

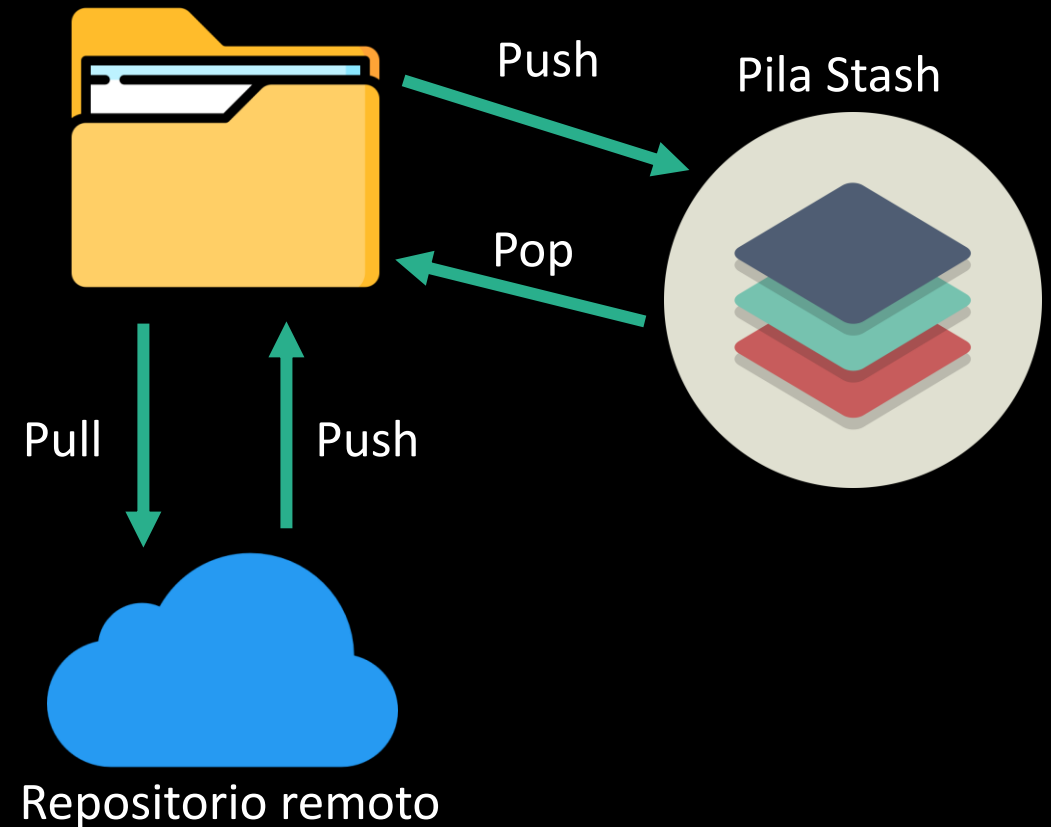
TÉCNICAS DE INTEGRACIÓN DE CÓDIGO

STASH / SOURCETREE

Stash... ¿qué es eso?

- Si tenemos cambios en nuestros archivos, tenemos que bajar cambios o comenzar a trabajar en otra cosa, y queremos mantener nuestros cambios, podemos hacer un "stash".
- Los stash se almacenan en una pila.
- Los stash se almacenan de forma local, no se suben al remoto

Carpeta del repo
(local)



¿Cómo hacemos stash?

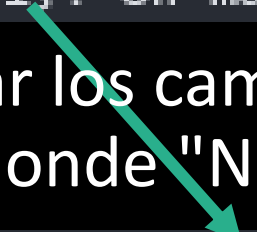
- Debemos de tener cambios en nuestros archivos por "stashear".
- Con el comando *git stash push -m "[MENSAJE_STASH]"* podemos crear un stash.

```
> git stash push -m "STASH DE PRUEBA DE DORX"
Saved working directory and index state On main: STASH DE PRUEBA DE DORX
```

- Podemos ver la lista de stashes que se tienen con *git stash list*.

```
> git stash list
stash@{0}: On main: STASH DE PRUEBA DE DORX
stash@{1}: On main: SEGUNDO STASH
```

- Para volver a aplicar los cambios del stash, podemos usar el comando *git stash apply N*, donde "N" es el número del stash.



```
> git stash apply 1
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   index.html
```

Consideraciones de Stash

- Con `git stash pop N` donde "N" es el número del stash, aplicamos el stash, y después, se elimina de la lista de stashes.
- Tanto al comando `git stash pop` y a `git stash apply` podemos no pasarle el número del stash y va a tomar el último de la lista.
- Si hacemos `git stash`, se creará un stash de todos los cambios sin mensaje.
- Podemos usar el comando `git stash drop N`, donde "N" es el número del stash, para eliminar un stash de la lista.
- Considera que los stashes **quitan los cambios** de los archivos en tu directorio local.

Probemos los stashes...



1. En individual, utiliza el repositorio de tu práctica para cambiar tu archivo "index.html" haciendo el cambio que quieras. Guarda cambios.
2. Si ejecutas `git status`, verás que tienes cambios pendientes en ese archivo. No haremos commit. Crea un stash de dicho archivo con `git stash push -m "MENSAJE"`, usando el mensaje que quieras.
3. Ejecuta `git stash list` para ver el stash que tienes. Verifica también que tus cambios en index.html ya no estarán.
4. Ahora, modifica tu hoja de estilos con lo que quieras, y repite el procedimiento de los pasos 1 a 3.
5. Ahora que tienes dos elementos en tu pila de stashes, utiliza **pop** para aplicar y eliminar el stash que hiciste de index.html, y **apply** para aplicar, sin eliminar, el stash de tu hoja de estilos.

Más pruebas de stashes... why not?

