

The background features abstract green geometric shapes. On the left, a solid green trapezoid points upwards. On the right, a complex arrangement of overlapping translucent green triangles and polygons creates a layered, crystalline effect. The text is centered in the white space between these elements.

GIT - Repositorios remotos

Repositorios remotos en GIT

- ▶ Para colaborar en cualquier proyecto Git, necesitas saber cómo gestionar tus repositorios remotos.
- ▶ Son versiones de tu proyecto que se encuentran alojados en Internet o en algún punto de la red
- ▶ Puedes tener varios, cada uno de los cuales puede ser de sólo lectura, o de lectura/escritura, según los permisos que tengas
- ▶ Colaborar con otros implica gestionar estos repositorios remotos, y mandar (push) y recibir (pull) datos de ellos cuando necesites compartir cosas
- ▶ Gestionar repositorios remotos implica:
 - ▶ Conocer cómo añadir repositorios nuevos
 - ▶ Eliminar aquellos que ya no son válidos
 - ▶ Gestionar ramas remotas e indicar si están bajo seguimiento o no
 - ▶ Etc.

Repositorios remotos en GIT

MOSTRANDO SUS REPOSITORIOS REMOTOS

- ▶ Para ver qué repositorios remotos tenemos configurados, se puede ejecutar el comando “git remote”
- ▶ Mostrará una lista con los nombres de los remotos que se hayan especificado
- ▶ Si has clonado tu repositorio, deberías ver por lo menos "origin" (nombre predeterminado que le da Git al servidor del que clonaste)

```
$ git clone git://github.com/schacon/ticgit.git Initialized empty  
Git repository in /private/tmp/ticgit/.git/ remote: Counting  
objects: 595, done. remote: Compressing objects: 100% (269/269),  
done. remote: Total 595 (delta 255), reused 589 (delta 253)  
Receiving objects: 100% (595/595), 73.31 KiB | 1 KiB/s, done.  
Resolving deltas: 100% (255/255), done.  
$ cd ticgit  
$ git remote  
origin
```

Repositorios remotos en GIT

MOSTRANDO SUS REPOSITORIOS REMOTOS

- Se puede añadir la opción -v, que muestra la URL asociada a cada repositorio remoto:

```
$ git remote -v origin git://github.com/schacon/ticgit.git  
(fetch) origin git://github.com/schacon/ticgit.git (push)
```

- Si tienes más de un remoto, te lista todos los repositorios. Ejemplo:

```
$ cd grit
```

```
$ git remote -v
```

```
bakkdoor  git://github.com/bakkdoor/grit.git  
cho45     git://github.com/cho45/grit.git  
defunkt   git://github.com/defunkt/grit.git  
koke      git://github.com/koke/grit.git  
origin    git@github.com:mojombo/grit.git
```

Repositorios remotos en GIT

AÑADIENDO REPOSITORIOS REMOTOS

- Para añadir un nuevo repositorio Git remoto, asignándole un nombre con el que referenciarlo fácilmente, ejecuta “git remote add [nombre] [url]”:

```
$ git remote origin
```

```
$ git remote add pb git://github.com/paulboone/ticgit.git
```

```
$ git remote -v
```

```
origin git://github.com/schacon/ticgit.git pb  
git://github.com/paulboone/ticgit.git
```

- Ahora la cadena "pb" se puede usar en lugar de la URL

Repositorios remotos en GIT

AÑADIENDO REPOSITORIOS REMOTOS

- Ejemplo: si quieres recuperar cierta información que todavía no tienes en tu repositorio, puedes ejecutar:

```
$ git fetch pb
```

```
remote: Counting objects: 58, done.
```

```
remote: Compressing objects: 100% (41/41), done.
```

```
remote: Total 44 (delta 24), reused 1 (delta 0)
```

```
Unpacking objects: 100% (44/44), done.
```

```
From git://github.com/paulboone/ticgit
```

```
* [new branch]      master      -> pb/master
```

```
* [new branch]      ticgit      -> pb/ticgit
```

Repositorios remotos en GIT

AÑADIENDO REPOSITORIOS REMOTOS

- ▶ Para recuperar datos de tus repositorios remotos puedes ejecutar: `$ git fetch [remote-name]`
- ▶ Este comando recupera todos los datos del proyecto remoto que no tengas todavía. Después de hacer esto, deberías tener referencias a todas las ramas del repositorio remoto, que puedes unir o inspeccionar en cualquier momento

Repositorios remotos en GIT

ENVIANDO A TUS REPOSITORIOS REMOTOS

- ▶ Cuando tu proyecto se encuentra en un estado que quieres compartir, tienes que enviarlo a un repositorio remoto.
- ▶ El comando que te permite hacer esto es sencillo: “git push [nombre-remoto][nombre-rama]”.
- ▶ Si quieres enviar tu rama maestra (master) a tu servidor origen (origin), ejecutarías esto para enviar tu trabajo al servidor:
- ▶ `$ git push origin master`
- ▶ Este comando funciona únicamente si has clonado de un servidor en el que tienes permiso de escritura, y nadie ha enviado información mientras tanto. En caso de que haya cambios, deberías bajarte antes la última versión del servidor.

