

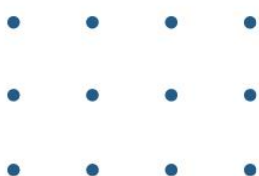


¡APRENDIENDO!

Programación orientada a objetos



Ing. Luis Enrique Jaramillo Montaña



2023

1. UNIDAD 1: FUNDAMENTOS BÁSICOS.

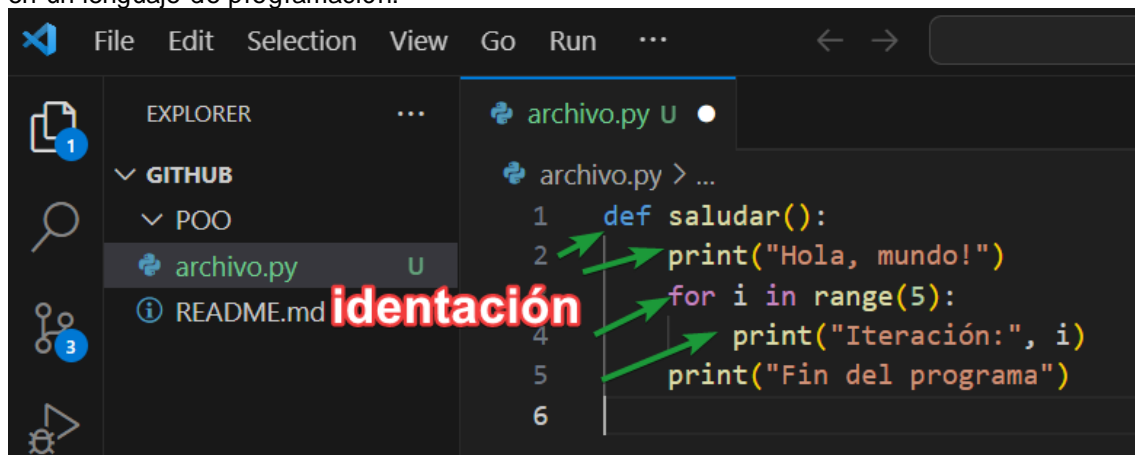
1.1. Entorno de desarrollo:

Un entorno de desarrollo, comúnmente conocido como IDE por sus siglas en inglés (Integrated Development Environment), es una herramienta de software integral que se utiliza para facilitar la programación y el desarrollo de software. Un IDE proporciona un conjunto de características y herramientas que ayudan a los programadores a escribir, depurar, compilar y administrar código de manera más eficiente.

Estos entornos están diseñados para simplificar el flujo de trabajo de desarrollo y mejorar la productividad del programador. Algunos componentes comunes de un entorno de desarrollo incluyen:

Editor de código: Ofrece funciones de edición de texto avanzadas, como resaltado de sintaxis, autoindentación y autocompletado, lo que facilita la escritura de código.

La Identación se refiere al formato visual utilizado para estructurar y organizar el código fuente en un lenguaje de programación.



Depurador: Permite a los programadores identificar y corregir errores en su código al proporcionar herramientas para establecer puntos de interrupción, inspeccionar variables y seguir la ejecución del programa paso a paso.

Compilador/Intérprete: Proporciona la capacidad de compilar y ejecutar el código directamente desde el IDE. Algunos IDE también son específicos para un lenguaje y pueden incluir un intérprete para ejecutar el código directamente.

Gestión de proyectos: Permite organizar y administrar proyectos de desarrollo, lo que incluye la creación, apertura y gestión de archivos y directorios.

Herramientas de versionado: Puede integrar sistemas de control de versiones, como Git, para realizar un seguimiento de los cambios en el código y colaborar con otros desarrolladores.

Asistencia en el desarrollo: Ofrece sugerencias y advertencias en tiempo real mientras se escribe código, lo que ayuda a detectar posibles errores y mejorar la calidad del código.

Herramientas de construcción: Facilita la construcción y automatización de tareas, como la compilación, la generación de documentación y la ejecución de pruebas.

Herramientas de depuración visual: Algunos IDE proporcionan una interfaz gráfica para depurar aplicaciones de manera más visual, lo que puede incluir representaciones gráficas de estructuras de datos y flujos de programa.


Integración con SDK y bibliotecas: Puede integrar SDK (kits de desarrollo de software) y bibliotecas relevantes para el lenguaje o plataforma de desarrollo específico.

Por ejemplo, un SDK de Android proporciona las herramientas y recursos necesarios para desarrollar aplicaciones para dispositivos Android, incluyendo bibliotecas para interactuar con el hardware y el sistema operativo, herramientas de depuración, un emulador de Android para probar aplicaciones y documentación detallada.