- En el mismo repositorio de tu práctica, crea 5 stashes modificando los archivos que quieras. Asegúrate de poner un mensaje en cada uno de los stashes.
- Verifica que los tienes correctamente con *git stash list*.
- Aplica sóloamente los stashes pares a tu carpeta, utilizando **pop** para que se eliminen de tu lista de stashes.
- Ahora, aplica los stashes nones a tu repositorio, utilizando **apply** para que no se eliminen.

Atlassian SourceTree

The screenshot displays the Atlassian SourceTree application interface. The top menu bar includes File, Edit, View, Repository, Actions, Tools, and Help. Below the menu is a toolbar with icons for Commit, Pull, Push, Fetch, Branch, Merge, Stash, Discard, and Tag. The main workspace is divided into three panes:

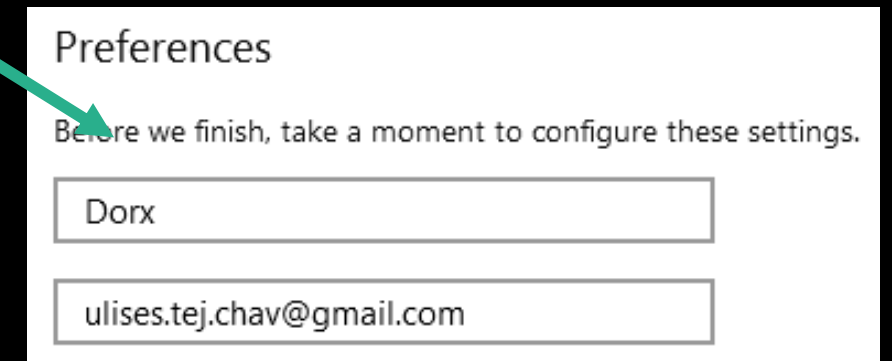
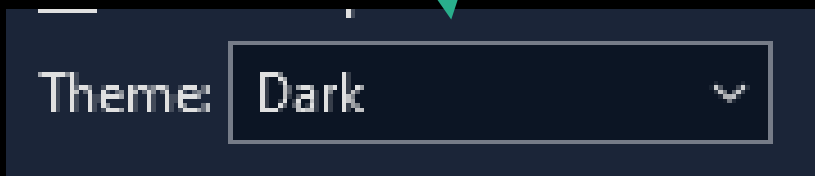
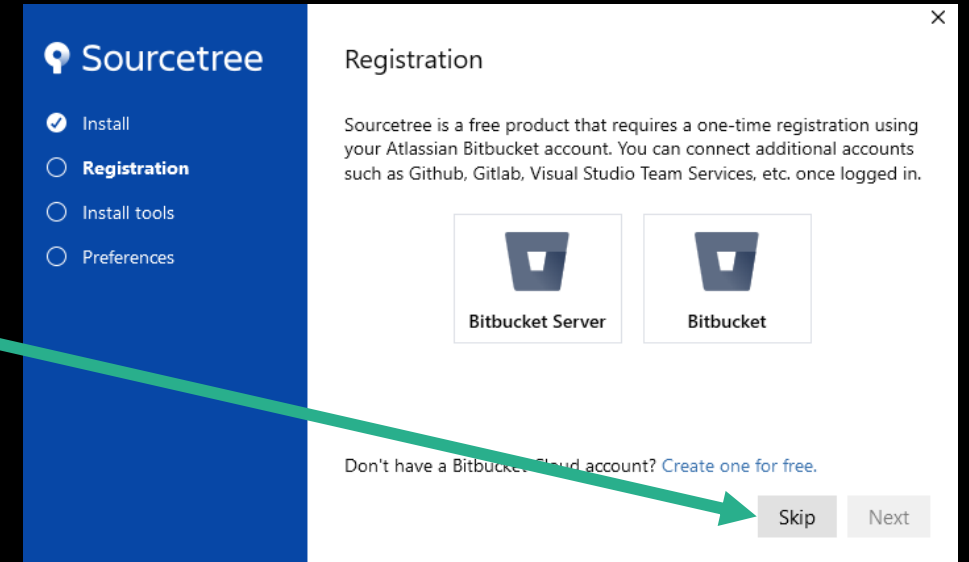
- Left Pane (Workspace):** Contains a sidebar with sections for File Status, History, Search, BRANCHES (listing develop, info_user_screen, master), TAGS, REMOTES (listing origin), and STASHES.
- Middle Pane:** Shows a commit history graph and a list of commits. The selected commit is 2f0d7c1, titled "Merge branch 'develop' of gitlab.com:uxbi-start/uxbi-backend into develop". The commit message describes updates to the report generation process and the addition of a new endpoint for generating reports.
- Right Pane:** Displays the code editor for the file `app/Classes/Reportes/AntiguedadSalDOS.php`. The editor shows the PHP code for the `AntiguedadSalDOS` class, including methods for generating reports and handling client data.

The commit history table includes columns for Date, Author, and Commit. The code editor shows the following PHP code:

```
55 ..... $idEmpresa = (int) $filters->get('id_empresa');
56 ..... $fechaCorte = $this->validateDate($filters->get('fecha_corte', null));
57 ..... $periodos = $filters->get('periodos', 30);
58 ..... $getSalDOS = (boolean) $filters->get('salDOS', false);
59 ..... $getSalDOS = $filters->get('salDOS', false);
60 ..... if ($getSalDOS == "false") {
61 .....     $getSalDOS = 0;
62 ..... } else {
63 .....     $getSalDOS = 1;
64 ..... }
65 ..... #.Obtiene arreglo de clientes
66 ..... if (is_string($arrClientes = $filters->get('clientes', []))) $arrClientes = json_decode($arrClientes, true);
67 ..... }
```

