Taller Git

# Objetivos:

* Modificar el proyecto colaborativamente utilizando Git.
* Resolver conflictos de integración de código utilizando diferentes estrategias.

# Antecedentes

El sistema de control de versiones GIT permite mantener un correcto manejo y control sobre los cambios realizados sobre el código fuente, pero además permite trabajar en el desarrollo de un proyecto de software de forma colaborativa y asíncrona. Esto puede provocar que existan conflictos entre los cambios realizados por diferentes miembros del equipo de desarrollo, especialmente cuando los cambios son realizados sobre la misma porción de código.

En este taller vamos a distribuir el trabajo entre 3 o 5 personas (grupos creados aleatoriamente), de forma que realicen cambios sobre los mismos archivos y se produzcan estos conflictos para luego aprender cómo solucionar los mismos. Para esto cada integrante debe crear una cuenta personal en la plataforma de Github, es muy importante que coloquen su nombre correcto, ya que esta cuenta servirá para identificar los cambios realizados por cada uno.

## Pasos previos

Elegir un líder de grupo para crear el repositorio remoto (proyecto en Github) y que todos puedan trabajar.

Cada estudiante debe instalar Git en su computadora y un IDE para JAVA (Eclipse o Netbeans son los recomendados).

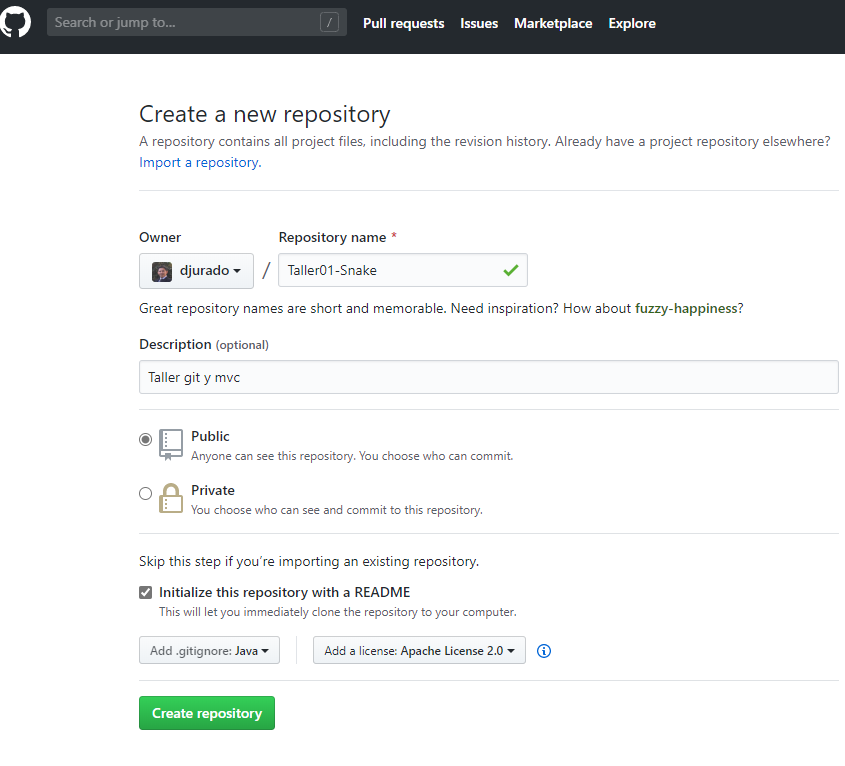
## Enlaces

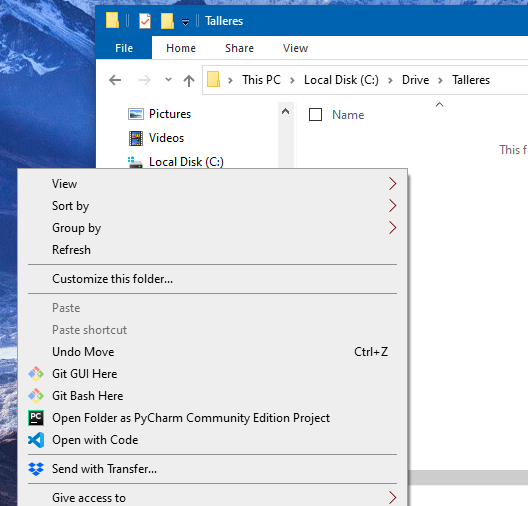
* Github: <https://github.com/>
* Git for Windows: <https://gitforwindows.org/>
* Git for MacOS: $ brew install git
