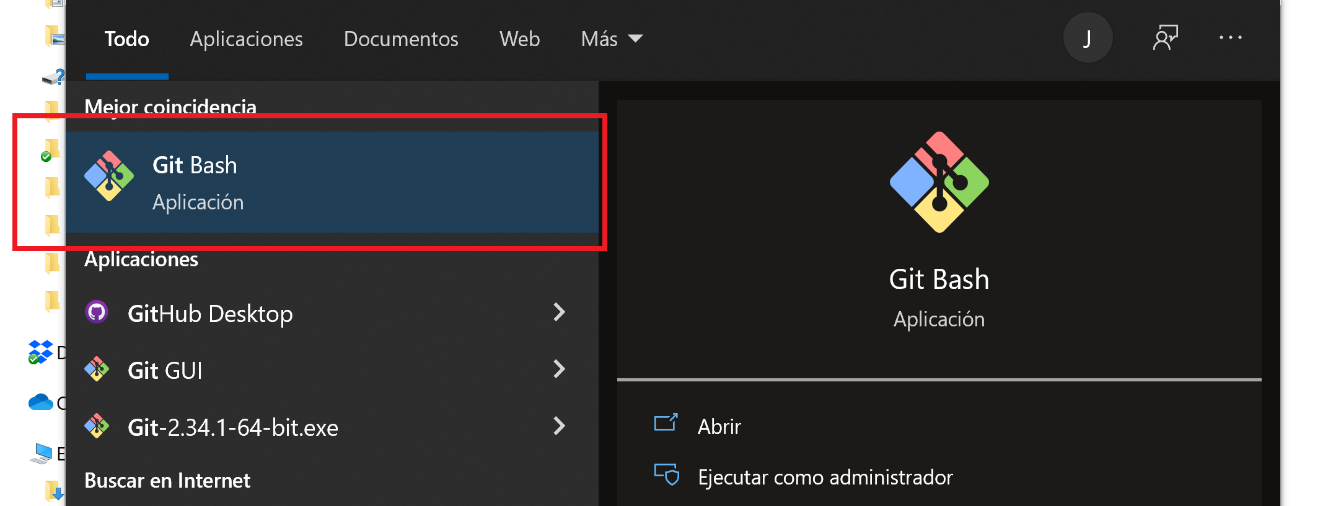
Descripción generada automáticamente

* 1. **II.2 Creación de llave SSH**

Para conectar Git con GitHub, es necesario crear y configurar una llave SSH, con el fin de que nuestro computador comience a autenticar la información que se envía al repositorio de GitHub cada vez que este recibe información nueva.

Fuente: <https://docs.github.com/es/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent>

1. Abrir Git desde Windows



1. Ejecutar el siguiente comando e ingresar su Correo de DUOC UC

$ ssh-keygen -t ed25519 -C "tu\_correo\_de\_duoc"

Texto

Descripción generada automáticamente

1. Solicitará ingresar un recordatorio, se recomienda que guarde esa frase, debido a que actuará como una password. Esta frase luego debe repetirse.

**NOTA**. Cuando ingresamos una frase o palabra clave, escribiremos en el teclado, pero no se verá en la consola, es debido a que es considerada como una palabra secreta

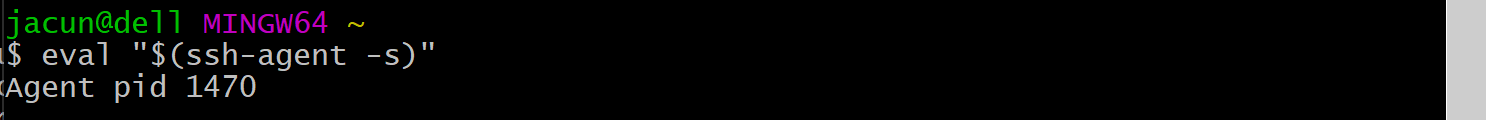
Texto

Descripción generada automáticamente

1. Ingresar el siguiente comando para levantar servicio

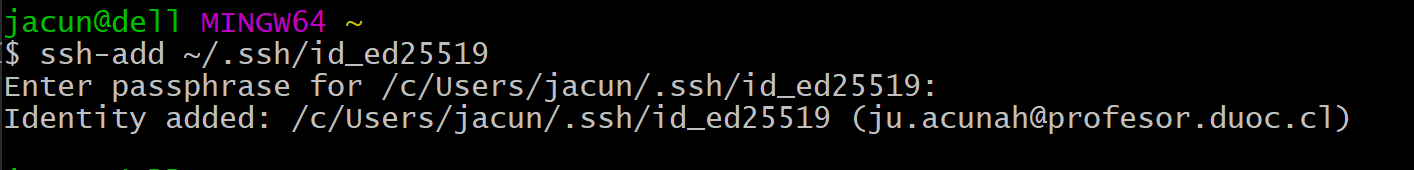
$ eval "$(ssh-agent -s)"