Los IDE varían según el lenguaje de programación y la plataforma de desarrollo. Ejemplos populares de IDE incluyen Visual Studio (para desarrollo en C# y .NET), Eclipse (para Java), IntelliJ IDEA (para Java y otros lenguajes), PyCharm (para Python) y muchos otros. La elección de un IDE suele depender de las preferencias del desarrollador y del entorno de desarrollo específico en el que trabaje.

1.2. características e instalación.

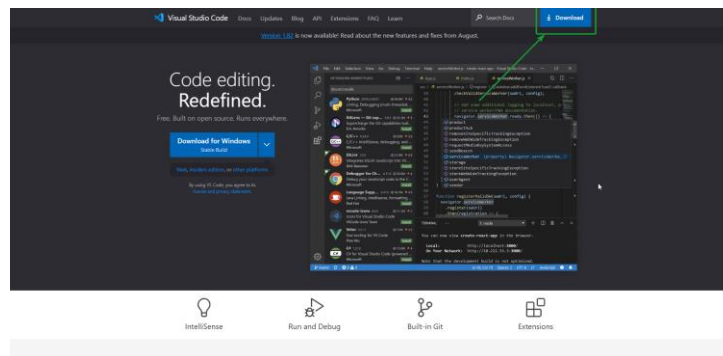
Para este curso se utilizará los lenguajes de programación de Python y Java, por lo que instalaremos dos entornos virtuales, para sacar el mayor provecho en el caso del lenguaje de programación Java.

El primer entorno virtual será Visual Studio Code.  y el IDE para el lenguaje de programación será Netbeans IDE 16.

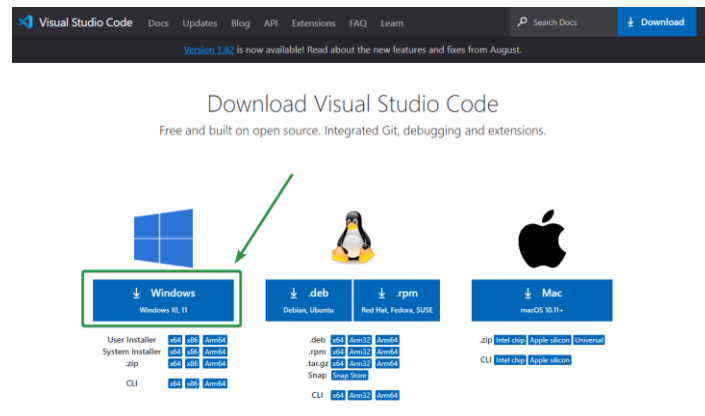
Para la instalación del IDE Visual Studio Code hay que seguir los siguientes pasos.

Paso 1: Descarga de VS Code

Dirígete al sitio web oficial de Visual Studio Code en <https://code.visualstudio.com/> y busca un botón de descarga, generalmente ubicado en la página principal. Aquí deberías encontrar una imagen que representa el botón de descarga.

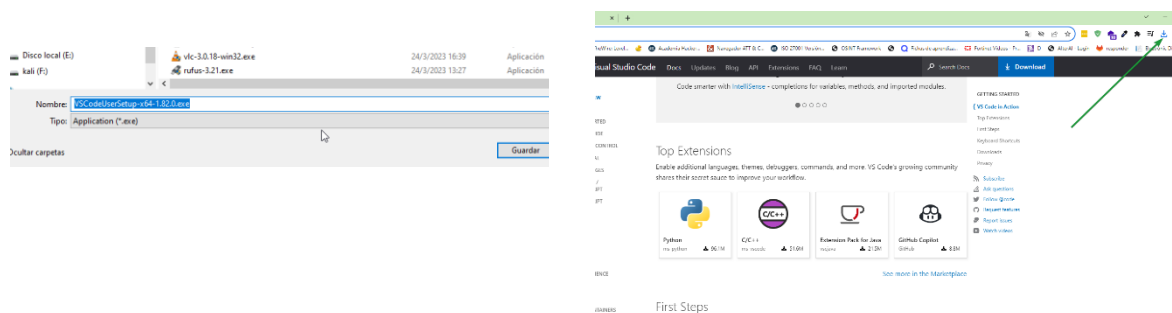


Paso 2: Selecciona tu Sistema Operativo, en mi caso Windows 10.



Paso 3: Descarga del Instalador

Después de seleccionar tu sistema operativo, deberías ser redirigido a una página que te permitirá descargar el instalador específico para tu plataforma. Aquí encontrarás un enlace de descarga o un botón para iniciar la descarga, donde señalarás la carpeta y realizará la descarga.



