Taller Git

# Objetivos:

* Modificar el proyecto colaborativamente utilizando Git.
* Crear ramas de Git para manejar desarrollo en paralelo.

# 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 4 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 tener una cuenta 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 tener instalado **Git** y **GitHub Desktop** en su computadora y un IDE para JAVA (Eclipse o Netbeans son los recomendados).

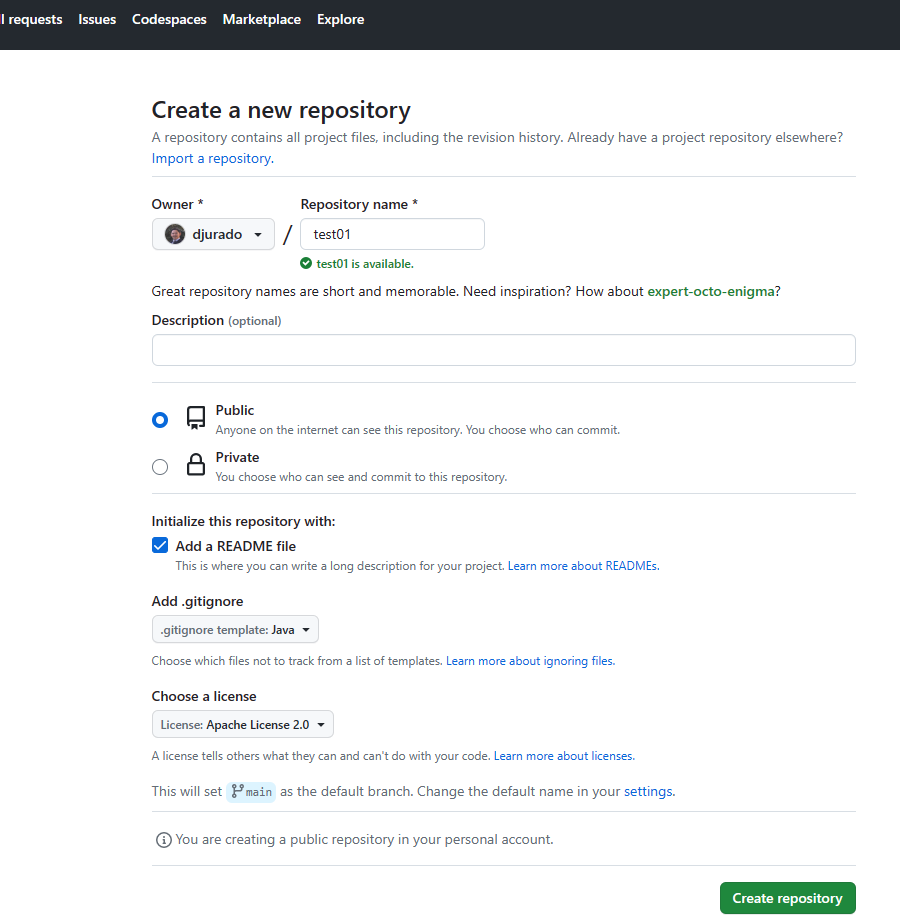
## Enlaces

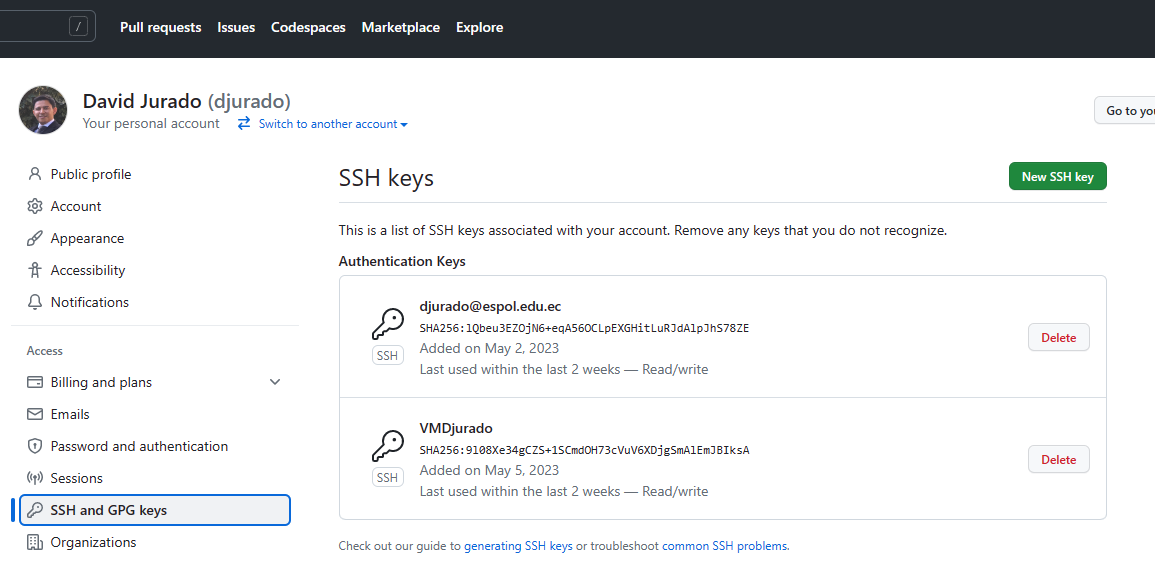
* Github: <https://github.com/>
* Git for Windows: <https://gitforwindows.org/>
* Git for MacOS: $ brew install git
* Git for Linux: sudo apt-get install git
* Eclipse: <https://www.eclipse.org/downloads/>

## 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 main**
* Descargar los nuevos cambios desde el repositorio remoto: **git pull origin main**
* Crear una nueva rama a partir de la rama actual: **git branch <nueva\_rama>**
* Crear una nueva rama a partir de los cambios actuales: **git stash** y luego **git stash branch <nueva\_rama>**
* Cambiarse a otra rama: **git checkout <rama>**
* Muestra todas las ramas: **git branch**
* Unir ramas:
  + git checkout <rama\_destino>
  + git merge <rama\_con\_cambios>
* Subir al repositorio remoto todos los commit de la rama: **git push origin <rama>**
* Comando para ver en consola las ramas en forma gráfica: **git log –all –graph –decorate --oneline**
* Si hay un usuario grabado en su computadora y desea utilizar otro: **git config --local credential.helper ""**

# Parte 1. Configuración del taller

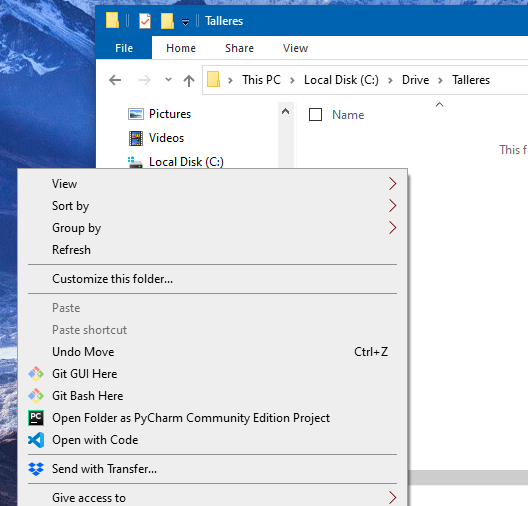
1. El líder debe crear un nuevo repositorio público con el nombre “Taller02-Ramas”. 
2. El líder debe agregar a los integrantes del grupo al repositorio. Dentro del repositorio ir a Settings > Collaborators > Add people y colocar el usuario de cada uno de los integrantes del grupo.
   1. Cada integrante debe configurar su clave pública y privada SSH y copiar la clave pública (.pub) a su cuenta en GitHub desde Settings > SSH and GPG keys > New SSH key.



