

Pre-requisitos

1. Este taller se debe realizar en **parejas**.
2. Para esta sesión va a requerir una cuenta de **Github**. Si no la ha creado, es el momento de hacerlo en <https://github.com>. Note que Github requiere autenticación de doble factor (2FA), actívela usando por ejemplo la app de Github o de una aplicación de autenticación.
3. Descargue el **cliente de Git** para su sistema operativo en <https://git-scm.com/downloads>.
4. Instale el cliente. **Atención:** deje todas las configuraciones por defecto, excepto el editor, para el cual se recomienda seleccionar VSCode.
5. La entrega de este taller consiste en un **reporte** por grupo y unos **archivos de soporte**.
6. Cree el archivo de su **reporte** como un documento de texto en el que pueda fácilmente incorporar capturas de pantalla, textos y similares. Puede ser un archivo de word, libre office, markdown, entre otros.
7. En su **reporte** indique claramente el número de la sección y el numeral que está respondiendo. Se deben incluir las evidencias de los dos miembros del equipo.
8. Como **archivos de soporte** incluya los archivos agregados a los repositorios en su versión final.

1. Cree un repositorio Git local

1. Cree una carpeta local con el nombre taller2-primernombre, e.g., taller2-juan.
2. En VSCode vaya a File, Open Folder y seleccione la carpeta creada
3. En el menú de la izquierda seleccione Source Control.
4. Click en Initialize repository para iniciar el repositorio local
5. Verifique que la carpeta local contenga una carpeta (oculta) `.git`. Tome un pantallazo e inclúyalo en su **reporte**.
6. Incluya un archivo de python (`.py`) a su carpeta. Puede usar un archivo de un taller pasado.
7. Agregue el archivo al control de versiones (staging) usando el símbolo `+`.
8. Incluya un mensaje para el commit y realice el commit a la rama main.
9. En la terminal use el comando `git log` para mostrar el estado actual de su repositorio con el commit inicial. Tome un pantallazo e inclúyalo en su **reporte**.

10. Introduzca cambios a su archivo, modifique al menos una línea de código y agregue una línea más. Guarde (CTRL+s).
11. En la vista de control de versiones, note que el cambio queda marcado. Agregue el cambio (staging), incluya un comentario que describa el cambio realizado y envíe el commit.
12. Use nuevamente el comando `git log` para mostrar el estado actualizado de su repositorio con los dos commits. Tome un pantallazo e inclúyalo en su **reporte**.

2. Ramas en su repositorio Git local

1. En VSCode, vista de control de versiones, use la opción Branch → Create Branch para crear una nueva rama. Use su primer nombre como parte del nombre de la rama, e.g., `juan-viz`.
2. Use nuevamente el comando `git log` para mostrar el estado actualizado de su repositorio con la nueva rama. Tome un pantallazo e inclúyalo en su **reporte**.
3. Realice un cambio en el archivo.
4. Agregue el cambio al control de versiones (staging) y realice el commit.
5. Use nuevamente el comando `git log` para mostrar el estado actualizado de su repositorio con la nueva rama y la modificación al archivo. Tome un pantallazo e inclúyalo en su **reporte**.
6. Vuelva ahora a la rama main: use la opción Checkout para hacer el checkout de la rama main.
7. Use nuevamente el comando `git log` para mostrar el estado actualizado de su repositorio en la rama main. Tome un pantallazo e inclúyalo en su **reporte**.
8. Incorpore la otra rama a la rama main usando la opción Branch → Merge Branch.
9. Use nuevamente el comando `git log` para mostrar el estado actualizado de su repositorio en la rama main, donde se deben observar los cambios realizado en la rama auxiliar y el HEAD debe apuntar a las dos ramas (main y auxiliar). Tome un pantallazo e inclúyalo en su **reporte**.

3. Publique su repositorio en Github

1. En VSCode, vista de control de versiones, asegúrese de estar en el repositorio creado anteriormente, rama main.
2. Use el botón Publish Branch para publicar la rama main. Aquí debe autorizar a VSCode para enlazarse con su cuenta de GitHub. Para este taller, asegúrese de publicar en un repositorio **público**.

3. En su **reporte** incluya el enlace a su repositorio en github, y asegúrese de que es accesible públicamente.
4. Use el comando `git log` para mostrar el estado actualizado de su repositorio, donde se debe observar ahora también la rama origin/main asociada al repositorio remoto. Tome un pantallazo e inclúyalo en su **reporte**.
5. Realice ahora un nuevo cambio sobre el archivo, localmente, usado VSCode.
6. Agregue el cambio al control de versiones (staging) y realice el commit. En este caso realice una sincronización (Sync Changes) que ejecuta un push y un pull a la ubicación remota.
7. Use el comando `git log` para mostrar el estado actualizado de su repositorio, donde se debe observar ahora también el cambio realizado sobre la rama origin/main asociada al repositorio remoto. Tome un pantallazo e inclúyalo en su **reporte**.
8. Use el comando `git log` para mostrar el estado actualizado de su repositorio, donde se debe observar ahora también la rama origin/main asociada al repositorio remoto. Tome un pantallazo e inclúyalo en su **reporte**.
9. Asegúrese de que en Github su repositorio incluya el cambio realizado.

4. Colaborar a través de repositorios

1. En GitHub incluya como colaboradores a su compañero/a de grupo. En su **reporte** incluya un pantallazo de la página de Colaboradores con su compañero/a incluido/a.
2. Permita acceso restringido a través de pull requests a su rama principal (master/-main).
3. Clone el repositorio de su compañero/a localmente.
4. Sobre este nuevo repositorio, cree (Create Branch) y haga Checkout de una rama.
5. Sobre esta rama realice modificaciones (visibles y que tengan sentido) al archivo de su compañero.
6. Publique la rama en el repositorio remoto.
7. Use el comando `git log` para mostrar el estado actual del repositorio, donde se debe observar la rama sobre la que está trabajando. Tome un pantallazo e inclúyalo en su **reporte**.
8. En Github cree un Pull Request (PR) en la que solicite que se agregue su rama a la rama main. Tome un pantallazo e inclúyalo en su **reporte**.
9. En Github, vaya ahora a su repositorio y verifique que su compañero/a haya solicitado el PR.

10. Verifique los cambios y acéptelos. Revise que los cambios sean visible en Github.
11. Vuelva a su repositorio localmente y realice un pull.
12. Use el comando `git log` para mostrar el estado actual del repositorio, donde se debe observar la rama main actualizada con los cambios realizados por su compañero/a en la rama incluida a través del PR. Tome un pantallazo e inclúyalo en su **reporte**.
13. No olvide incluir la URL de su repositorio (público) en Github en su **reporte**.