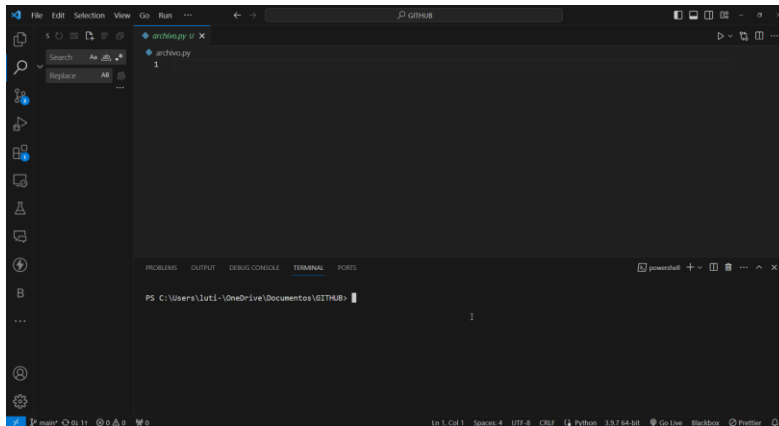
Paso 4: Ejecuta el Instalador

Una vez que se complete la descarga, ejecuta el instalador haciendo doble clic en él (en Windows) o siguiendo las instrucciones específicas para tu sistema operativo.

Durante la instalación, se te pueden presentar opciones de configuración. Estas opciones pueden variar según la versión y la plataforma. Generalmente, se te preguntará si deseas crear accesos directos en el escritorio o en el menú de inicio (en el caso de Windows), así como otras preferencias.

Paso 5: Finaliza la Instalación

Finalmente, cuando se complete la instalación, verás una pantalla que confirma que Visual Studio Code se ha instalado correctamente.



Paso 6: Configuración de los lenguajes.

La configuración de los lenguajes es muy importante ya que este nos permite al momento de ejecutar los archivos con sus diferentes extensiones que corran de manera correcta, para este curso instalaremos Python y luego netbeans por lo que los pasos a seguir son los siguientes:

Paso 1: Descargar Python: Ve al sitio web oficial de Python en <https://www.python.org/downloads/>.

En la página de descargas, verás las últimas versiones de Python disponibles. Elige la versión más reciente de Python 3 (por ejemplo, Python 3.9.6).

Desplázate hacia abajo hasta encontrar la sección de descargas de archivos ejecutables. Dependiendo de tu sistema operativo (Windows, macOS o Linux), selecciona el enlace de descarga adecuado. Para Windows, es posible que veas dos opciones: una para 32 bits y otra para 64 bits. Si tienes un sistema operativo de 64 bits, selecciona la versión de 64 bits.

Paso 2: Ejecutar el Instalador de Python: Una vez que se complete la descarga, haz doble clic en el archivo ejecutable que descargaste para iniciar el instalador de Python.

Asegúrate de marcar la casilla que dice "Add Python X.Y to PATH" (donde "X.Y" representa la versión de Python que estás instalando). Esto añadirá Python al PATH de tu sistema, lo que facilitará su uso desde la línea de comandos.

Haz clic en "Install Now" para comenzar la instalación.

Paso 3: Verificar la Instalación de Python: Abre una terminal o línea de comandos, Escribe el siguiente comando y presiona Enter:

```
bash
Copy code
python --version
```

Esto mostrará la versión de Python que has instalado, como "Python 3.9.6" o similar.

Instalación del Netbeans.

1.3. Sistemas de control de versionamiento (vcs).

Un sistema de control de versiones es como una máquina mágica que ayuda a los equipos que hacen programas de computadoras a trabajar juntos y a no perder ningún cambio que hagan

en sus programas. También les ayuda a volver atrás en el tiempo si algo sale mal y a resolver problemas si dos personas hacen cambios diferentes al mismo tiempo. Git y GitHub son como dos superhéroes que hacen esto muy bien. Git es como la herramienta que usamos para guardar y ver los cambios, y GitHub es como un lugar seguro en internet donde podemos guardar nuestros programas para que todos los demás puedan ayudar.

1.3.1. Git y GitHub.

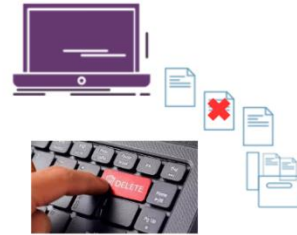
Para un mejor entendimiento veamos la siguiente secuela de imágenes.

A



Un vsc, es un repositorio donde podemos guardar la información que utilizaremos en un proyecto.