Repositorios remotos en GIT

INSPECCIONANDO REPOSITORIOS REMOTOS

- ▶ Si quieres ver más información acerca de un repositorio remoto en particular, puedes usar el comando `git remote show [nombre]`.
- ▶ Si ejecutas este comando pasándole el nombre de un repositorio, como origin, obtienes algo así:

```
$ git remote show origin * remote origin URL:  
git://github.com/schacon/ticgit.git Remote branch merged with 'git  
pull' while on branch master master Tracked remote branches master  
ticgit
```

- ▶ Esto lista la URL del repositorio remoto, así como información sobre las ramas bajo seguimiento. Este comando te recuerda que si estás en la rama maestra y ejecutas "git pull", automáticamente unirá los cambios a la rama maestra del remoto después de haber recuperado todas las referencias remotas. También lista todas las referencias remotas que ha recibido

Repositorios remotos en GIT

ELIMINANDO Y RENOMBRANDO REPOSITORIOS

- ▶ Si quieres renombrar una referencia a un repositorio remoto, en versiones recientes de Git puedes ejecutar `git remote rename`. Por ejemplo, si quieres renombrar pb a paul, puedes hacerlo de la siguiente manera:

```
$ git remote rename pb paul
```

```
$ git remote
```

```
origin
```

```
paul
```

- ▶ Esto también cambia el nombre de tus ramas remotas. Lo que antes era referenciado en pb/master ahora está en paul/master.
- ▶ Si por algún motivo quieres eliminar una referencia(has movido el servidor o ya no estás usando un determinado mirror, o quizás un contribuidor ha dejado de contribuir) puedes usar el comando `git remote rm`:

```
$ git remote rm paul
```

```
$ git remote
```

```
origin
```

Repositorios remotos - GitHub

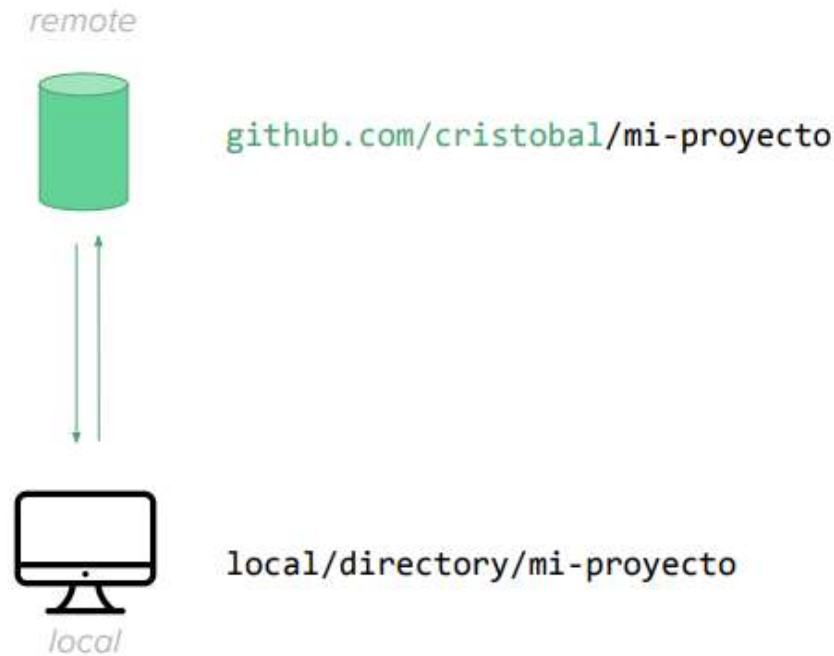
- ▶ GitHub es una plataforma de desarrollo colaborativo de software para alojar proyectos utilizando el sistema de control de versiones (VCS) Git
- ▶ Brinda herramientas muy útiles para el trabajo en equipo dentro un proyecto
- ▶ Es necesario crear una cuenta en GitHub para trabajar con repositorios remotos

<https://github.com/>

Repositorios remotos - GitHub

Recordando:

- Un repositorio es la unidad básica. La forma más fácil de imaginarlo es como la carpeta donde se encuentra nuestro proyecto.
- En él tendremos nuestro código, podremos hacer una copia local (repositorio local) desde la cual podremos actualizar y subir cambios al repositorio remoto, y un largo etc



Repositorios remotos - GitHub

TRABAJANDO CON “COMMIT”

- ▶ Conjunto de cambios que se hacen al repositorio.
- ▶ Imaginemos que modificamos una línea de código de un archivo de nuestro repositorio. La forma de actualizarlo es hacer un “commit” de ese cambio.
- ▶ Antes de hacerlo, hemos de ver qué cambios han surgido en nuestro repositorio local comparándolo con el repo remoto. Después añadimos los cambios y finalmente hacemos el commit (empaquetamos los cambios).

Repositorios remotos - GitHub

COMPROBAR DIFERENCIAS ENTRE REPOSITORIO “LOCAL” Y “REMOTO”

► \$ git status

```
λ git status
On branch ocp-3.6
Your branch is up-to-date with 'origin/ocp-3.6'.
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   cicd-template.yaml

no changes added to commit (use "git add" and/or "git commit -a")
```

Repositorios remotos - GitHub

COMPLETANDO UN “COMMIT”

- ▶ `$ git add .`
- ▶ `$ git status`

```
λ git status
On branch ocp-3.6
Your branch is up-to-date with 'origin/ocp-3.6'.
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    modified:   cicd-template.yaml
```

Repositorios remotos - GitHub

COMPLETANDO UN “COMMIT”

- ▶ Finalmente empaquetamos los cambios añadidos en un commit.
- ▶ `$ git commit -m “mensaje sobre los cambios”`

```
λ git status
On branch ocp-3.6
Your branch is up-to-date with 'origin/ocp-3.6'.
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    modified:   cicd-template.yaml
```


Repositorios remotos - GitHub

HACER UN “PUSH” AL REPOSITORIO REMOTO

- ▶ Cuando hacemos un PUSH significa que estamos “empujando” el paquete de cambios (el commit) al repositorio remoto.
- ▶ En el caso de que estemos en la rama ejemplo del repo local, los cambios serán actualizados en la rama ejemplo del repo remoto.
- ▶ `$ git push`

Repositorios remotos - GitHub

ACTUALIZAR REPOSITORIO LOCAL - USO DE “PULL”

- ▶ Para actualizar tu repositorio local al commit más nuevo se ejecuta el pull en el directorio de trabajo para “bajar y fusionar” los cambios remotos
- ▶ Si el repositorio local está actualizado, la consola mostrará un “Already up-to-date”
- ▶ `$ git pull`

Repositorios remotos - GitHub

CONFLICTOS PUSH/PULL

- ▶ Se puede dar el caso en el que existan conflictos
- ▶ Esto ocurre cuando se intenta hacer un PUSH a un repositorio que no tenemos actualizado en local
- ▶ Para ello, deberíamos hacer un PULL del cambio que no tenemos en local, generar un commit de nuevo y hacer un push de él (habiendo solucionado los conflictos)

Repositorios remotos - GitHub

TRABAJANDO CON PUSH/PULL

\$ git push

Caso 1: Todo ha ido bien. El repo se actualiza.

Caso 2: \nexists conflictos. Pero otrx compañerx ha hecho *push* antes que tú. Esto implica:

1. `$ git pull`
2. `$ git commit -m "merge with latest changes"`
3. `$ git push`

Caso 3: \exists conflictos. El código que pretendes “empujar” el repo entra en contradicción con el del último *commit*.

1. `$ git pull`
2. `-- resolver conflictos (visibles en tu IDE) --`
3. `$ git add .`
4. `$ git commit -m "solved merge conflicts"`
5. `$ git push`

Repositorios remotos - GitHub

- ▶ Ejercicio: Trabajar con GitHub. Crear nuestro primer repositorio remoto y subir nuestro proyecto local
 - ▶ Crear una cuenta
 - ▶ Sincronizarlo con tu repositorio local. Es decir. Hacer un “PUSH” desde el repositorio local
 - ▶ Ojo: en nuestro primer uso, tenemos que autenticarnos en GitHub con claves generas HTTP/SSH desde GIT:
 - ▶ <https://help.github.com/articles/set-up-git/>
 - ▶ Tutorial para subir la clave SSH generada a nuestro repositorio remoto:
<https://www.adictosaltrabajo.com/tutoriales/github-first-steps-upload-project/>
 - ▶ Tutorial para configurar Github:
 - ▶ <https://conociendogithub.readthedocs.io/en/latest/data/dinamica-de-uso/>