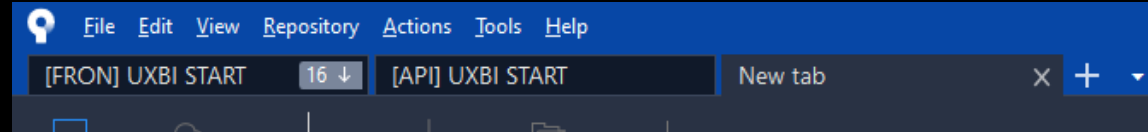
Comencemos a usar SourceTree


- Descarga la aplicación desde <https://www.sourcetreeapp.com/>.
- Al momento de instalarlo, si te pide hacer un registro, lo puedes omitir con "skip":
- Configura tu correo y nombre para tus commits.
- Por el amor de Dios, cambia la interfaz a modo oscuro en "Tools" -> "Options" -> "General" -> "Theme":

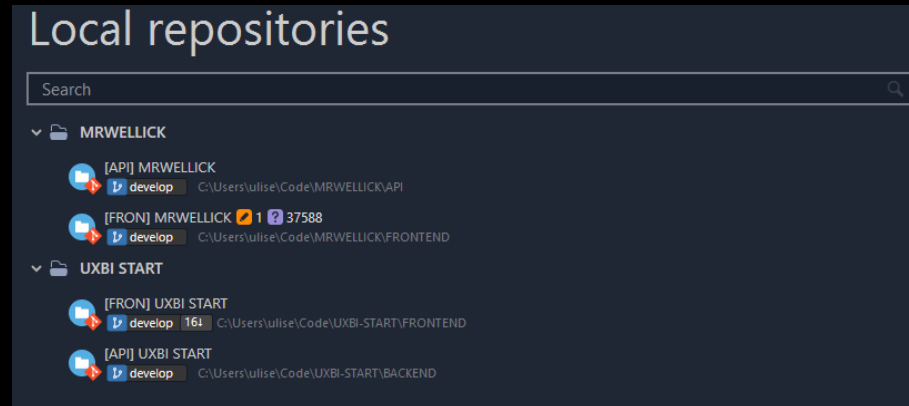


A saber antes de clonar...

- Las pestañas que tienes arriba son los repositorios:



- Puedes abrir una nueva pestaña con el , y ahí, tendrás la lista de tus repositorios locales:



- Los puedes organizar por carpetas (no son directorios de tu explorador de archivos) creando una nueva con  New Folder .

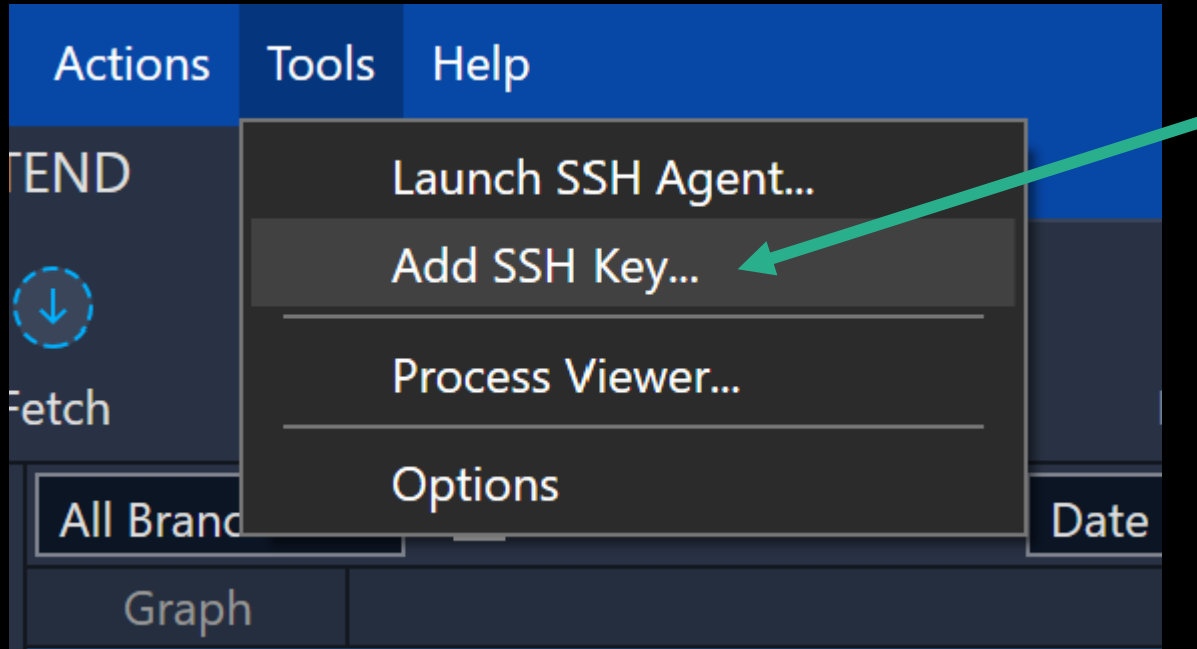
Los comandos que podemos hacer:

- Clone
- Status (implícito)
- Checkout
- Fetch
- Add
- Branches
- Commit
- Pull
- Push
- Ignore
- Stash
- Y más...