B



Que pasa si realizamos una modificación o eliminación de un archivo sea de manera intencional o accidental, y queremos volver al archivo a como estaba antes.

C



La mayoría diría ctrl+z, pero en caso que este archivo este guardado y no permita revertir el cambio.

D



Estaríamos realmente preocupados, en especial si se trata de un archivo muy importante, pues para eso nos sirve el vsc.

E

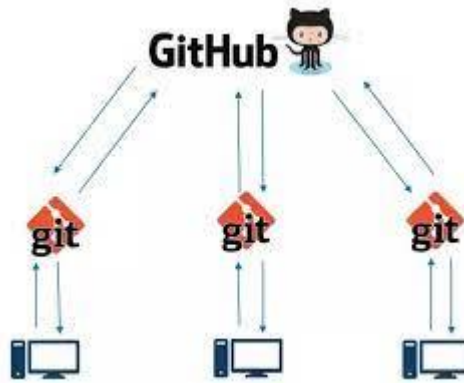


Haz de referencia que con el vsc podrían utilizar una máquina del tiempo, y regresar al pasado y poder recuperar

F



En conclusión, se podría tener los archivos en un repositorio de manera segura.



Además, podemos otorgar permisos a otras personas para que trabajen de manera simultánea, con la posibilidad de recuperar las versiones de los archivos cuando los requiramos.

Para el manejo del sistema de control de versión se utiliza una herramienta en Windows llamada Git, para el uso de este sistema de manera online, se utiliza plataforma web, la más conocida es GitHub.

	Git	GitHub
Definición	Git es un sistema de control de versiones distribuido y de código abierto que se utiliza para rastrear los cambios en el código fuente de un proyecto de software.	GitHub es una plataforma en línea basada en la web que utiliza Git como sistema de control de versiones subyacente. Proporciona servicios de alojamiento de repositorios en línea y herramientas de colaboración para equipos de desarrollo.
Función principal:	Git se utiliza para gestionar y controlar el historial de versiones de un proyecto, lo que incluye realizar un seguimiento de los cambios, crear ramas de desarrollo, fusionar cambios y revertir a versiones anteriores.	GitHub se utiliza para alojar repositorios de Git de forma remota en servidores en línea. Permite a los equipos colaborar en proyectos, realizar seguimiento de problemas, revisar y aprobar cambios, y gestionar la colaboración de manera efectiva.
Ubicación:	Git se instala en tu computadora local y se utiliza para trabajar en un repositorio de código en tu máquina.	GitHub está en línea y aloja repositorios Git en servidores remotos. Los usuarios acceden a los repositorios y colaboran a través de la plataforma web de GitHub.
Acceso	Puedes usar Git sin estar conectado a internet. Es ideal para el desarrollo local y colaborativo en redes privadas.	Para trabajar con GitHub, necesitas una conexión a internet. Es ideal para colaboración en línea y proyectos de código abierto.

1.3.2. Instalación.

Para la instalación del Git en un sistema operativo Windows, se realiza el siguiente procedimiento.

- Ve al sitio web oficial de Git en <https://git-scm.com/download/win>.
- Descarga el instalador para Windows haciendo clic en el enlace "64-bit Git for Windows Setup" o "32-bit Git for Windows Setup" según tu versión de Windows.
- Ejecuta el archivo descargado para iniciar el instalador.
- En la primera pantalla del instalador, puedes dejar las opciones predeterminadas y hacer clic en "Next" (Siguiente).

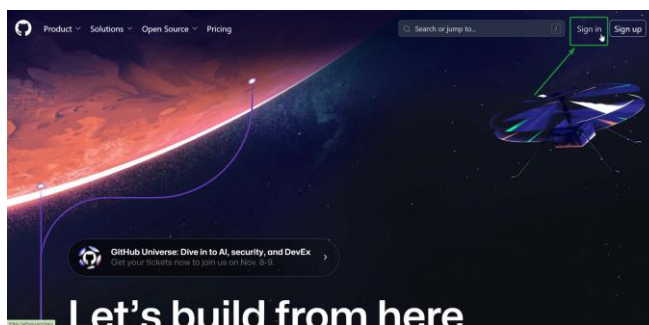
- Selecciona "Use Git from Git Bash only" para utilizar Git desde la línea de comandos de Git Bash o "Use Git from the Windows Command Prompt" si deseas utilizar Git desde la línea de comandos de Windows. Luego, haz clic en "Next" (Siguiente).
- Selecciona "Use the OpenSSL library" y deja las demás opciones predeterminadas. Haz clic en "Next" (Siguiente).
- Selecciona "Checkout Windows-style, commit Unix-style line endings" y haz clic en "Next" (Siguiente).
- Elige "Use Windows' default console window" y haz clic en "Next" (Siguiente).
- En la pantalla "Select the default behavior of 'git pull'", elige "Rebase" y haz clic en "Next" (Siguiente).
- Selecciona "Enable file system caching" y haz clic en "Next" (Siguiente).
- Elige "Enable Git Credential Manager" y haz clic en "Next" (Siguiente).
- En la pantalla "Configuring the line ending conversions", selecciona "Checkout as-is, commit Unix-style line endings" y haz clic en "Next" (Siguiente).
- Haz clic en "Install" (Instalar) para comenzar la instalación de Git.
- Espera a que se complete la instalación y luego haz clic en "Finish" (Finalizar).