* Git for Linux: sudo apt-get install git
* Eclipse: <https://www.eclipse.org/downloads/>

## Comandos importantes

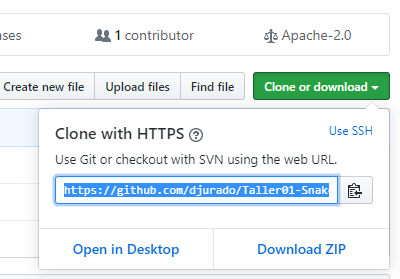
* Conocer el estado de su repositorio local: **git status**
* Clonar localmente un repositorio remoto: **git clone <repositorio\_remoto>**
* Agregar todos los cambios realizados al stage: **git add .**
* Guardar todos los cambios agregados al stage: **git commit -m “Comentarios”**
* Subir al repositorio remoto todos los commit sin enviar: **git push origin master**
* Descargar los nuevos cambios desde el repositorio remoto: **git pull**
* Si hay un usuario grabado en su computadora y desea utilizar otro: **git config --local credential.helper ""**

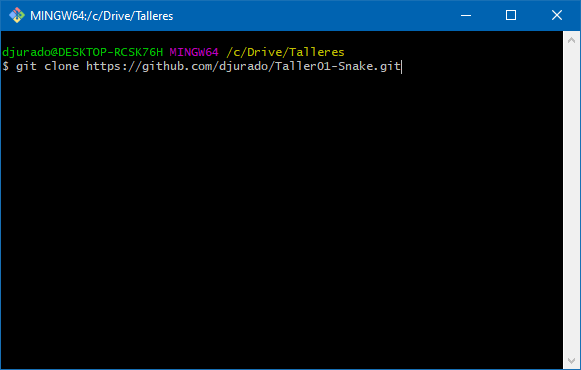
# Parte 1. Configuración del taller

1. El líder debe crear un nuevo repositorio público con el nombre “Taller01-Snake”. 
2. El líder debe agregar a los integrantes del grupo al repositorio. Dentro del repositorio ir a Settings > Manage access > Invite a collaborator y colocar el usuario de cada uno de los integrantes del grupo.
3. El líder debe clonar el repositorio en un directorio de su computadora.
   1. Para esto debe seleccionar un directorio en donde se guardarán los archivos del proyecto y con el ratón dar clic derecho y seleccionar Git Bash. Esto permitirá abrir la consola para ejecutar los comandos de GIT.

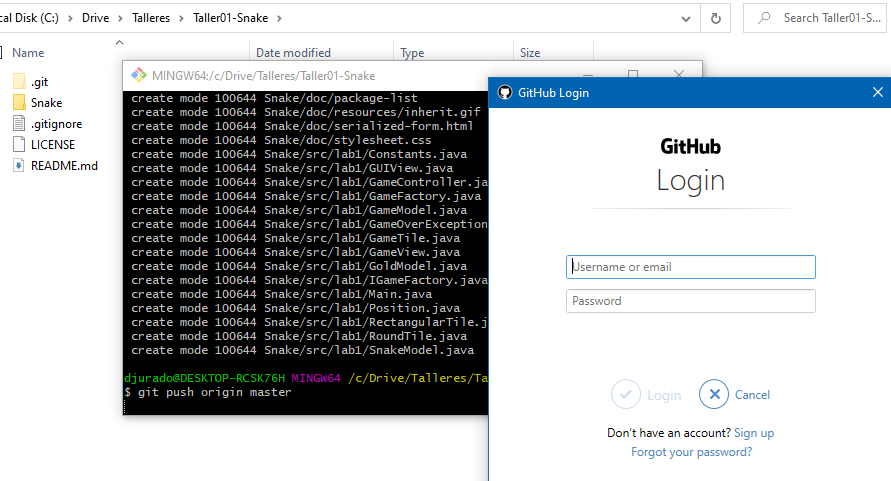


* 1. Copie el enlace correspondiente para clonar el repositorio desde la interfaz web en Github.



