



Tecnológico Nacional de México **Campus Querétaro**

Presenta(n):

Nombre	Numero de Control
José Alejandro Barrón Hernández	22140883

Estudiante(s) de la carrera(s):

Ingeniería Industrial

Inscrito en:

Materia	Grupo
Investigación de Operaciones I	4A

Docente:

Ángeles Hurtado Luis Alberto

Trabajo:

Standard Operating Procedure (SOP) for create a new branch

Fecha de entrega:

Santiago de Querétaro a 11 de septiembre del 2024

SOP for Create a new branch.

Titulo del SOP	Crear un Branch en GitHub
ID del SOP	SOP-GIT-002
Versión	1.0
Fecha de Emisión	06 de septiembre de 2024
Fecha de Revisión	
Elaboración del SOP	Barrón Hernández José Alejandro
Supervisor del SOP	Ángeles Hurtado Luis Alberto
Aprobado por	
Revisión N.º	1
Estado	Activo

Definiciones y Acrónimos.

- **Branch (Rama):** Una bifurcación del código fuente principal que permite trabajar en tareas de desarrollo de manera aislada.
- **Git:** Un sistema de control de versiones distribuido, utilizado para rastrear cambios en el código fuente durante el desarrollo de software. Git permite a múltiples desarrolladores trabajar en el mismo proyecto de manera simultánea sin conflictos.
- **GitHub:** Una plataforma de alojamiento de código fuente basada en la web que utiliza Git como sistema de control de versiones. Permite a los desarrolladores colaborar, revisar y gestionar proyectos de software de forma remota.
- **Repositorio:** Una ubicación central donde se almacena el código fuente de un proyecto junto con su historial de cambios. Puede estar alojado localmente en una máquina o remotamente en una plataforma como GitHub.
- **Clonar (Clone):** Es el proceso de crear una copia local de un repositorio remoto en GitHub en una máquina local, permitiendo al usuario trabajar en el proyecto de manera independiente.
- **SSH (Secure Shell):** Un protocolo de red utilizado para establecer una conexión segura y cifrada entre dos dispositivos. En el contexto de Git, se utiliza para autenticar y realizar operaciones de clonación o envío de cambios al repositorio remoto sin necesidad de ingresar el nombre de usuario y la contraseña en cada acción.
- **HTTPS (Hypertext Transfer Protocol Secure):** Un protocolo de comunicación utilizado para transferir datos de forma segura a través de la web. También puede usarse para clonar repositorios de GitHub, aunque generalmente requiere autenticación con nombre de usuario y contraseña.
- **Terminal (o Símbolo del Sistema):** Una interfaz de línea de comandos en sistemas operativos como macOS, Linux y Windows, donde los usuarios pueden ejecutar comandos directamente para interactuar con el sistema operativo y herramientas como Git.

- **Directorio:** Una carpeta en tu sistema de archivos donde se almacenan archivos o subcarpetas. En este contexto, es el lugar donde se guardará el repositorio clonado.
- **URL (Uniform Resource Locator):** La dirección utilizada para acceder a un recurso en la web, como un repositorio en GitHub. Puede presentarse en formatos como HTTPS o SSH.
- **cd:** Un comando de línea utilizado para cambiar de directorio en la terminal o símbolo del sistema. Por ejemplo, `cd /ruta/a/tu/directorio` cambiará al directorio especificado.
- **commit:** Una acción de Git que guarda los cambios realizados en el código en el repositorio local, acompañado de un mensaje que describe los cambios.
- **push:** Un comando de Git utilizado para enviar los cambios locales realizados en el repositorio a la versión remota en GitHub.
- **Fetch:** Recupera los cambios del repositorio remoto a la copia local sin integrarlos automáticamente.
- **Clave SSH:** Un par de claves criptográficas (pública y privada) utilizadas para autenticar y asegurar las comunicaciones entre tu máquina local y el servidor de GitHub. La clave pública se almacena en GitHub, mientras que la clave privada se mantiene en tu máquina local.
- **git checkout:** Un comando de Git usado para cambiar de rama.
- **git branch:** Un comando de Git para listar, crear o eliminar ramas.
- **git push:** Un comando de Git para enviar cambios locales a un repositorio remoto.

Propósito del documento:

Proporcionar instrucciones detalladas para crear una nueva rama (branch) en un repositorio de GitHub con sistema operativo macOS. Un branch permite trabajar en nuevas características, correcciones de errores u otras tareas de desarrollo en paralelo al código principal, sin afectar la versión estable del proyecto hasta que los cambios estén listos para ser integrados.

Escenario de aplicación:

Aplicable a cualquier desarrollador o colaborador que necesite trabajar en una nueva característica o corrección de errores sin interferir con la rama principal del proyecto. Este SOP es relevante para usuarios de GitHub con acceso al repositorio y familiaridad básica con Git.

Pre-requisitos:

- Cuenta de GitHub activa.
- Instalación de Git en la máquina local.
- Instalación de Visual Studio Code.
- Conexión a Internet.
- Permisos de acceso al repositorio.
- Repositorio clonado localmente.
- Conocimiento básico de la línea de comandos.
- Configuración de clave SSH.

