# Pre-requisitos

1. Este taller se debe realizar en **parejas**.
2. Para esta sesi´on va a requerir una cuenta de **Github**. Si no la ha creado, es el momento de hacerlo en [https://github.com](https://github.com/). Note que Github requiere autentica- ci´on de doble factor (2FA), act´ıvela usando por ejemplo la app de Github o de una aplicaci´on de autenticaci´on.
3. Descargue el **cliente de Git** para su sistema operativo en

<https://git-scm.com/downloads>.

1. Instale el cliente. **Atenci´on**: deje todas las configuraciones por defecto, excepto el editor, para el cual se recomienda seleccionar VSCode.
2. La entrega de este taller consiste en un **reporte** por grupo y unos **archivos de soporte**.
3. Cree el archivo de su **reporte** como un documento de texto en el que pueda f´acil- mente incorporar capturas de pantalla, textos y similares. Puede ser un archivo de word, libre office, markdown, entre otros.
4. En su **reporte** indique claramente el nu´mero de la secci´on y el numeral que est´a respondiendo. Se deben incluir las evidencias de los dos miembros del equipo.
5. Como **archivos de soporte** incluya los archivos agregados a los repositorios en su versi´on final.

# 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 menu´ de la izquierda seleccion 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 inclu´yalo 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 s´ımbolo +.
  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 reposi- torio con el commit inicial. Tome un pantallazo e inclu´yalo en su **reporte**.
  10. Introduzca cambios a su archivo, modifique al menos una l´ınea de c´odigo y agregue una l´ınea m´as. 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 env´ıe 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 inclu´yalo en su **reporte**.

# Ramas en su repositorio Git local

* 1. En VSCode, vista de control de versiones, use la opci´on 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 inclu´yalo 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 modificaci´on al archivo. Tome un pantallazo e inclu´yalo en su **reporte**.
  6. Vuelva ahora a la rama main: use la opci´on 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 inclu´yalo en su **reporte**.
  8. Incorpore la otra rama a la rama main usando la opci´on 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 inclu´yalo en su **reporte**.

# Publique su repositorio en Github

* 1. En VSCode, vista de control de versiones, asegu´rese de estar en el repositorio creado anteriormente, rama main.
  2. Use el bot´on Publish Branch para publicar la rama main. Aqu´ı debe autorizar a VSCode para enlazarse con su cuenta de GitHub. Para este taller, asegu´rese de publicar en un repositorio **pu´blico**.
  3. En su **reporte** incluya el enlace a su repositorio en github, y asegu´rese de que es accesible pu´blicamente.
  4. Use el comando git log para mostrar el estado actualizado de su repositorio, donde se debe observar ahora tambi´en la rama origin/main asociada al repositorio remoto. Tome un pantallazo e inclu´yalo 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 sincronizaci´on (Sync Changes) que ejecuta un push y un pull a la ubicaci´on remota.
  7. Use el comando git log para mostrar el estado actualizado de su repositorio, don- de se debe observar ahora tambi´en el cambio realizado sobre la rama origin/main asociada al repositorio remoto. Tome un pantallazo e inclu´yalo en su **reporte**.
  8. Use el comando git log para mostrar el estado actualizado de su repositorio, donde se debe observar ahora tambi´en la rama origin/main asociada al repositorio remoto. Tome un pantallazo e inclu´yalo en su **reporte**.
  9. Asegu´rese de que en Github su repositorio incluya el cambio realizado.

# Colaborar a trav´es de repositorios

* 1. En GitHub incluya como colaboradores a su compan˜ero/a de grupo. En su **reporte**

incluya un pantallazo de la p´agina de Colaboradores con su compan˜ero/a incluido/a.

* 1. Permita acceso restringido a trav´es de pull requests a su rama principal (master/- main).
  2. Clone el repositorio de su compan˜ero/a localmente.
  3. Sobre este nuevo repositorio, cree (Create Branch) y haga Checkout de una rama.
  4. Sobre esta rama realice modificaciones (visibles y que tengan sentido) al archivo de su compan˜ero.
  5. Publique la rama en el repositorio remoto.
  6. Use el comando git log para mostrar el estado actual del repositorio, donde se debe observar la rama sobre la que est´a trabajando. Tome un pantallazo e inclu´yalo en su **reporte**.
  7. En Github cree un Pull Request (PR) en la que solicite que se agregue su rama a la rama main. Tome un pantallazo e inclu´yalo en su **reporte**.
  8. En Github, vaya ahora a su repositorio y verifique que su compan˜ero/a haya solici- tado el PR.
  9. Verifique los cambios y ac´eptelos. Revise que los cambios sean visible en Github.
  10. Vuelva a su repositorio localmente y realice un pull.
  11. 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 compan˜ero/a en la rama incluida a trav´es del PR. Tome un pantallazo e inclu´yalo en su **reporte**.
  12. No olvide incluir la URL de su repositorio (pu´blico) en Github en su **reporte**.