



Universidad de las Fuerzas Armadas

Informe N° 1: Del proceso que realizamos

Nombre y Apellido:

Sheyla Bernal

Programación Orientada a Objet

Docente:

LUIS ENRIQUE JARAMILLO MONTAÑO

Fecha:

17 de noviembre del 2024

Contenido

INTRODUCCION	3
DESARROLLO.....	4
• Proceso:	4
➤ Creamos una cuenta en GitHub	4
➤ Descargamos Git.....	4
➤ Creamos un repositorio.....	4
➤ Creamos una clave SSH.....	5
➤ Configurar la clave ssh en GitHub	6
CONCLUSIONES	9
BIBLIOGRAFIA.....	10

INTRODUCCION

GitHub es un servicio basado en la nube que aloja un sistema de control de versiones (VCS) llamado Git. Éste permite a los desarrolladores colaborar y realizar cambios en proyectos compartidos, a la vez que mantienen un seguimiento detallado de su progreso.

- **GIT :**

Git es un sistema de control de versiones distribuido, lo que permite a cualquier desarrollador del equipo con acceso gestionar tanto el código fuente como su historial de cambios a través de herramientas de línea de comandos.

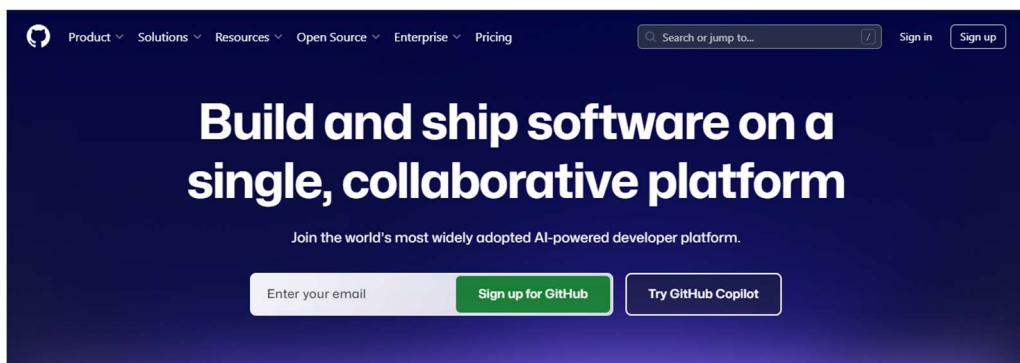
A diferencia de los sistemas de control de versiones centralizados, Git permite trabajar con ramas de características. Esto significa que cada integrante del equipo puede crear una rama separada en su repositorio local, lo que les brinda un entorno aislado para realizar modificaciones en el código.

Estas ramas de características no afectan directamente a la rama principal, donde se encuentra el código base del proyecto. Una vez que los cambios estén completos y listos para integrarse, la rama de características puede fusionarse con la rama principal, aplicando así las actualizaciones al proyecto.

DESARROLLO

🌈 Proceso:

➤ Creamos una cuenta en GitHub



➤ Descargamos Git



➤ Creamos un repositorio


- Ponemos como queremos que se llame nuestro repositorio
- Lo podemos poner si queremos que quede Publico o privado

Crear un nuevo repositorio

Un repositorio contiene todos los archivos del proyecto, incluido el historial de revisiones. ¿Ya tienes un repositorio de proyectos en otro lugar? [Importa un repositorio.](#)

Los campos obligatorios están marcados con un asterisco (*).



Dueño * Nombre del repositorio *

 Sheylabc05 /

Los buenos nombres de repositorios son breves y fáciles de recordar. ¿Necesitas inspiración? ¿Qué tal?

pulpo-palito de pescado verboso ?

Descripción (opcional)

- ☒  **Público**
Cualquier persona en Internet puede ver este repositorio. Tú eliges quién puede contribuir.
- ☐  **Privado**
Tú eliges quién puede ver y comprometerse con este repositorio.