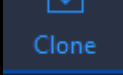


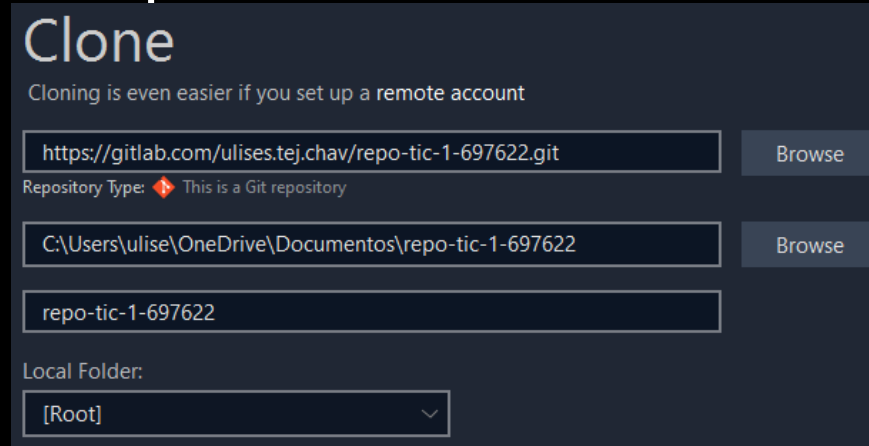
Llaves SSH

Con SourceTree puedes utilizar las llaves SSH que tienes para poder acceder a tus repositorios, desde "Tools" -> "Add SSH Key". Seleccionas tu archivo público (no el privado).



Clone


- En una nueva pestaña, usamos el botón .
- En esta interfaz, especificamos la URL del repositorio y la carpeta en la que estará, así como el nombre que le vamos a poner en SourceTree. Podemos ponerlo en una de las carpetas vistas hace un momento.



Clone

Cloning is even easier if you set up a remote account

Repository URL:

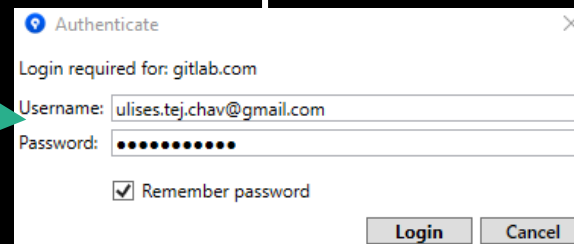
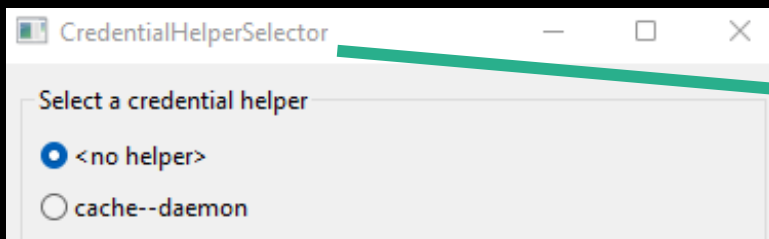
Repository Type:  This is a Git repository

Local Path:

Repository Name:

Local Folder:

- Si nos aparece el "CredentialHelperSelector", elegimos <no-helper> e iniciamos sesión en el manejador de repositorios.



- Finalmente, lo clonamos con .

Con esto, ya tenemos nuestro repo:

Rama actual

Gráfico de ramas y commits

Ramas locales

Ramas en el remoto

Stashes

Mensajes de commits

Fecha del commit

Autor del commit

Información del commit seleccionado

Cambios del commit seleccionado

The screenshot shows the Git GUI application interface. The top menu bar includes File, Edit, View, Repository, Actions, Tools, and Help. The main window is divided into several sections:

- Left Sidebar:** Contains 'WORKSPACE' (File Status, History, Search), 'BRANCHES' (main), 'TAGS', 'REMOTES' (origin: HEAD, main, rama_prueba_1), and 'STASHES'.
- Top Bar:** Shows repository name 'repo-tic-1-697622' and various action buttons like Commit, Pull, Push, Fetch, Branch, Merge, Stash, Discard, and Tag.
- Graph View:** Displays a commit history graph with a table of commits. The current branch 'testing' is highlighted.
- Commit Details:** A box showing details for the selected commit (1d85ffcd50290a3acd795ff7bdb027c833b95de1), including its parents, author, date, and committer.
- File Explorer:** Shows the selected file 'archivo_prueba.html' with its changes highlighted in a diff view.

Description	Date	Author	Commit
testing	21 feb. 2023 11:23	Dorx <ulises.tej.cl	1d85ffc
origin/rama_prueba_1 Se agregó gitignore	1 feb. 2023 12:56	Dorx <ulises.tej.ch	d199200
Update archivo_prueba.html	31 ene. 2023 16:39	Ulises Tejeda Cháv	13694b6
Ajustes iniciales archivo prueba	31 ene. 2023 15:52	Dorx <ulises.tej.ch	fea58bb
Carga inicial archivo_prueba.html	31 ene. 2023 15:00	Dorx <ulises.tej.ch	d957d6c
Initial commit	31 ene. 2023 13:58	Ulises Tejeda Cháv	14c42de

Commit: 1d85ffcd50290a3acd795ff7bdb027c833b95de1 [1d85ffc]
Parents: 13694b6406
Author: Dorx <ulises.tej.chav@gmail.com>
Date: martes, 21 de febrero de 2023 11:23:22 a. m.
Committer: Dorx

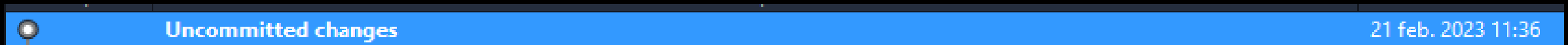
archivo_prueba.html

Hunk 1 : Lines 8-13

```
8 .....<h1>Hola mundo</h1>  
9 .....<p>Soy un ejemplo para la clase de TI  
10 .....<p>Soy un segundo párrafo</p>  
11 .....PRUEBA  
12 .....</body>  
13 .....</html>
```

Status

- Cuando hacemos cambios, en el historial de commits nos aparecerá así:



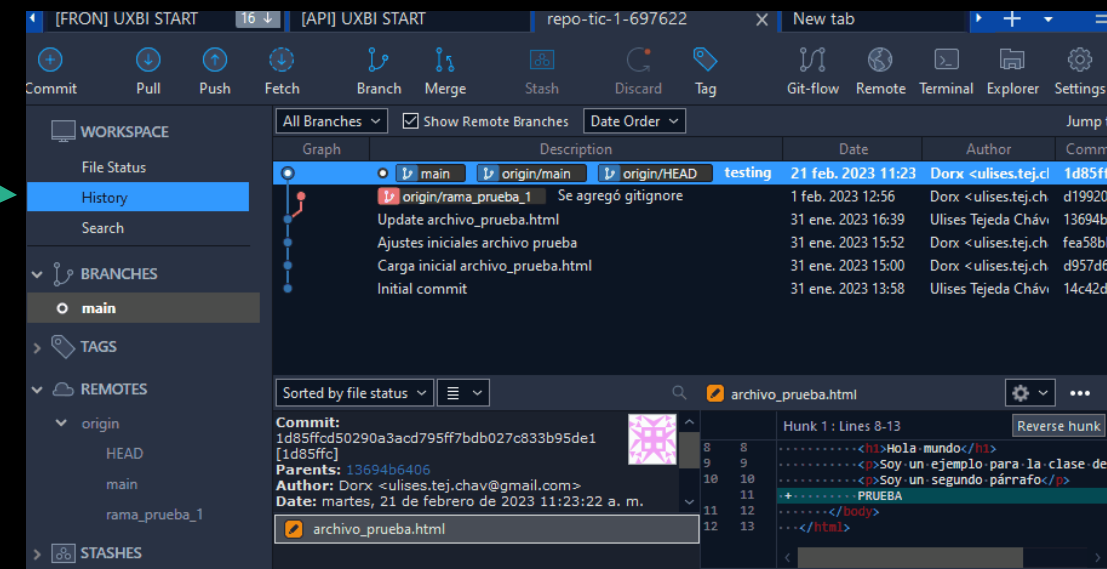
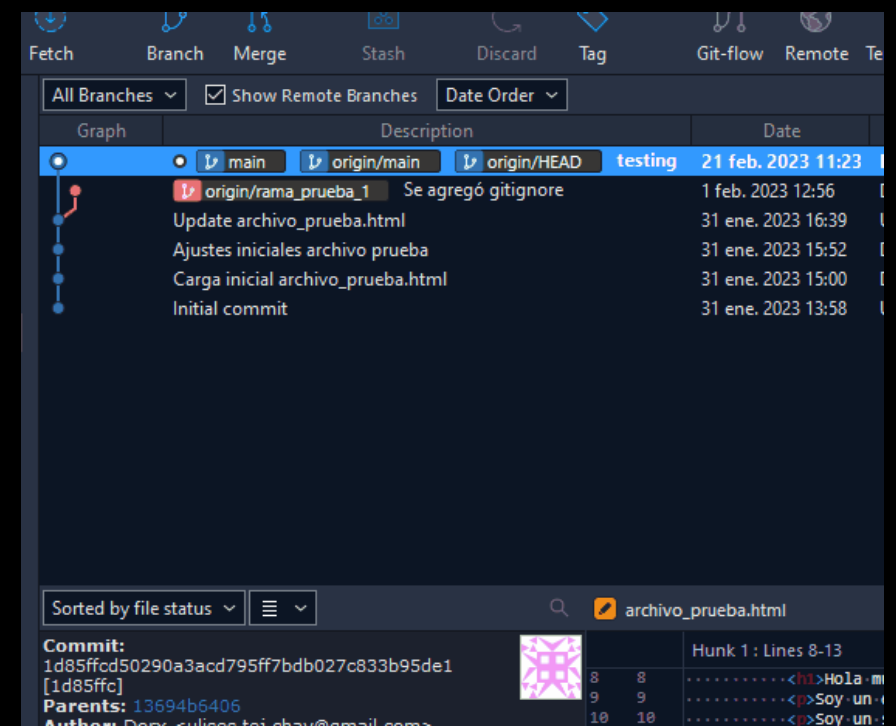
- Y al seleccionarlo, nos aparecerá así en la parte de abajo:



- Puedes dar Click derecho y "Discard" para borrar los cambios de tu directorio local.

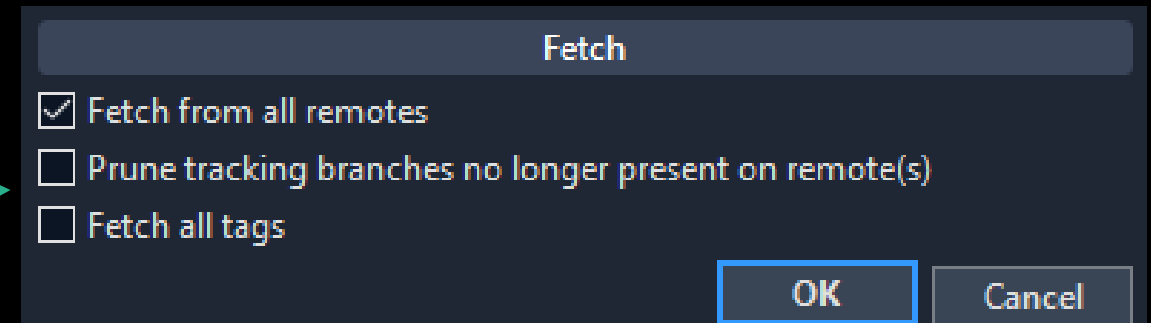
Checkout

- Para hacer checkout de un commit, basta con hacer doble click al commit que quieras. Considera que esto moverá tu Head.
- Para hacer checkout de una rama, basta con hacer doble click a la misma en el menú de la izquierda. Esto creará la rama en tu repositorio local.
- Si no la tienes ya, se creará en local. Si ya las tienes, puedes cambiar entre ellas dando doble click a la que quieras.



