



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**  
**FACULTAD DE CONTADURÍA Y ADMINISTRACIÓN**



**DIVISIÓN SISTEMA UNIVERSIDAD ABIERTA Y EDUCACIÓN A DISTANCIA**

## **PROGRAMACION DE DISPOSITIV.MOVILES**

### **UNIDAD 2**

#### **ACTIVIDAD 1**

##### **NOMBRE DEL ALUMNO:**

PAULINA RODRIGUEZ SAMPEDRO

##### **NOMBRE DEL ASESOR:**

CRISTIAN CARDOSO ARELLANO

##### **SEXTO SEMESTRE**

GRUPO: 9696

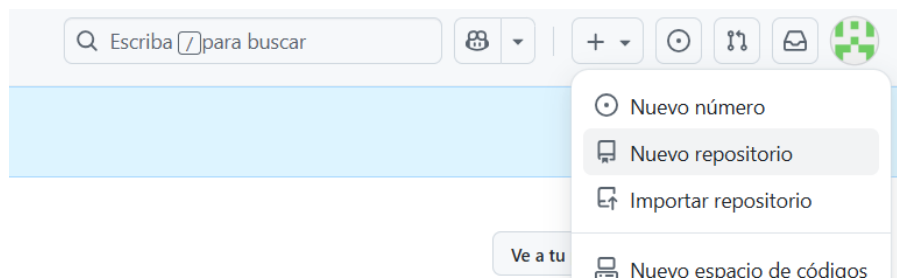
## Mi segundo repositorio en Git/Github

Objetivos de la actividad:

- Conocer las ventajas del sistema de control de versiones con protocolo de seguridad Secure Shell Protocol (SSH).
- Aplicar correctamente el ciclo de vida de un cambio en Git, que incluye: preparar (git add), guardar (git commit) y sincronizar (git push) los cambios entre el entorno local y el repositorio remoto.
- Comprender la utilidad de las ramas al crear una rama secundaria para desarrollar una tarea específica de manera aislada de la rama principal, promoviendo un desarrollo organizado.
- Generar un commit con un mensaje claro y estructurado, asegurando que el historial de versiones sea legible y rastreable para futuros colaboradores.

### Desarrollo de la actividad

Ilustración 1. Creación de segundo repositorio en Github



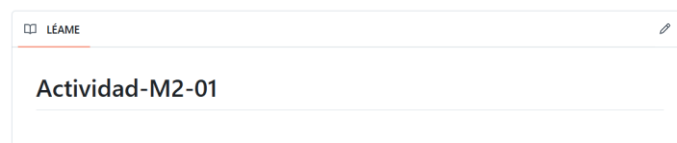
Se inició la creación de un nuevo repositorio remoto en la plataforma GitHub. Esto se hizo en la interfaz web para establecer un nuevo proyecto independiente de cualquier otro trabajo previo, cumpliendo con el requisito de tener un segundo repositorio público.

## Ilustración 2. Asignación de nombre al repositorio

The screenshot shows the GitHub repository creation interface. It is divided into two sections: 'General' and 'Configuración'. In the 'General' section, the owner is 'PaulinaRS27' and the repository name is 'Actividad-M2-01', which is marked as available. A description field is empty. In the 'Configuración' section, the visibility is set to 'Público'. The 'Agregar README' option is checked. The 'Añadir .gitignore' option is set to 'No hay archivo .gitignore'. The 'Agregar licencia' option is set to 'Sin licencia'. A green 'Crear repositorio' button is at the bottom right.

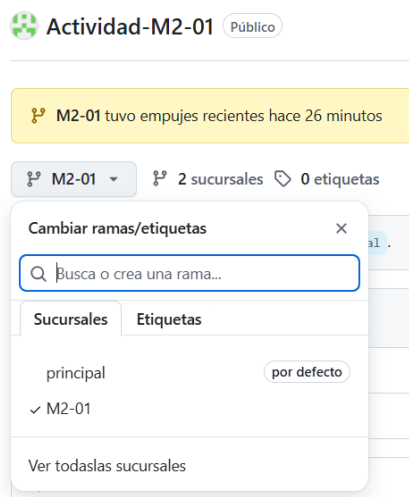
Se le asignó un nombre descriptivo (Actividad-M2-01) y se seleccionó la visibilidad pública. Se marcó la opción de añadir un archivo README (README.md) para inicializar el repositorio con un archivo y facilitar la clonación inmediata.

## Ilustración 3. Repositorio creado



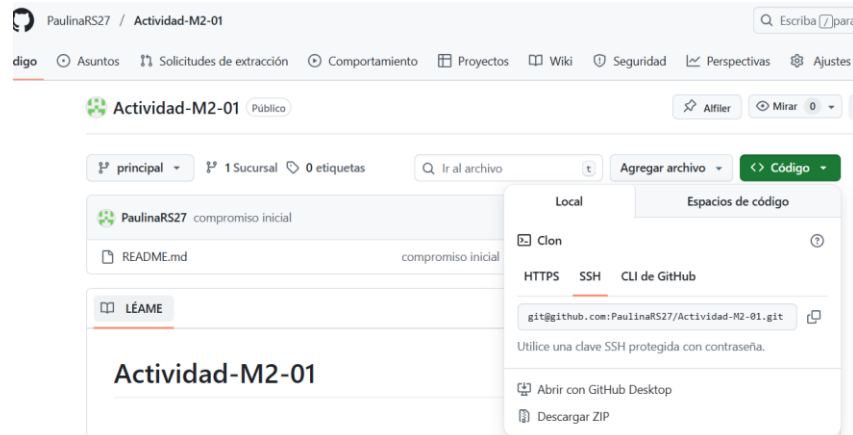
Se confirma la finalización del proceso, mostrando el repositorio listo en la nube.