* 1. En la consola ingrese el comando “git clone ” y pegue el enlace de su repositorio remoto. 

1. El líder debe agregar el código fuente inicial al repositorio remoto.
   1. Descomprimir los archivos del proyecto dentro de la carpeta “Taller01-Snake”.
   2. En la consola abrir la carpeta con el comando “cd Taller01-Snake”
   3. Luego debe añadir los nuevos archivos al stage con el comando “git add .” y guardar los cambios realizados con el comando: git commit -m “Agregando código base”
   4. Finalmente, debe subir los cambios al repositorio remoto con “git push origin master” y colocar sus credenciales de github.



1. Cada integrante debe abrir su correo y aceptar la invitación al repositorio y proceder **a clonarlo en su computadora local (literal 3 de esta parte).**
2. **Cada integrante para subir sus cambios deberá seguir los pasos indicados en el literal 4, exceptuando el literal d, ya que en este los integrantes deberán usar únicamente “git push”.**

# Parte 2. Solucionando conflictos

Cada integrante debe clonar en su computadora el repositorio remoto y realizar los cambios indicados para cada uno, el líder será quien ayude y verifique que todos los miembros generen y solucionen un conflicto entre la versión individual y la versión remota actualizada.

Nota1: Cada integrante debe resolver un conflicto de integración con ayuda del líder de grupo.

Nota2: Tengan en cuenta que cada integrante debe realizar los cambios que le corresponde, pero al finalizar deben subir en el orden (Líder, integrante 1, integrante 2, integrante 3, integrante 4) sin actualizar a los cambios hechos por los otros. Esto es para tratar de generar un conflicto de integración de código con cada integrante.

1. **Líder**:
   1. Ventana principal:
      1. Cambiar el texto del botón de “Start Game” a “Jugar”.
   2. Subir Cambios al repositorio remoto.
2. **Integrante1**:
   1. Ventana principal:
      1. Cambiar el texto del botón de “Start Game” a “Let’s Go!!!”.
   2. Juego Gold:
      1. Por cada ficha recolectada, asigne 2 puntos en lugar de 1.
   3. Subir Cambios al repositorio remoto.
3. **Integrante2**
   1. Ventana principal:
      1. Cambiar el texto del botón “Start Game” a “Let’s Play”
   2. Juego Snake:
      1. Para el juego ‘Snake’ cambie el color de la serpiente a café.
   3. Subir Cambios al repositorio remoto.
4. **Integrante3 ( Si existe )**:
   1. Ventana principal:
      1. Cambiar el texto del botón “Start Game” a “Empezar”
      2. Cambiar el color de fondo de la pantalla principal de blanco a gris.
   2. Juego Gold:
      1. Por cada ficha recolectada, asigne 3 puntos en lugar de 1.
   3. Subir Cambios al repositorio remoto.
5. **Integrante4 ( Si existe )**:
   1. Juego Gold:
      1. Cambie el color de relleno de las fichas a azul.
      2. Cambiar el texto del botón “Start Game” a “Go”
   2. Juego Snake:
      1. Para el juego ‘Snake’ cambie el color de la serpiente a negro.
   3. Subir Cambios al repositorio remoto.

# Rúbrica

|  |  |  |
| --- | --- | --- |
| Criterio | Descripción | Ponderación |
| **Creación de repositorio** | El grupo creo correctamente el repositorio con los archivos del taller y todos los miembros fueron agregados correctamente. | 35% |
| **Solución de conflictos** | Cada uno de los integrantes pudo generar y resolver un conflicto entre versiones. | 65% |