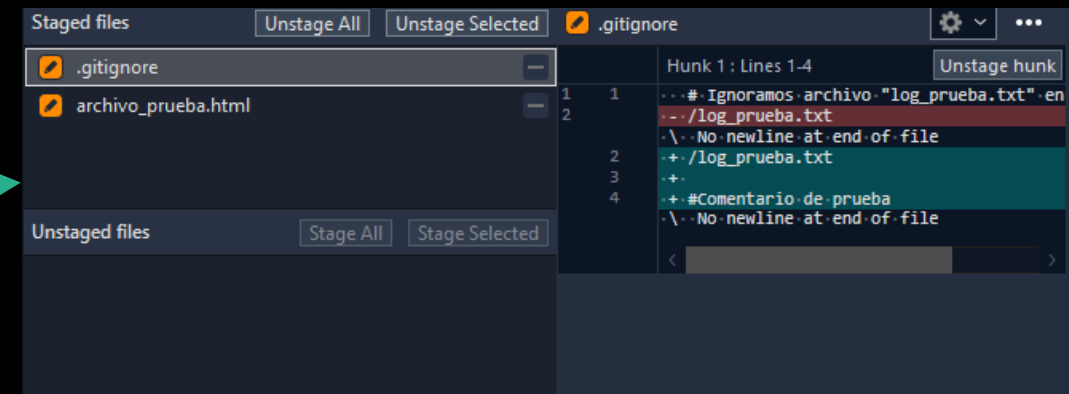
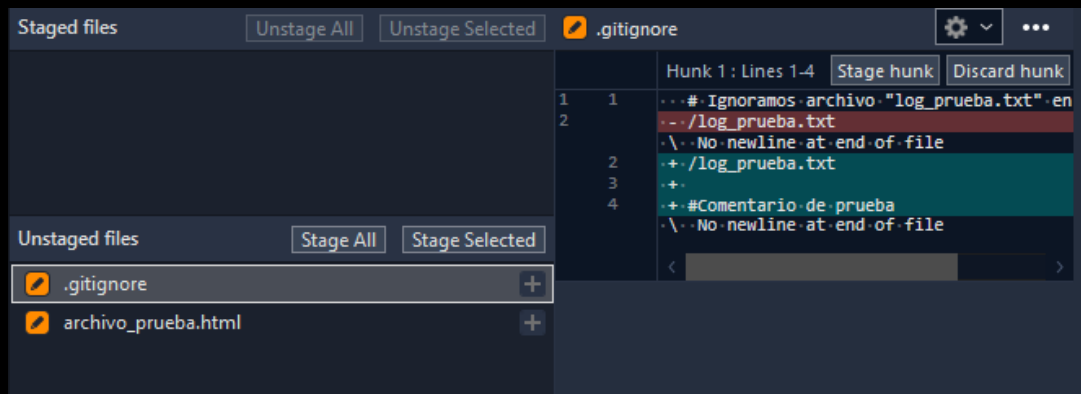
Fetch

- Pues... pícale a "Fetch":

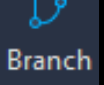


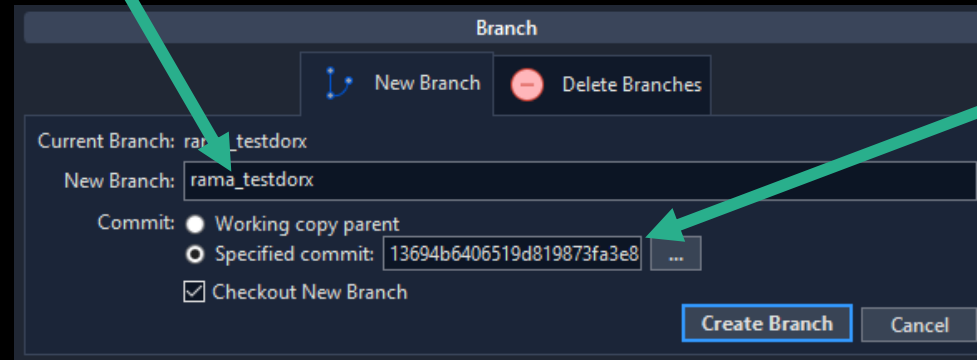
Add

- Puedes agregar los archivos seleccionados a tu siguiente commit con el botón **Stage Selected**, o para agregarlos todos con **Stage All**. Puedes seleccionar varios con Ctrl o con Shift, como archivos del explorador.
- Con eso, ya tendrás los archivos agregados para commitear en la sección "Staged Files". Los puedes quitar con **Unstage All** o **Unstage Selected**.

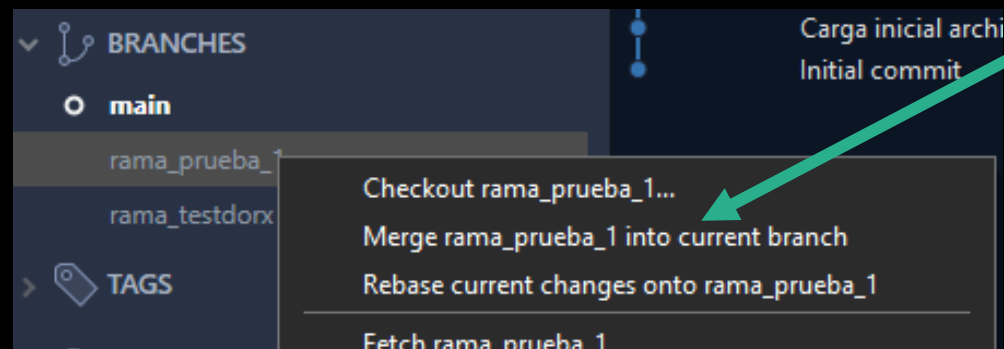


Branch

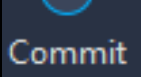
- Podemos crear ramas nuevas utilizando el botón  .
- Especificamos el nombre de la rama y, si queremos, a partir de qué commit:



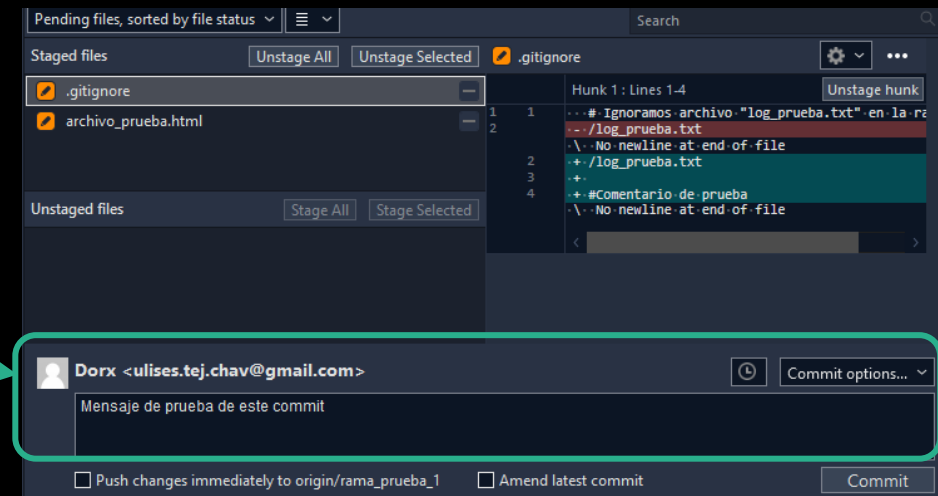
- Podemos hacer merge de otras ramas en la rama actual dando click derecho en la rama que queremos mergear y eligiendo la opción "Merge [RAMA] into current branch".



Commit


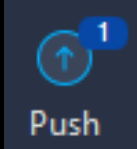
- Puedes hacer commit con . Este botón tiene una insignia que te indica los cambios pendientes (en este caso, dos).
- Te mostrará esta interfaz con los cambios que tienes (puedes quitar o añadir cambios desde aquí también).

Espacio para mensaje del commit



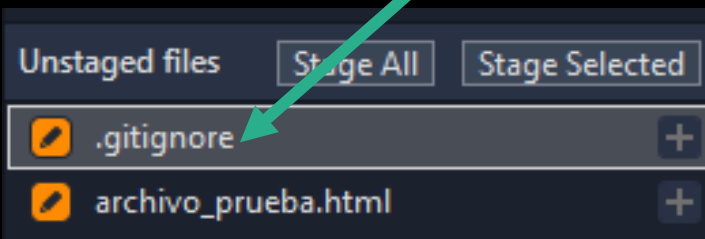
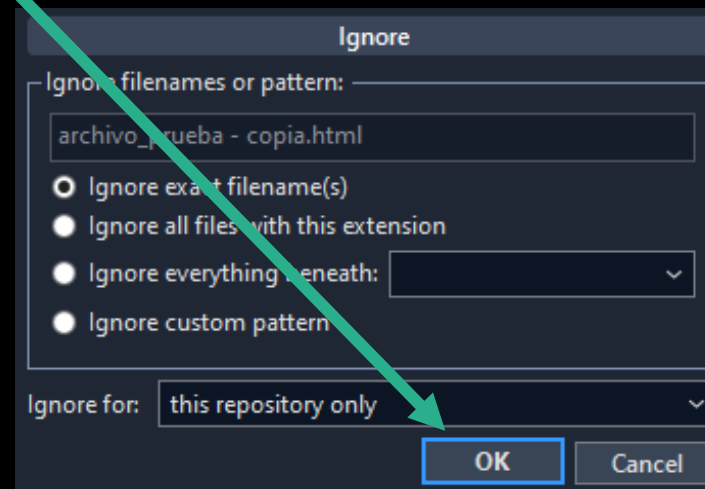
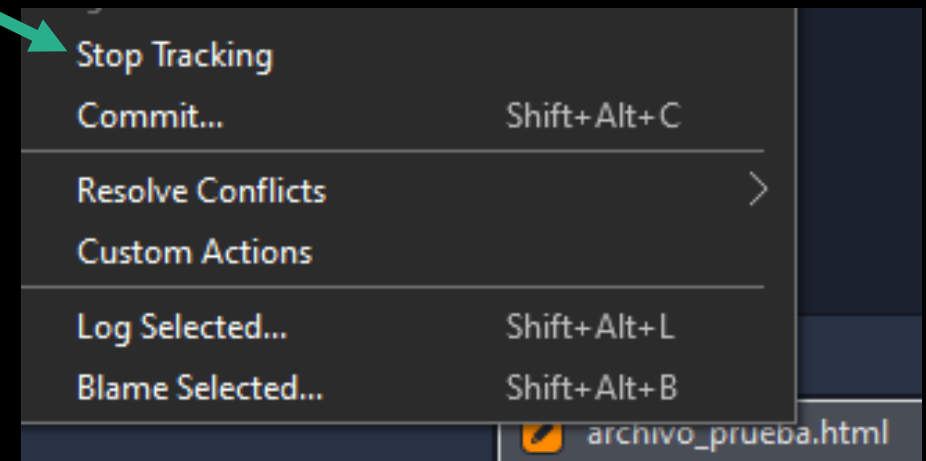
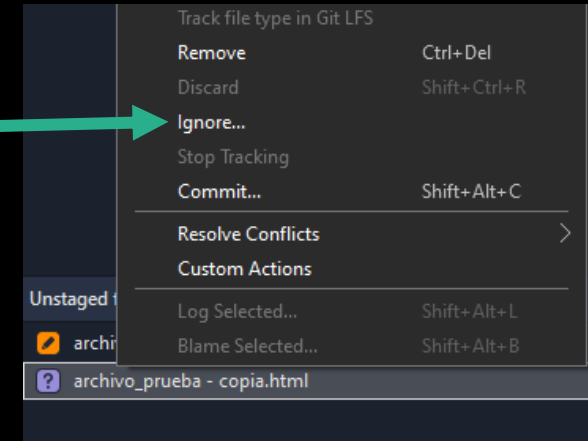
- Hay que especificar el mensaje del commit, y pulsar el botón .

