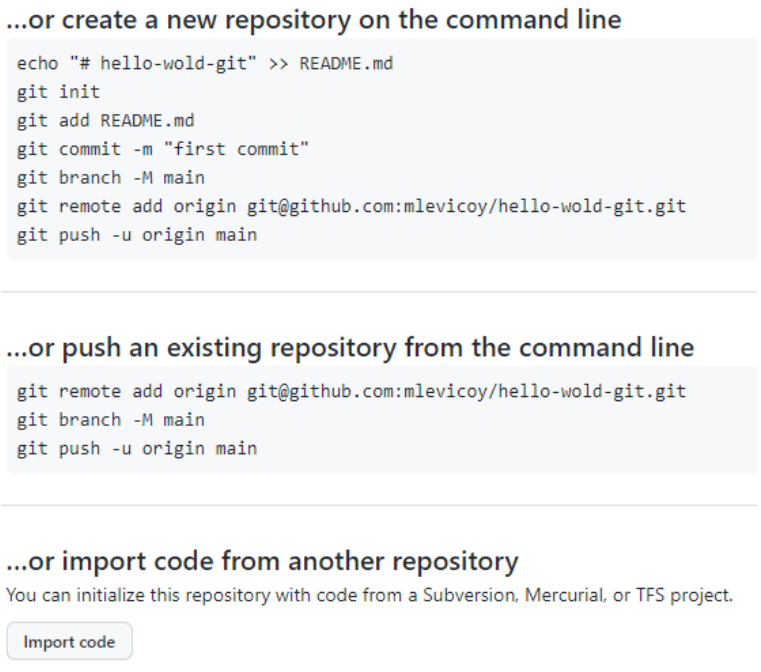
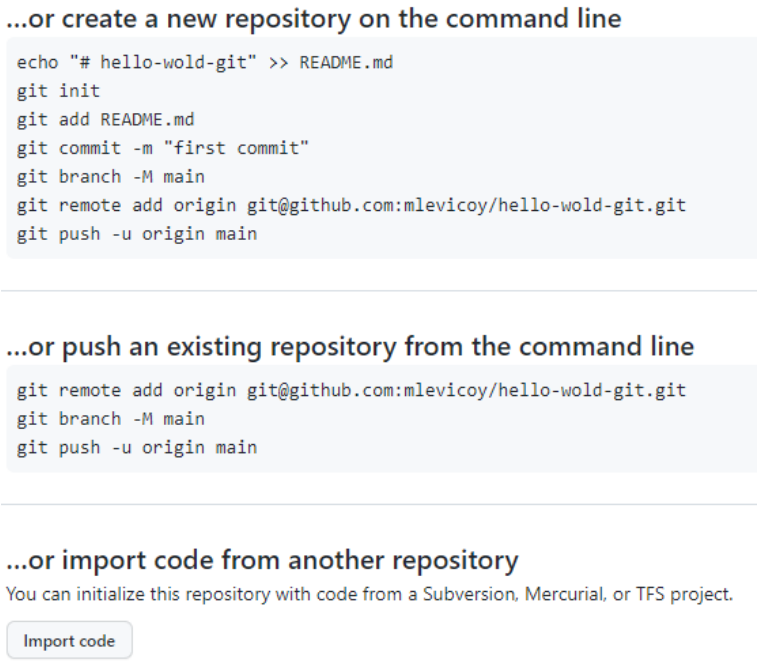
**GIT**:

1. Software de gestión de versiones.
2. Descargar GIT desde https://git-scm.com/
3. Inicializar repositorio con **git init**.
4. Al iniciar el repositorio aparecerá una carpeta **.git**, que manejara todo el CORE de los repositorios.
5. Con **git status** se puede ver el resumen de los cambios.
6. **On branch master**, quiere decir que estamos en la rama master. Puede haber múltiples ramas.
7. **Commits** es una confirmación de cambios.
8. **Untracked files** son los archivos que aún no están incluidos en el commits.
9. Con **git add <file>** incluimos el archivo en el tracking para el commits.
10. Si se quiere eliminar ese archivo de tracking se usa, **git rm --cached <file>**.
11. Para el commit se hace, **git commit -m "<mensaje>"**.
12. **Push** es para actualizar los cambios.
13. Para incluir todos los archivos del repositorio se hace, **git add --all**.
14. **Bitbucket**, repositorio similar a GitHub pero privado.
15. Al crear repositorio (hello-wold-git) en GitHub aparecerá:



16. Como ejemplo con GitHub:

    a. Luego de inicializar un contenedor local y hacer el submmit de todo lo necesario,
    b. Generamos el contenedor en GitHub, ej. Hellow-word-git, nos mostrar el mensaje:




    c. Conectamos el GitKraken a nuestro GitHub.
    d. Por consola agregamos el origen remoto del repositorio con **git remote add origin git@github.com:mlevicoy/hello-wold-git.git**
    e. Para verificar donde está apuntando el origen **git remote -v**
    f. Para eliminar esta conexión remota se hace **git remote remove origin**
    g. En el GitKraken crear SSH Key dentro de GitHub para poder conectarse correctamente.
    h. Se creará la rama master solo 1 vez al pulsar Push.

17. Nunca se debe trabajar con la rama Master, en este caso usamos **Develop**, Se creará en local y se hará un **push** para que se cree en el GitHub.
18. Para crear una nueva rama se usa **git branch <nombre_rama>**.
19. Para verificar la rama **git checkout <nombre_rama>**.
20. En GitKraken para una nueva rama, botón derecho sobre master y elegir "**Create Branch here**". Luego Push para que se actualice el GitHub.
21. Cuando se finaliza de trabajar en develop, por ejemplo, y se quiere pasar todo a master. En GitKraken se hace botón derecho sobre master y se elige "**Mege develop into master**", se generará un nuevo commit, luego doble clic en master y Push.