**NOTA**: el PID puede ser distinto para cada computador, lo cual no afecta en la conexión.



1. Ingrese el siguiente comando para incorporar la frase en el archivo id\_ed25519,

$ ssh-add ~/.ssh/id\_ed25519

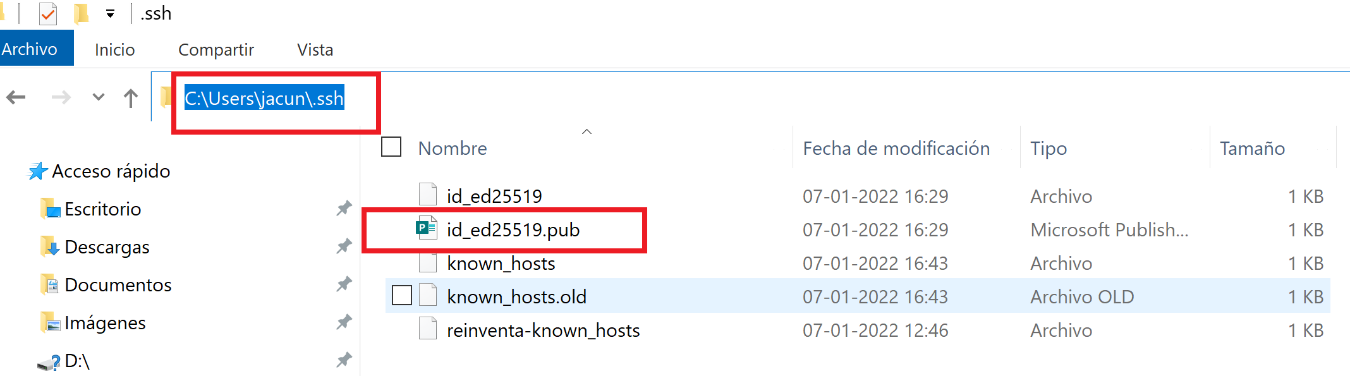


Recuerde ingresar la misma frase o palabra clave colocada en el paso 9

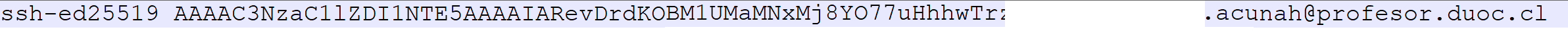
La forma de saber que todo está correcto es que la última línea indique que la identidad (su correo electrónico) ha sido incorporado en el archivo de seguridad.

1. Nos dirigiremos ahora al explorador de archivos, e iremos a la ruta C:\Users\Nombre\_De\_Su\_Usuario\.ssh

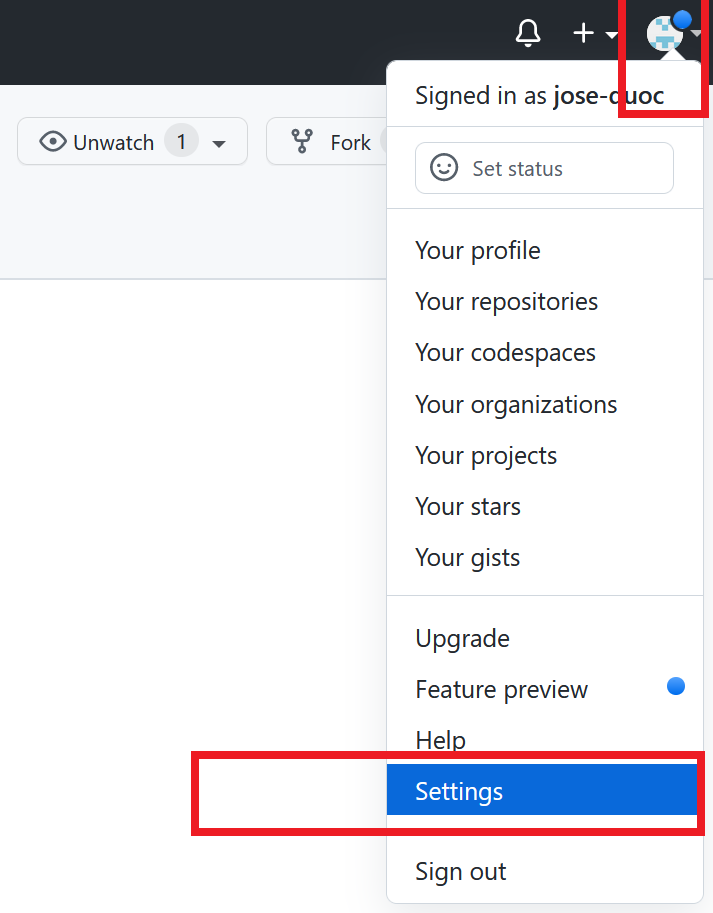
Abriremos el archivo id\_ed25519.pub con un editor de texto (WordPad, Notepad++)