Pull y Push


- Para hacer pull (descargar los cambios del repositorio), simplemente hay que pulsar el botón .
- Para hacer Push, hay que pulsar el botón . Este botón tiene una insignia con los commits pendientes por subir, en este caso, 1.

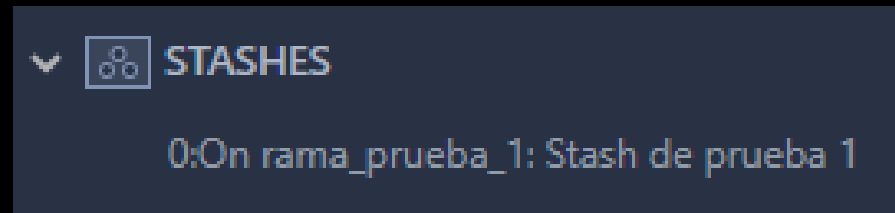
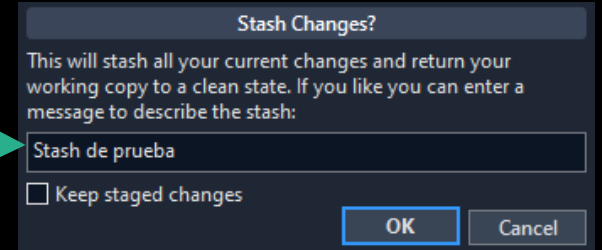
Ignore

- Para ignorar archivos nuevos, podemos darles click derecho y elegir la opción de "Ignore".
- Si es un archivo existente, primero tenemos que elegir "Stop Tracking" y después, "Ignore".
- Podemos elegir las condiciones del ignore, y al dar "OK", nos modificará el .gitignore para hacerle commit.

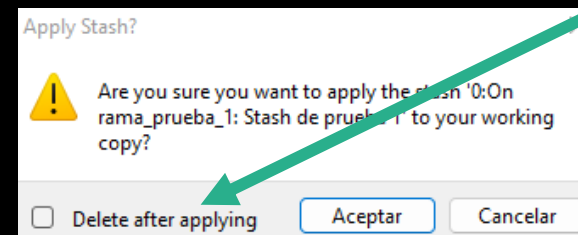
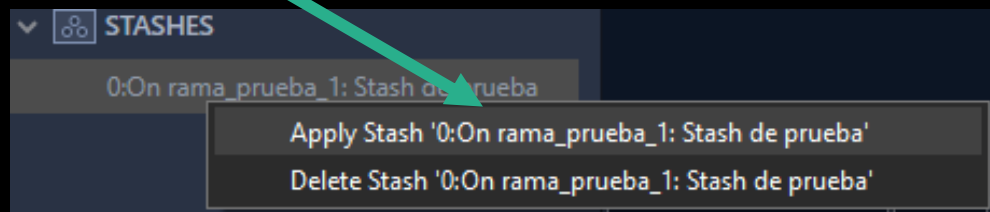


Stash:


- Para hacer stash, utilizamos el botón de  Stash.
- Nos pedirá un mensaje para el stash.
- Podemos ver la lista de stashes a la izquierda:

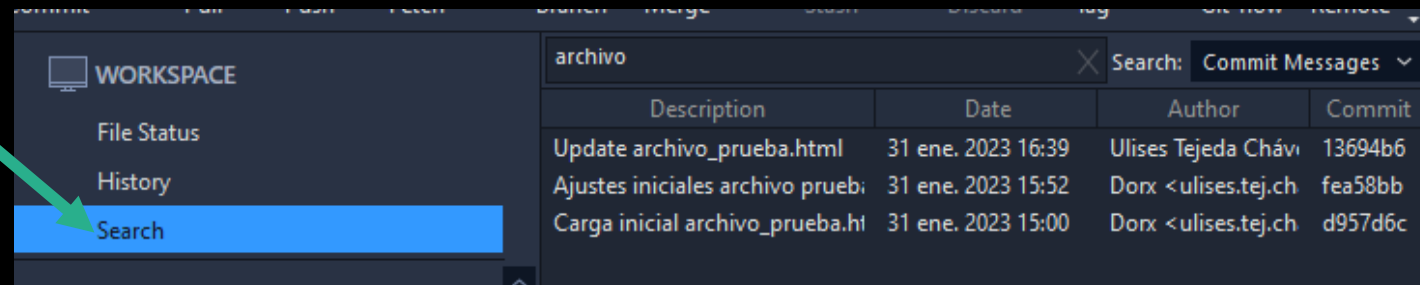


- Podemos aplicar o borrar el stash dando click derecho al mismo y eligiendo la opción deseada. Al aplicarlo, lo podemos borrar (hacer pop)



Y más...

- Si queremos trabajar desde la terminal para algún comando en específico, podemos hacerlo desde el botón  .
- Podemos buscar commits o cambios en específico en la parte de "Search". Puedes buscar por mensajes, autores o código específico.



Y la siguiente clase... ¿pa' qué?

Tendremos una actividad con los mismos equipos de la práctica. Subiré la actividad a Canvas, y no tomaré asistencia. Puedes venir a trabajar con tu equipo (estaré aquí por si tienes dudas), o hacerlo por tu cuenta.