➤ Creamos una clave SSH

- Abrimos Git Bash
- Vamos a la carpeta SSH

```
User@DESKTOP-PE0MV73 MINGW64 ~
$ cd ~/.ssh

User@DESKTOP-PE0MV73 MINGW64 ~/.ssh
$
```

- Vamos a poner el siguiente comando

```
User@DESKTOP-PE0MV73 MINGW64 ~/.ssh
$ ssh-keygen -t ed25519 -C "bernalshey1a986@gmail.com" -f id_ed25519
```

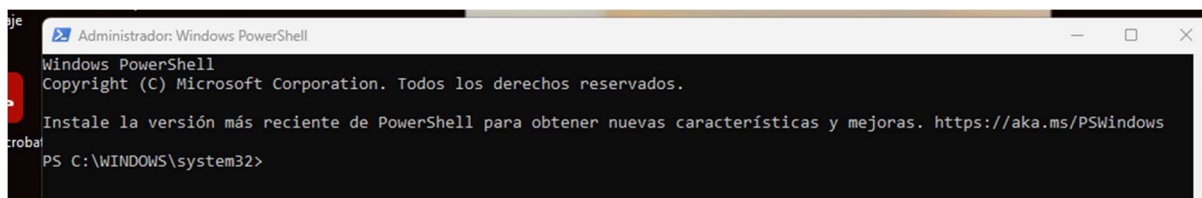
- A lo que ejecutamos nos va a pedir una contraseña extra

```
Your identification has been saved in id_ed25519
Your public key has been saved in id_ed25519.pub
The key fingerprint is:
SHA256:c5JBMm07whDXPttL481CtiME8jpL970JP0DVgZcFpna san99tiago@gmail.com GitHub
The key's randomart image is:
+--[ED25519 256]--+
|      =+o Eo+=.   |
|      .=0000+     |
|      000..       |
|      . ..=0      |
|      O =S..      |
|      o oBo       |
|      ...O++      |
|      .O. ..*=#   |
|      .O  ..B=O   |
+-----[SHA256]-----+
```

- Una vez echo esto se nos generan dos archivos uno que es la llave publica y otro la llave privada

```
-rw-r--r-- 1 User 197121 484 Nov 17 13:19 id_ed25519
-rw-r--r-- 1 User 197121 114 Nov 17 13:19 id_ed25519.pub
```

- Abrimos como administrador Windows PowerShell



- Vamos a ir a la carpeta donde se va a ejecutar el repositorio

```
PS C:\WINDOWS\system32> cd C:\Users\santi\Documents\BETEK\
```

- Insertamos los sig comandos

```
PS C:\Users\santi\Documents\BETEK> ssh-agent status
PS C:\Users\santi\Documents\BETEK> Get-Service -Name ssh-agent | Set-Service -StartupType Manual
PS C:\Users\santi\Documents\BETEK> ssh-agent start
PS C:\Users\santi\Documents\BETEK> ssh-add.exe C:\Users\santi\.ssh\id_ed25519_
```

- Nos va a pedir la clave que insertamos

```
Enter passphrase for C:\Users\santi\.ssh\id_ed25519:
Identity added: C:\Users\santi\.ssh\id_ed25519 (san99tiago@gmail.com GitHub)
PS C:\Users\santi\Documents\BETEK> 
```

- Una vez hecho esto vamos a Git Bash y insertamos la llave publica

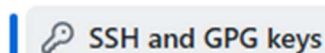
```
User@DESKTOP-PEOMV73 MINGW64 ~/.ssh
$ cat id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIBjptwWrr6Vbx5ZvLiS2kzXDL7pij182JhSse1gJY2U
bernalshayla986@gmail.com GitHub
```

➤ Configurar la clave ssh en GitHub

- Vamos a GitHub y entramos en settings



- De hay vamos a SSH and GPG keys



- Ponemos agregar llave y procedemos a llenar lo que nos pide

Add new SSH Key

Title

Key type

Authentication Key

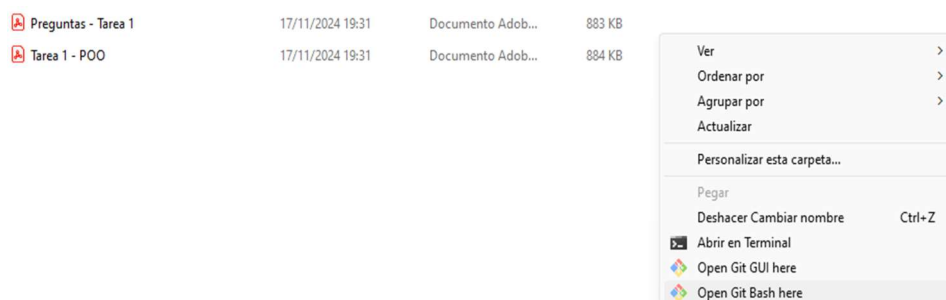
Key

Begins with 'ssh-rsa', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', 'ecdsa-sha2-nistp521', 'ssh-ed25519', 'sk-ecdsa-sha2-nistp256@openssh.com', or 'sk-ssh-ed25519@openssh.com'