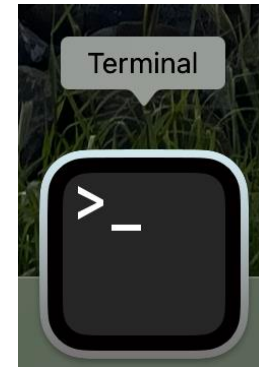
Ciclos del Procedimiento:

1. Verificar que Git esté instalado en la máquina local (git --version).

- Abrir la terminal.
- Escribe el siguiente comando: git --version.



```
alexbarron — zsh — 80x24
Last login: Thu Sep  5 16:26:50 on ttys001
alexbarron@MacBook-Air-de-Alex ~ % git --version
git version 2.46.0
alexbarron@MacBook-Air-de-Alex ~ %
```



- Si git está instalado, te aparecerá una versión impresa (por ejemplo, git version 2.46.0).
- Si no está instalado, descarga e instala Git desde git-scm.com. Sigue las instrucciones del sitio web para el sistema operativo macOS.
- Configura tu nombre de usuario y correo electrónico de GitHub en tu máquina local (Sólo si es primera vez).

```
alexbarron@MacBook-Air-de-Alex ~ % git config user.name "TuNombre"
alexbarron@MacBook-Air-de-Alex ~ % git config user.mail "TuMail@gmail.com"
```

2. Tener una cuenta de GitHub activa con los permisos necesarios para crear una rama.

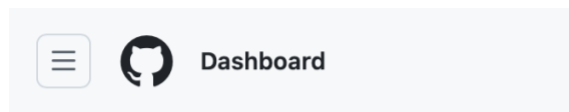
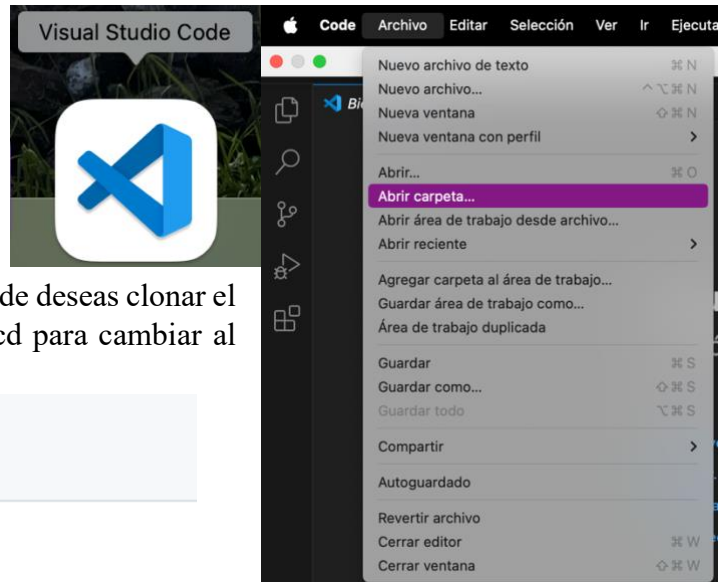
- Abre tu navegador favorito (ej. Safari).
- Busca en el navegador GitHub"



- Inicia Sesión en GitHub, si aún no tienes cuenta, crea una siguiendo los pasos mostrados la propia plataforma.
- Ingresa a un repositorio por invitación o crea uno y sube un archivo siguiendo los pasos que aparecen en la plataforma.

3. Repositorio clonado localmente.


- Abre Visual Studio Code.
- Abre la carpeta creada anteriormente, seleccionando Archivo → Abrir Carpeta.
- Navega hacia el directorio donde deseas clonar el repositorio. Usa el comando `cd` para cambiar al directorio deseado.



Top repositories

New

Find a repository...

 [DrLAlbertoAngelesH/InvestigacionDeOperaciones](https://github.com/DrLAlbertoAngelesH/InvestigacionDeOperaciones)

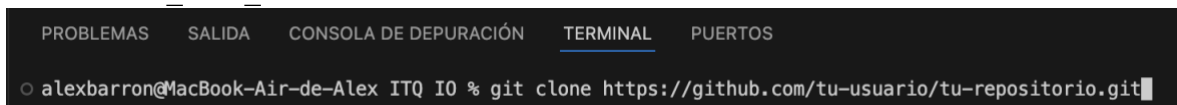
Ve a tu repositorio en GitHub y copia la URL del repositorio.

HTTPS SSH GitHub CLI

Copy url to clipboard

`https://github.com/DrLAlbertoAngelesH/Inv`

- En la terminal de Visual Studio Code ejecuta el siguiente código: `git clone URL_DEL_REPOSITORIO`.



4. Cambiar al directorio del repositorio clonado con el siguiente código: `cd tu-repositorio`.



5. Verifica el branch actual (`git branch`) y asegurarse de estar en la rama principal (main), con el siguiente código: `git branch`.