**22.** Para clonar un repositorio, por consola, se hace:

- **Git clone <URL>**

**23.** Cuando se trabaja en colaborativo, si se quiere mantener los repositorios actualizados es necesario hacer un **Pull**. Por consola **git pull**, dentro del repositorio.

**24.** Para ignorar un archivo o carpeta, crear archivo ".**gitignore**", y colocar las extensiones o carpetas.

**25.** Para verificar lo ignorado "**git status --ignored**".

**26.** Para forzar a añadir algo, "**git add -f <archivo>**".

**27.** Para crear archivos temporales se usa "**stage**".

**28.** Los **tags** marcan las versiones. Ejemplo:

- **Git tag v0.0.1 -m "Primera versión"**
- **Git tag,** para ver el estado de las versiones.
- **Git show v0.0.2,** para ver el estado del repositorio en la versión de ejemplo.
- **Git push --tags**, para enviar los tags a GitHub.
- **Git push origin v0.0.4**, para enviar un tags en especifico.

**29. Git Flow**:

**a.** Ramas principales:

| | | |
|---|---|---|
| **Master (origin/master)** | : | Estado listo para producción |
| **Develop (origin/develop)** | : | Contiene los últimos cambios para la próxima versión del software. Llamado rama de integración. |

**b.** Ramas auxiliares:

| | | |
|---|---|---|
| **Ramas de funcionalidad (Feature branches)** | : | - Pueden partir de: develop.<br>- Debe fusionarse con: develop.<br>- Convención de nombre: feature-NUMissue-*. |
| **Ramas de versión (Release branches)** | : | - Pueden partir de: develop.<br>- Debe fusionarse con: develop y master.<br>- Convención de nombre: release-*. |
| **Ramas de parches (Hotfix branches)** | : | - Pueden partir de: master.<br>- Debe fusionarse con: develop y master.<br>- Convención de nombre: hotfix-*. |

**c.** Esta extensión hay que instalarla.

**d.** Uso, ejemplo:

i. Iniciar el espacio de trabajo con "**git flow init**".
ii. Comprobar las ramas con "**git branch**".
iii. Crear y trabajar con ramas de funcionalidades con "**git flow feature start feature_branch**".
iv. Dejar de usar la rama feature y hacer commit y finalizar con "**git flow feature stop feature_branch**". Esto actualizara develop.
v. Crear y trabajar con ramas de versiones con "**git flow release start 0.1.0**".
vi. Dejar de usar la rama reléase con "**git flow release finish '0.1.0'**". Esto integrara automáticamente con master y subirá a producción.
vii. Crear y trabajar con ramas hotfix, se usa para cambios rápidos que no puedan esperar la próxima integración con release, "**git flow hotfix start hotfix_branch**".
viii. Finalizar hotfix y fusionar con master y develop, "**git flow hotfix finish hotfix_branch**".

**e.** Usar git flow con GitKraken:

i. Ir a File > Preferences > GitFlow, y pulsar "**Inicialize GitFlow**".
ii. En la barra izquierda aparecerá GIT FLOW y un botón para inicializar la rama deseada.

**30. Markdow**:

- **a.** Cabecera: Se usa almohadilla (#), va de H1 a H6 y mientras más almohadilla disminuye la cabecera.
- **b.** Underline: Son alternativas a la cabecera.
- **c.** Énfasis: itálica, negrita y tachado.
- **d.** Listas: Ordenadas y/o desordenadas.
- **e.** Link: Enlaces.
- **f.** Imágenes.
- **g.** Code Snippets.
- **h.** Tablas.
- **i.** Citas.
- **j.** Líneas divisoras.
- **k.** Saltos de línea.

```
# Version del curso
v.1.2.2

# Cabecera H1
## Cabecera H2
### Cabecera H3
#### Cabecera H4
##### Cabecera H5
###### Cabecera H6

Underine 1
------------
Underline 2
============

## Enfasis
---------
- Formato *Italico* de la la primera forma.
- Formato _italico_ de la segunda forma.
- Formato **bold o strong** de la primera forma.
- Formato __bold o strong__ de la segunda forma.
- Formato ~~tachado~~, forma normal.
- Aqui podemos usar *formato italico*, pero tambien **bold** y ~~tachado~~.
```

```
## Listas
---
1. Item 1.
2. Item 2.
3. Item 3.
- Item 4
- Item 5
- Item 6.
* Item 7.
* Item 8.
* Item 9.

## Links
---
- <a href="https://www.google.cl">Esto es un link HTML</a>
- [Esto es un link en Markdown](https://www.google.cl)


## Imagenes
---
![Logo Github](https://qph.cf2.quoracdn.net/main-qimg-729a22aba98d1235fdce4883accaf81e)

## Code Snippets
```

## Code Snippets
---
```JSON
[
    {
        "title": "applet",
        "count": [12000, 20000],
        "description": {"text": "...", "sensitive": false}
    }
]
```

## Tablas
---
| Nombre | Apellido | Documento |
| ------ | -------- | --------- |
| Juan   | Perez    | 123       |
| Ana    | Godoy    | 345       |
| Tom    | Sawyer   | 567       |

## Citas
---
Esto es un texto referente a una cita que pondremos debaja:
> Esto es una cita.

Esto es otro texto que no es cita:
> Esto es otra cita.

## Líneas divisoras
---
Esto es un texto que será dividio por guiones medios.

---
Esto es otro texto dividido por asteriscos.

***
Esto es otro texto dividido por guines bajos.
___

## Saltos de línea
___
Esto es nuestro primer parrafo.

Esto es nuestro segundo parrafo.

Esto es nuestro tercer parrafo.