Add SSH key

Publicar archivos a GitHub

- Vamos a los archivos y damos clic derecho en donde nos va a salir git bash here y seleccionas esa opción



- Una vez echo esto nos va a aparecer lo que deseamos subir



- Vamos a insertar los comandos que nos da una vez creado el repositorio

...o crear un nuevo repositorio en la línea de comandos

```
echo "# yo" >> README.md
git init
git add README.md
git commit -m "primer commit"
git branch -M main
git remote add origin https://github.com/Sheylabc05/yo.git
git push -u origin main
```

- Insertado el 1r comando nos aparecerá lo sig

```
User@DESKTOP-PE0MV73 MINGW64 ~/Desktop/Lulu
$ git init
Initialized empty Git repository in C:/Users/User/Desktop/Lulu/.git/
```

- Añadimos los dos siguientes comandos y nos aparecerá los archivos que vamos a subir al repositorio

```
User@DESKTOP-PE0MV73 MINGW64 ~/Desktop/Lulu (master)
$ git add .

User@DESKTOP-PE0MV73 MINGW64 ~/Desktop/Lulu (master)
$ git commit -m "se agrega fotos"
[master (root-commit) ef26239] se agrega fotos
5 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Imagen de WhatsApp 2023-12-30 a las 00.22.14_4c99ac4e.jpg
create mode 100644 Imagen de WhatsApp 2024-07-02 a las 22.26.51_c6f3ecbb.jpg
create mode 100644 Imagen de WhatsApp 2024-07-02 a las 22.39.58_a7bdd03f.jpg
create mode 100644 WhatsApp Image 2023-11-23 at 3.34.38 PM.jpeg
create mode 100644 lulu.jpg
```


- Insertamos los dos últimos códigos y ya se subieron los archivos

```
User@DESKTOP-PE0MV73 MINGW64 ~/Desktop/Lulu (master)
$ git branch -M main

User@DESKTOP-PE0MV73 MINGW64 ~/Desktop/Lulu (main)
$ git remote add origin https://github.com/Sheylabc05/yo.git

User@DESKTOP-PE0MV73 MINGW64 ~/Desktop/Lulu (main)
$ git push -u origin main
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 4 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 270.52 KiB | 22.54 MiB/s, done.
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Sheylabc05/yo.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

- Y refrescamos la página y ya está subido los archivos al repositorio


Tarea-1
Public
Unpin
Unwatch 1

main
1 Branch
0 Tags

Add file
Code

Commit	Message	Time
392a67b	Update Código	2 hours ago
	3 Commits	
Código	Update Código	2 hours ago
Preguntas - Tarea 1.pdf	Tarea 1	3 days ago
Tarea 1 - POO.pdf	Tarea 1	3 days ago

CONCLUSIONES

GitHub es una plataforma esencial para el trabajo colaborativo en proyectos de desarrollo, basada en Git, un poderoso sistema de control de versiones distribuido. Algunas conclusiones clave son:

1. Control de versiones eficiente: Git permite gestionar de manera efectiva el código fuente y su historial, proporcionando un registro detallado de todos los cambios realizados.
2. Trabajo en equipo mejorado: Gracias a su modelo distribuido, cada desarrollador puede trabajar de forma independiente en ramas locales, evitando conflictos directos en la rama principal.
3. Aislamiento de cambios: Las ramas de características ofrecen un entorno aislado para desarrollar nuevas funcionalidades o corregir errores, reduciendo el riesgo de introducir errores en el código base.
4. Facilidad de integración: Una vez que los cambios son verificados y considerados listos, las ramas de características pueden fusionarse con la rama principal, asegurando una integración controlada y ordenada.

Git y GitHub son herramientas imprescindibles para gestionar proyectos de software, facilitando la colaboración, la innovación y la calidad en el desarrollo.

BIBLIOGRAFIA

<https://www.hostinger.es/tutoriales/que-es-github>

<https://www.youtube.com/watch?v=Y9jXwyDbTmQ&t=190s>