1. Copiaremos la cadena completa hasta el correo electrónico el contenido y nos dirigiremos a GitHub.com



1. Ingresaremos a la opción “**settings”** dentro del perfil de usuario



1. Nos dirigimos a la sección “**SSH and GPG Keys**” y luego presionado el botón “**New SSH Key**”

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. Incorporaremos la llave copiada desde el archivo y la nombramos “**llave**”, luego guardaremos la clave presionando “**Add SSH key**”, con esto ya tendremos lista la conexión de nuestro equipo hacia GitHub utilizando Git.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

* 1. **II.3 Creación de proyecto a subir a Github**

1. Ahora, crearemos un proyecto sencillo en Visual Studio trabajando con Python, lo llamaremos “**Mi proyecto python**”

def hola\_mundo():

print ('hola mundo')

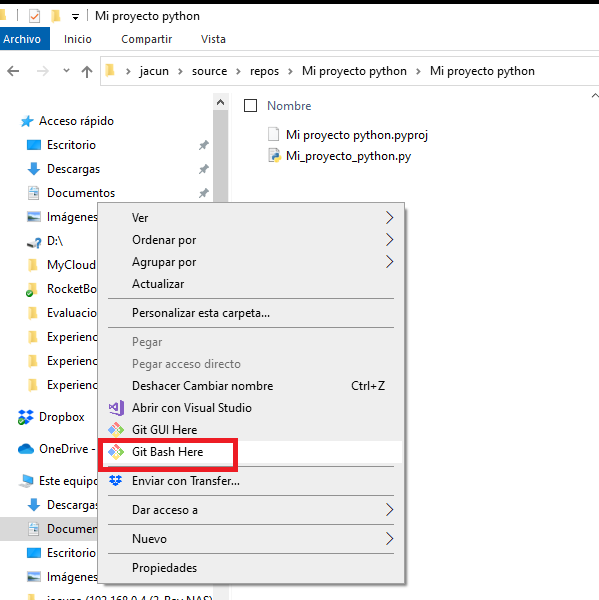
hola\_mundo()

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

1. Nos dirigiremos hacia la ubicación del proyectoInterfaz de usuario gráfica, Texto, Aplicación

   Descripción generada automáticamente
2. Daremos click derecho, y seleccionamos la opción “**Git Bash Here**”



1. Se abrirá la siguiente ventana

Texto

Descripción generada automáticamente

1. Configuraremos el nombre y correo

**Nota**: debe ser los mismos username y correo de tu cuenta de GitHub

$ git config --global user.name "Nombre de usuario"

$ git config --global user.email "correo"

Texto

Descripción generada automáticamente

1. Ahora traspasaremos nuestro proyecto que se encuentra en el “**área de trabajo**” de Git, hacia el “área de pruebas” (**staging area**)