Instalación de GitHub.

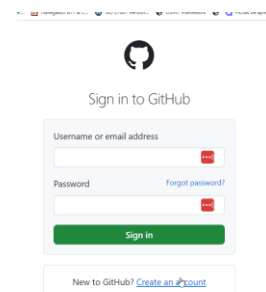
GitHub no se instala en tu computadora como una aplicación independiente. Es una plataforma en línea que se utiliza a través de un navegador web para alojar repositorios de código y colaborar en proyectos de desarrollo de software. Aquí tienes un paso a paso sobre cómo utilizar GitHub:

1. Crear una cuenta en GitHub:

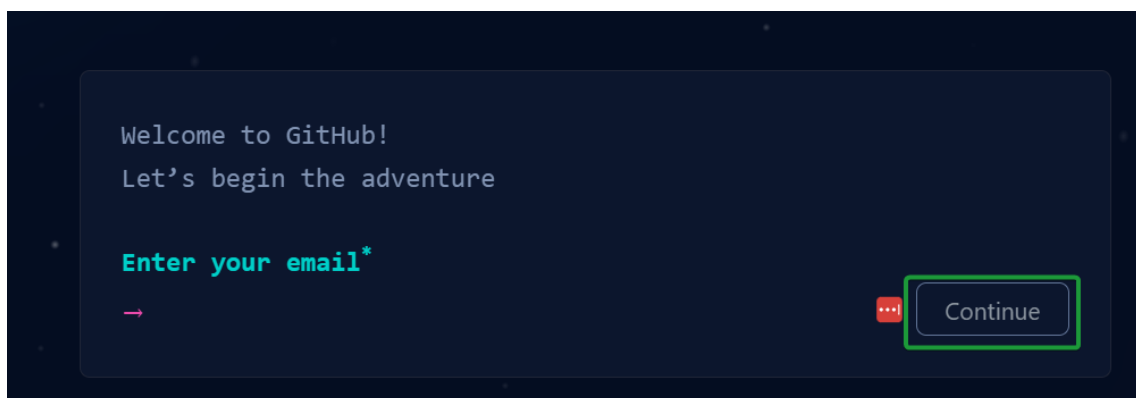
Abre tu navegador web y ve al sitio web oficial de GitHub en <https://github.com/>. Haz clic en "Sign up" (Registrarse) para crear una cuenta. Deberás proporcionar una dirección de correo electrónico válida, un nombre de usuario y una contraseña.



Ventana de GitHub



Ventana Login GitHub



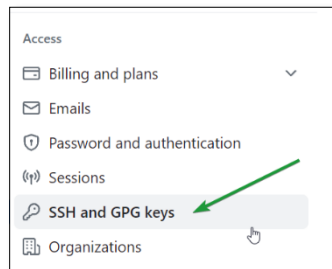
Completar el formulario

1.3.3. Manejo de GitHub.

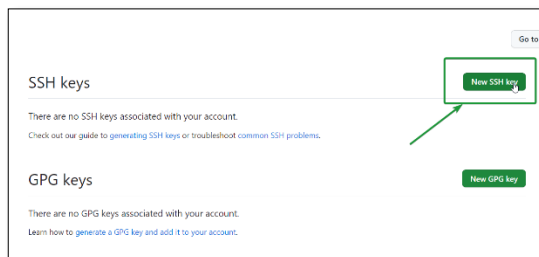
Para utilizar GitHub en el ordenador se debe realizar los siguientes pasos:

1. Configurar el equipo para realizar la conexión con la plataforma de GitHub, mediante una conexión en este caso mediante el protocolo **SSH**.
 - a. Abrir Powershell como administrador.
 - b. Utilizar el siguiente comando. **cd C:/Users/nombre-usuario/.ssh**
 - c. Escribir el siguiente código. **ssh-keygen.exe -o -t rsa -C "correo@dominio.com"** el correo es con el que te registraste en GitHub.
 - d. Leer la clave con el comando pública y copiarla. **type .\id_rsa.pub**
 - e. Cargar llave a GitHub.
 - Iniciar sesión de GitHub.
 - Cargar la llave a GitHub, como se ve en la imagen de la derecha.
 - Cargar las llaves conforme los siguientes pasos.

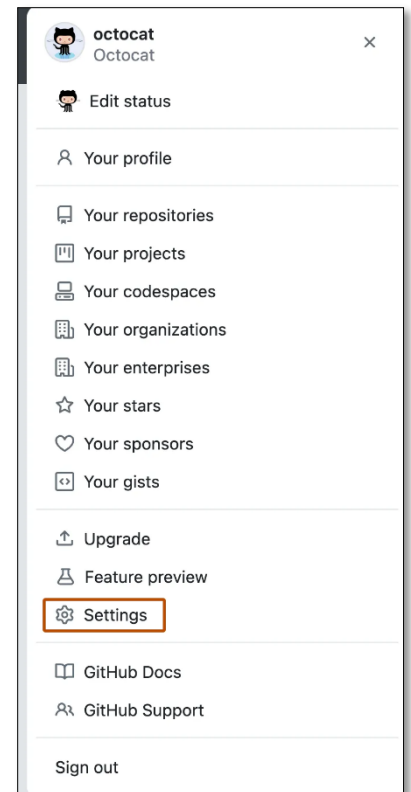
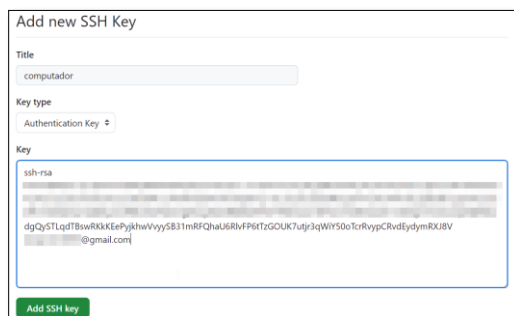
2



3



4



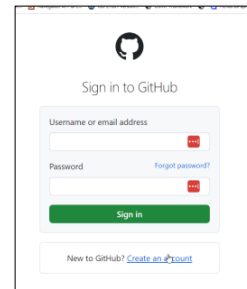
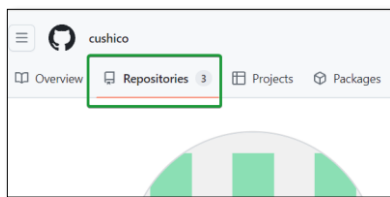
1

Finalizado estos pasos que permiten la conexión entre nuestro GitHub, y el equipo que utilizaremos podemos empezar a crear nuestro repositorio, si vas a manejar desde otro equipo adicional debes realizar nuevamente esta configuración, recuerda borrar las credenciales de equipos que no utilices.

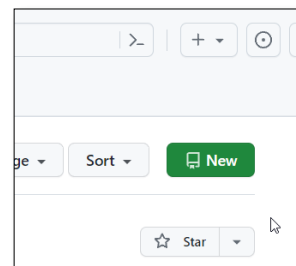
2. Crear un repositorio.

Una vez que hayas creado una cuenta, inicia sesión en GitHub utilizando tu nombre de usuario y contraseña e ingresado a la plataforma de GitHub, crearemos un repositorio el cuál permitirá tener tu espacio de trabajo en la nube, para realizar esto se realizan los siguientes pasos:

1) Ingresamos a la plataforma de GitHub.



2) Vamos a la pestaña **Repositorios**.



3) Seleccionamos **New**

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner * cushico Repository name *

Great repository names are short and memorable. Need inspiration? How about [silver-giggle](#) ?

Description (optional)

☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

☐ **Add a README file**
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore
.gitignore template: **None**

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license
License: **None**

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

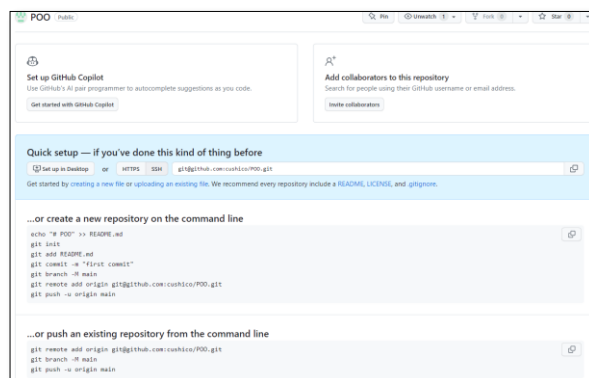
📘 You are creating a public repository in your personal account.

Create repository

4) Configuramos el nombre del repositorio, el acceso al repositorio.