6. Actualiza el branch principal con los últimos cambios del repositorio remoto (`git pull origin main`).



7. Crea una nueva Branch con un nombre descriptivo (`git branch NombreDelBranch`).

```
PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS
alexbarron@MacBook-Air-de-Alex ITQ IO % git branch NombreDelBranch
```

8. Cambiar a la nueva branch creada (git checkout NombreDelBranch).

```
PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS
alexbarron@MacBook-Air-de-Alex ITQ IO % git checkout NombreDelBranch
```

9. Verificar que la nueva branch está activa (git branch)

```
PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS
alexbarron@MacBook-Air-de-Alex ITQ IO % git branch
```

10. Realizar los cambios necesarios en el código o archivos del repositorio.

11. Añadir los cambios realizados al área de preparación (git add .)

```
PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS
alexbarron@MacBook-Air-de-Alex ITQ IO % git add .NombreDelArchivo
```

12. Realizar un commit con un mensaje descriptivo de los cambios (git commit -m "Descripción de cambios").

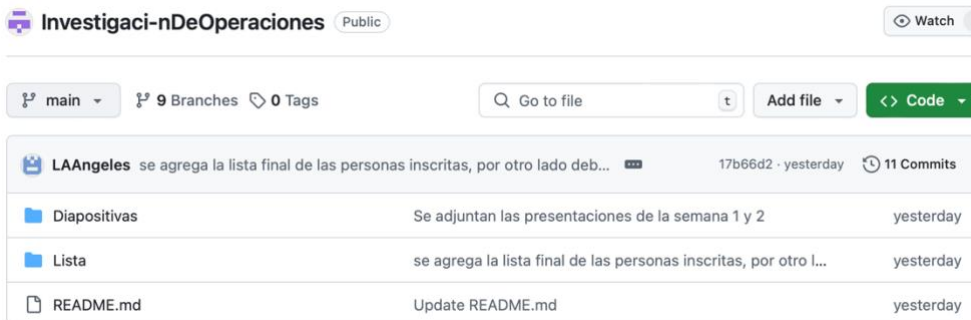
```
PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS
alexbarron@MacBook-Air-de-Alex ITQ IO % git commit -m "Descripción de cambios"
```

13. Enviar el nuevo branch al repositorio remoto (git push u- origin NombreDelBranch).

```
PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS
alexbarron@MacBook-Air-de-Alex ITQ IO % git push origin NombreDelBranch
```

14. Verificar en GitHub que el nuevo branch hayacreado correctamente.

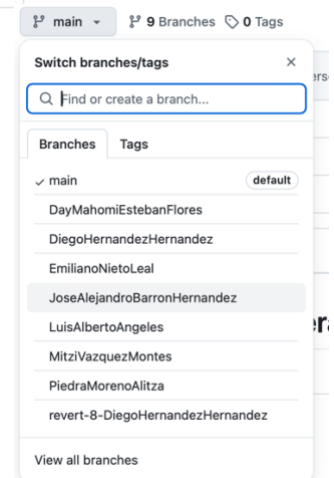
- Abre nuevamente tu navegador.
- Dirigite a la página de tu repositorio.



- Haz click en el menú desplegable.
- Busca NombreDelBranch en la lista de ramas (ej. JoseAlejandroBarronHernandez).

JoseAlejandroBarronHernandez

Listo! Haz creado un nuevo Branch exitosamente.



Control de Cambios.

N.º de Revisión	Fecha de Revisión.	Descripción del Cambio	Revisado por	Aprobado por
0	05/09/24	Creación del documento inicial.	Barrón Hernández	
1				

Documentos Relacionados.

- SOP-GIT-001: “Clonar un repositorio en GitHub”
- Documentación de Git <https://git-scm.com>.

Distribución.

Este SOP se distribuirá electrónicamente a todos los miembros del grupo de compañeros, estudiantes, profesores y cualquier personal interesado que requiera al acceso de clonación de repositorios en GitHub.

Checklist.

N.º	Paso	Completado
	Tener máquina local.	
	Tener conexión a algún servicio de internet.	
	Tener instalado Visual Studio Code.	
1	Verificar que Git esté instalado en la máquina local (git --version)	
2	Tener una cuenta de GitHub activa con los permisos necesarios para crear una rama.	
3	Repositorio clonado localmente.	
4	Cambiar al directorio del repositorio clonado (cd/ruta/a/tu/repositorio)	
5	Verificar el branch actual (git branch) y asegurarse de estar en la rama principal (main).	
6	Actualizar el branch principal con los últimos cambios del repositorio remoto (git pull origin main)	
7	Crear una nueva branch con un nombre descriptivo (git branch NombreDelBranch)	
8	Cambiar a la nueva branch creada (git checkout NombreDelBranch)	
9	Verificar que la nueva branch está activa (git branch)	
10	Realizar los cambios necesarios en el código o archivos del repositorio.	
11	Añadir los cambios realizados al área de preparación (git add .)	
12	Realizar un commit con un mensaje descriptivo de los cambios (git commit -m "Descripción de cambios")	
13	Enviar el nuevo branch al repositorio remoto (git push u- origin NombreDelBranch)	
14	Verificar en GitHub que el nuevo branch haya creado correctamente.	