Diagrama, Esquemático

Descripción generada automáticamente



1. Ejecutamos el siguiente comando

$ git init

Texto

Descripción generada automáticamente

Git crea una carpeta oculta, la podemos observar colocando el siguiente comando

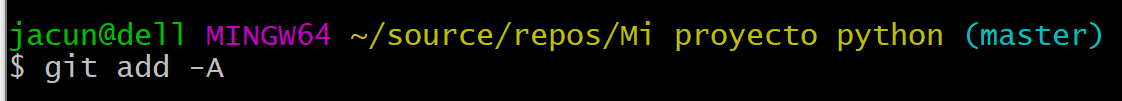
$ ls -alh

Imagen que contiene Texto

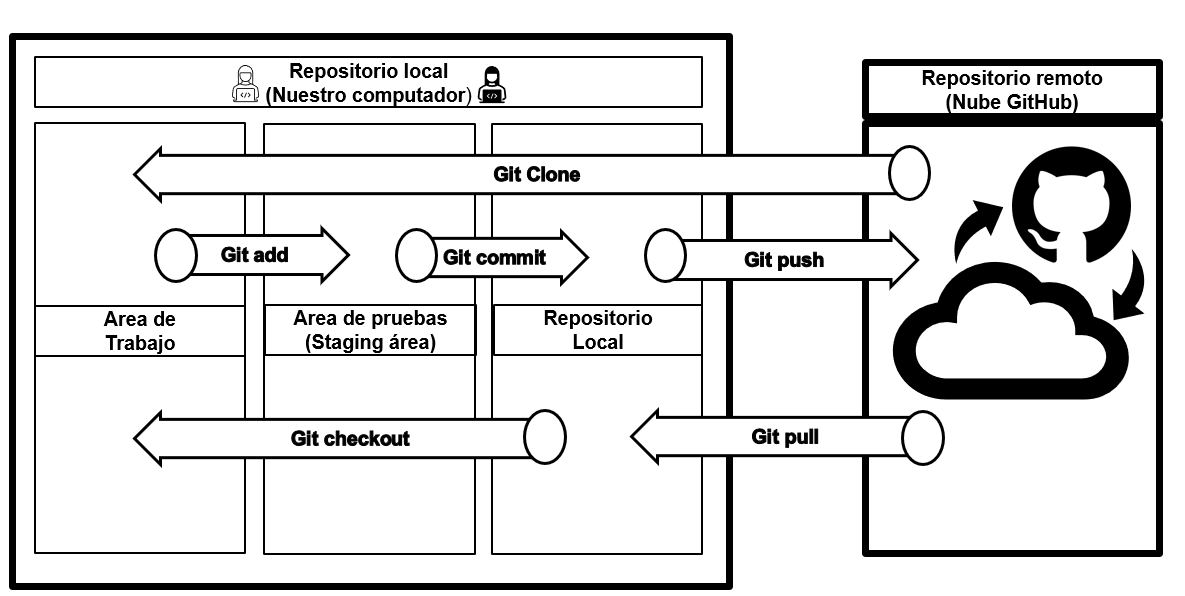
Descripción generada automáticamente

1. Realizaremos la copia del proyecto hacia el staging area

$ git add -A



1. Ahora traspasamos hacia el repositorio local



1. Escribiremos el siguiente comando

$ git commit -m "mi primer commit"

Texto

Descripción generada automáticamente

* 1. **II.4 Creación repositorio en GitHub**

1. Conectaremos nuestro repositorio local con GitHub.

Interfaz de usuario gráfica, Aplicación, Sitio web

Descripción generada automáticamente

1. Crearemos un repositorio para ubicar nuestro proyecto.

Interfaz de usuario gráfica, Texto, Correo electrónico

Descripción generada automáticamente

1. Una vez creado, utilizaremos los comandos propuestos por GitHub para realizar la carga, a lo que debe seleccionar la opción “SSH”

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

* 1. **II.5 Envío de repositorio local a repositorio remoto**

1. Escribiremos el siguiente comando

$ git remote add origin git@github.com:su\_usuario/My-proyecto-python.gi

Texto

Descripción generada automáticamente

1. Ubicamos nuestro proyecto en la rama principal “**Main**”

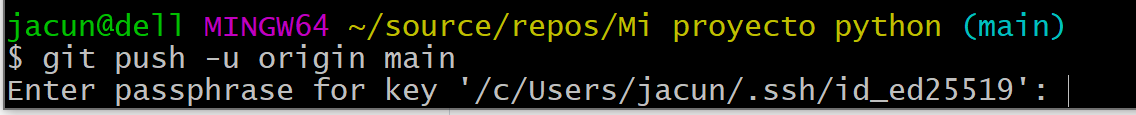
$ git branch -M main

Texto

Descripción generada automáticamente con confianza media

1. Ingresaremos el siguiente comando para cargar el repositorio local en el repositorio remoto.

git push -u origin main



Una vez ingresado, Git Bash solicitará la frase o clave incorporada en el archivo (la misma frase que se colocó en el paso 9 de esta guía)

1. Luego de colocar la frase o clave, Git Bash confirmada los archivos subidos, también los podremos corroborar en GitHub

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente