Java / JDK / JRE / JVM

1. Git status : validar si existe el repositorio git
2. Git init: Crea el repositorio git
3. Se crea un archivo .getignore y se le agregan las rutas de lo que no queremos incluir en el git
4. Los commits son paquetes de cambios al SF, se aconseja realizarlos constantemente para identificar fácilmente errores.
5. Antes del comitt se deben cargar los archivos que voy a versionar (git add)
6. Git commit -m (adicionar un comentario al commit)

En caso de configurar cuentas por primer commit:

PS C:\Users\PM Andrés Henao\IdeaProjects\Primero> git config --global user.email "user.email andres\_henao@elpoli.edu.co"

PS C:\Users\PM Andrés Henao\IdeaProjects\Primero> git config --global user.name "Andres Henao"

PS C:\Users\PM Andrés Henao\IdeaProjects\Primero> git commit -m "prueba de comentario"

Git Branch: ver lista de ramas

Para crear una nueva: git Branch main

Para cambiar entre ramas: git checkout -b main

Cuando haga cambios, valido con git status si lo toma, y para agregarlo git add + ruta

Finalmente, el git commit -m para el comentario

Para traer cambios de una rama a otra:

Git merge main

Para subir

git push origin main

Creaer nuevo package .json

npm init

**…or create a new repository on the command line**

echo "# Primero" >> README.md

git init

git add README.md

git commit -m "first commit"

git branch -M main

git remote add origin https://github.com/ces3-2023-Poli/Primero.git

git push -u origin main

**…or push an existing repository from the command line**

git remote add origin https://github.com/ces3-2023-Poli/Primero.git

git branch -M main

git push -u origin main

Cuando se cree un proyecto local y a su vez uno en nube, para poder dar permisos de edición sobre el proyecto local siguiente comando:

git pull origin main --allow-unrelated-histories