Publico: Acceso a todas las personas.
Privado: Acceso personal.

- 5) Obtenemos una configuración paso a paso que tendremos que seguir desde el ordenador.



- Una vez obtenida la información procedemos a la configuración en el ordenador con siguiendo los siguientes pasos.

<p>1) Seleccionamos una carpeta que nos servirá para conectar con el repositorio</p>	
<p>2) Accedemos a la carpeta desde la consola.</p>	
<p>3) Ejecutamos los siguientes comandos.</p> <pre>git config --global user.email "@gmail.com" git config --global user.name "Your user Name"</pre>	
<p>4) Ejecutar los siguientes comandos.</p> <pre>git init git branch -M main</pre>	
<p>5) Los pasos obtenidos en la plataforma de github, copiamos y pegamos los comandos que faltarían.</p> <pre>git remote add origin git@github.com:cushico/POO.git</pre>	<p>El error que sale, es porque ya está registrado el remote origin</p>

Una vez configurado, podemos realizar las pruebas para subir nuestro primer archivo a GitHub, para lo cual tendremos que tener presente la siguiente imagen.

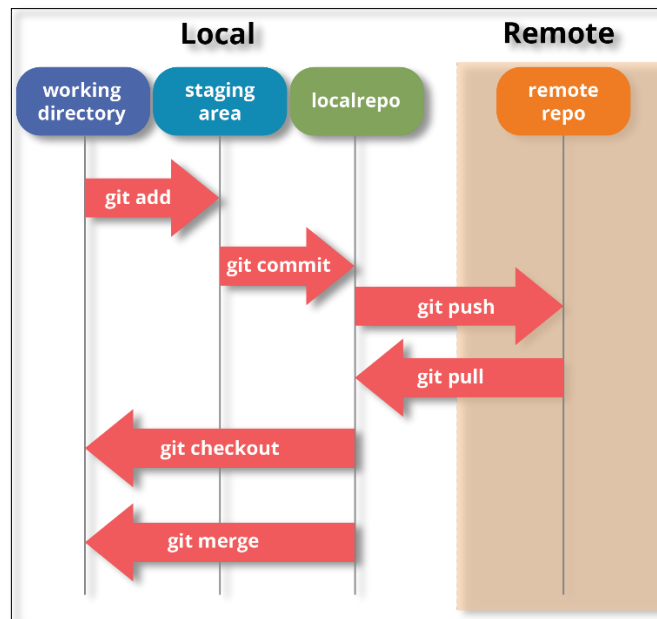
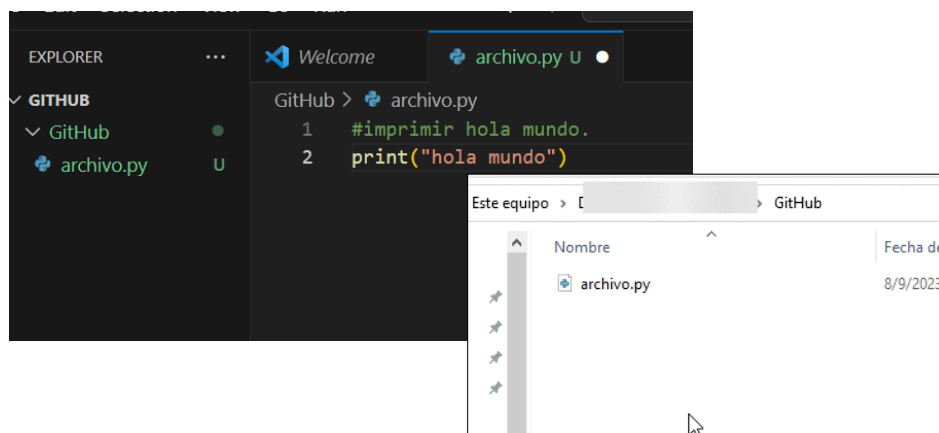


Imagen principal GitHub

- Cargar un archivo a GitHub.
 - Creamos un archivo que utilizaremos para subir a GitHub.



- Con la opción `git status -s` podemos observar el estado de los archivos.



- Pasamos al staging area, como vimos en la imagen principal de GitHub, con el comando **git add**.

```
GitHub> git add archivo.py
```

- Para pasar al repositorio local, colocamos el comando.