## Ilustración 4. Existencia de la rama M2-01



Tras clonar el repositorio localmente, se ejecutó el comando `git checkout -b M2-01` en la terminal Git Bash. Este comando sirvió para crear una nueva rama de desarrollo llamada M2-01 y, al mismo tiempo, cambiar a ella, permitiendo trabajar en la tarea específica de forma aislada de la rama principal.

### Ilustración 5. Obtención de SSH



En la interfaz web de GitHub, se accedió al botón `<> Código` para obtener la URL del repositorio. Se seleccionó específicamente la pestaña SSH para copiar la dirección URL. Este es el paso previo para asegurar que la comunicación entre la computadora local y GitHub se realice a través del protocolo Secure Shell (SSH), el cual utiliza la llave pública previamente configurada para la autenticación, ofreciendo mayor seguridad y comodidad que HTTPS.

### Ilustración 6. Ejecución por SSH en terminal Git Bash

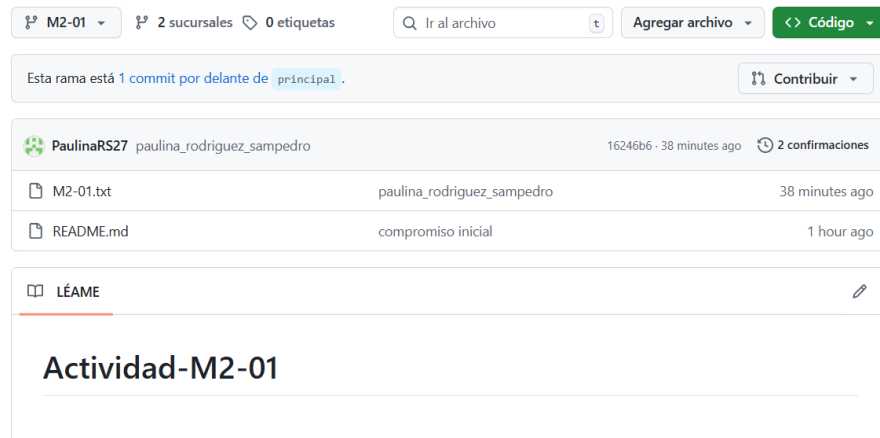
```
MINGW64/c/Users/Kim/OneDrive/Escritorio/SEXTO SEMESTRE/PROGRAMACION DE DISP.MOVILES/UNIDAD 2
/c/Users/Kim/OneDrive/Escritorio/SEXTO SEMESTRE/PROGRAMACION DE DISP.MOVILES/UNIDAD 2
Kim@LAPTOP-I9VHPQBF MINGW64 ~/OneDrive/Escritorio/SEXTO SEMESTRE/PROGRAMACION DE DISP.MOVILES/UNIDAD 2
$ dir
U2A1  android_2026_m2_01_Rodriguez_Sampedro_Paulina.docx

Kim@LAPTOP-I9VHPQBF MINGW64 ~/OneDrive/Escritorio/SEXTO SEMESTRE/PROGRAMACION DE DISP.MOVILES/UNIDAD 2
$ git clone git@github.com:PaulinaRS27/Actividad-M2-01.git
Cloning into 'Actividad-M2-01'...
The authenticity of host 'github.com (140.82.114.4)' can't be established.
ED25519 key fingerprint is: SHA256:+DiY3wvV6TuJJhpZisF/zLDA0zPMSvHdkr4UvCOqU
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
```

Se demuestra la ejecución exitosa del comando `git clone` en la terminal local (Git Bash). El comando fue: `git clone git@github.com:PaulinaRS27/Actividad-M2-01.git`.

La utilización de la URL SSH confirma la ejecución por SSH del repositorio, descargando una copia del mismo a la computadora.

### Ilustración 7. Commit generado y presencia del archivo M2-01.txt



Se creó y se agregó el archivo M2-01.txt con el comando `touch M2-01.txt`. Se añadió el archivo con el comando `git add M2-01.txt`. Posteriormente se registró el cambio con el commit con el comando `git commit -m "paulina_rodriguez_sampedro"`. Finalmente, se subieron los cambios a GitHub con el comando `git push origin M2-01`, confirmando la presencia del archivo y el mensaje de commit en la rama M2-01.

## Conclusión

La actividad realizada ha sido una inmersión completa y práctica en el Sistema de Control de Versiones Git, comenzando con la utilización de los comandos fundamentales como `git config` para establecer la identidad del autor (nombre y correo), seguido de `ssh-keygen` y `ssh-add` para generar y activar las llaves SSH. Esto cumplió con el objetivo de establecer una conexión segura y autenticada con GitHub, eliminando la dependencia de contraseñas y aprovechando el protocolo Secure Shell.

Posteriormente el comando `git clone` fue esencial para descargar una copia de los repositorios remotos a la máquina local, permitiendo iniciar el trabajo. El comando `git checkout` para gestionar ramas, creando y moviéndose entre los diferentes contextos de los repositorios.

El comando `touch` se usó para crear los archivos, `git add` se empleó para seleccionar los archivos y prepararlos para la inclusión en el historial, `git commit` se utilizó para registrar el cambio permanentemente en la historia local, etiquetándolo con un mensaje específico (en esta actividad se nombró con el nombre completo del

alumno) y por último el comando `git push` para sincronizar y publicar los cambios locales (ramas, archivos y commits) en los repositorios remotos de GitHub.