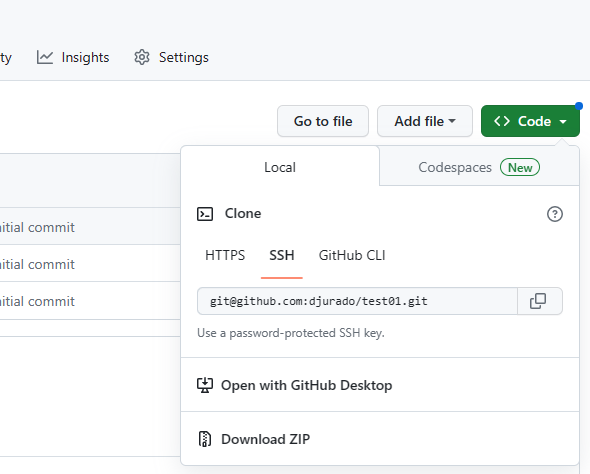
Para realizar el proceso anterior pueden seguir las instrucciones del siguiente enlace:

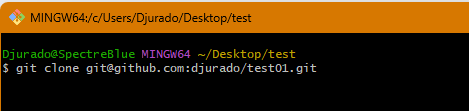
<https://docs.github.com/en/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent>

* 1. El líder debe clonar el repositorio en un directorio de su computadora. 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 SSH para clonar el repositorio desde la interfaz web de GitHub.



* 1. En la consola ingrese el comando “git clone ” y pegue el enlace de su repositorio remoto. 
  2. Descomprimir el código fuente de Taller02-TopMusical.zip en la carpeta del repositorio local.
  3. En la consola abrir la carpeta con el comando “cd TopMusical”
  4. Luego debe añadir los nuevos archivos al stage y crear el commit con el comando: git commit -am “Agregando código base”
  5. Finalmente, debe subir los cambios al repositorio remoto con “git push origin main”.
  6. Hay que confirmar que se subieron los cambios en GitHub.
  7. Todos los integrantes deben clonar en sus computadoras el repositorio con el código sin modificación alguna. Solo deberían haber 2 commits.

# Parte 2. Creando Ramas

A continuación, vamos a crea una rama de desarrollo por cada integrante, incluido el líder, y luego debe realizar el cambio correspondiente, guardarlo en su rama y subir la rama con los cambios al repositorio remoto.

Para el siguiente paso, todos deben trabajar en paralelo y subir sus cambios en la rama correspondiente al repositorio remoto.

1. **Líder**:
   1. Crear una nueva rama con nombre ‘titulo’.
   2. Elija un nuevo nombre para mostrar como título del Top 10 de canciones.
   3. Agregar al readme una pequeña captura del resultado.
   4. Crear commit con un mensaje adecuado.
   5. Subir su rama con los cambios al repositorio remoto.
2. **Integrante1**:
   1. Crear una nueva rama con nombre ‘orden’.
   2. Cambie el orden para mostrar el Top 10 de canciones, ahora debe estar en forma descendente.
   3. Agregar al readme una pequeña captura del resultado.
   4. Crear commit con un mensaje adecuado.
   5. Subir su rama con los cambios al repositorio remoto.
3. **Integrante2**
   1. Crear una nueva rama con nombre ‘artista’.
   2. Coloque el nombre del artista o grupo antes del nombre de la canción.
   3. Agregar al readme una pequeña captura del resultado.
   4. Crear commit con un mensaje adecuado.
   5. Subir su rama con los cambios al repositorio remoto.
4. **Integrante3 ( Si existe )**:
   1. Crear una nueva rama con nombre ‘numero’.
   2. Coloque el número de la posición de la canción al lado derecho del texto
   3. Agregar al readme una pequeña captura del resultado.
   4. Crear commit con un mensaje adecuado.
   5. Subir su rama con los cambios al repositorio remoto.
5. **Integrante4 ( Si existe )**:
   1. Crear una nueva rama con nombre ‘info’.
   2. Mostrar el nombre del artista en la sección derecha que aparece al hacer clic sobre una canción.
   3. Agregar al readme una pequeña captura del resultado.
   4. Crear commit con un mensaje adecuado.
   5. Subir su rama con los cambios al repositorio remoto.

# Parte 3. Fusionando ramas

Para esta parte todos deben fusionar su rama y corregir los conflictos que surjan en el proceso.

**Cada integrante**:

1. Cambiarse a la rama main.
2. Descargarse los cambios del repositorio remoto.
3. Traer la rama main los cambios de la rama que ud mismo creó anteriormente.
4. Si aparece algún conflicto debe solucionarlo manteniendo ambos cambios.
5. Subir los cambios realizados al repositorio remoto.