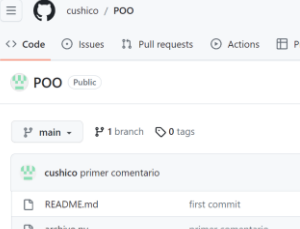
```
PS C:\Users\l...> \GitHub> git status -s
D README.md
A archivo.py
PS C:\Users\l...> \GitHub> git commit -m "primer comentario"
[main 3d59b71] primer comentario
1 file changed, 2 insertions(+)
create mode 100644 archivo.py
PS C:\Users\l...> \GitHub>
```

Revisamos que el archivo esté en estado agregado "A"

Agregamos el comentario

git commit "comentario que quiera colocar"

- Para pasar del repositorio local a la nube, utilizamos el comando

<pre>PS C:\Users\l...> \GitHub> git push -u origin main Enumerating objects: 6, done. Counting objects: 100% (6/6), done. Delta compression using up to 8 threads Compressing objects: 100% (3/3), done. Writing objects: 100% (6/6), 493 bytes 246.00 KiB/s, done. Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 To github.com:cushico/POO.git * [new branch] main -> main branch 'main' set up to track 'origin/main'.</pre>		<p>git push -u origin main</p>
<p>Una vez subida la información podemos comprobar en la plataforma</p>		

Errores que pueden llegarse a producirse.

<p>Verifica en tu carpeta principal de GitHub local, la carpeta /.git/ dentro de la misma encontrarás un documento de configuración. Con el comando type config, podemos observar en url que sea la ruta a la cuál queremos cargar el archivo. En caso que se necesite modificar se puede utilizar el comando git remote set-url origin https://github.com/nuevo-usuario/nuevo-repo.git</p>	<pre>PS C:\> \GitHub> cd .. PS C:\> > cd .\git\ PS C:\> \git> type config [core] repositoryformatversion = 0 filemode = false bare = false logallrefupdates = true symlinks = false ignorecase = true [remote "origin"] url = git@github.com:cushico/POO.git fetch = +refs/heads/*:refs/remotes/origin/* [branch "main"] remote = origin merge = refs/heads/main</pre>
<p>Si deseas agregar un nuevo control remoto en lugar de cambiar uno existente, puedes hacerlo de esta</p>	<p>git remote add nuevo_nombre nuevo_url_del_repositorio</p>

manera:

- Creando un archivo desde la plataforma de GitHub.

The first screenshot shows the GitHub repository page for 'cushico / POO'. The 'Add file' dropdown menu is open, showing 'Create new file' (highlighted) and 'Upload files'. The repository contains files 'README.md' and 'archivo.py'.

The second screenshot shows the file editor for 'archivo2.py' in the 'main' branch. The file content is `print ("nuevo archivo")`. The 'Edit' tab is active.

The third screenshot shows the 'Commit changes' dialog box. The 'Commit message' field contains 'Create archivo2.py'. The 'Extended description' field contains 'nuevo archivo desde github'. The 'Commit directly to the main branch' radio button is selected. The 'Commit changes' button is highlighted.

-

Como eliminar un archivo.

```
rm .\archivo.py
git rm archivo.py
git commit -m "archivo borrado"
git push -u origin main
git checkout
git checkout 3d20b21d895cada1702acb819027899c818ba0d7 /archivo.py
git log
git show <hash_del_commit>
git log
```

branch

3. Crear un repositorio:

Un repositorio en GitHub es como una carpeta en línea donde puedes guardar tu código.

Haz clic en el botón "New" (Nuevo) en la página principal de GitHub para crear un nuevo repositorio. Sigue las instrucciones para darle un nombre y una descripción.

5. Clonar un repositorio (opcional):

Si estás trabajando en un proyecto existente en GitHub, puedes "clonar" el repositorio a tu computadora local utilizando Git. Para ello, necesitarás tener Git instalado en tu máquina.

Ve al repositorio en GitHub y haz clic en el botón "Code". Luego, copia la URL del repositorio. Abre la terminal o línea de comandos en tu computadora y utiliza el comando `git clone` seguido de la URL para clonar el repositorio.

6. Subir código a tu repositorio:

Puedes subir tu código a tu repositorio de GitHub utilizando Git y comandos como `git add`, `git commit`, y `git push`.

Consulta la documentación de GitHub para obtener más información sobre cómo subir tu código.

7. Colaborar en proyectos:

Puedes colaborar en proyectos de código abierto o en equipo a través de GitHub. Puedes enviar solicitudes de extracción (pull requests), crear "issues" para hacer seguimiento de tareas y mucho más.

8. Explorar proyectos:

Además de crear tus propios repositorios, puedes explorar proyectos de código abierto y contribuir a ellos.

GitHub es una herramienta poderosa para la colaboración en desarrollo de software y el control de versiones en línea. Una vez que tengas una cuenta y un repositorio, puedes comenzar a utilizar GitHub para alojar, colaborar y contribuir a proyectos de código abierto y privados.