

Introducció a GIT

Laura Rivera

Laura Rivera



- Enginyera informàtica
- Màster en Ciència de dades
- Co-fundadora Class Escape
- Professora de base de dades i programació a la UAB i Tecnocampus de Mataró.

<https://www.linkedin.com/in/laura-rivera-08039819/>

Objectiu

Entendre els conceptes bàsics de que és GIT, com s'utilitza i ho practicarem per a que ho apliqueu als vostres projectes.

Distribució de la sessió

1. Part teòrica

2. Exemples guiats

3. Exercici individual

4. Exercici en grup



Què és GIT?

Software de control de versions que ens permet tenir un registre dels canvis que es realitzen en els arxius i coordinar-nos amb altres persones de l'equip.

- És el més utilitzat.
- De codi obert / gratuïta
- Constants millores.
- Integració en molts IDE* (Android Studio, Eclipse, etc.)

Com ho fa?

- Crea una instantània del codi / fitxer
- Es guarda els canvis d'una versió a l'altre: què s'ha eliminat, què s'ha afegit.
- Permet la creació de branques.

*IDE

Acrònim del terme anglès Integrated Development Environment o, el que és el mateix, **Entorn de Desenvolupament Integrat**

GIT (Local)

GITHUB (Remot)



Repositori del projecte



- Lloc web i un servei al núvol (remot)
- Ajuda els desenvolupadors a emmagatzemar i administrar el seu codi

<https://github.com/>

- Portar un registre i control de qualsevol canvi sobre aquest codi (perquè utilitza GIT)
- Gratuït.
- Permet compartir codi.
- Permet recuperar codi en altre PC.
- Permet crear [pàgines](#) senzilles

Altres similars:

- Bitbucket
- Gitlab
- AWS (Amazon)
- GoogleCloud



Línia de comandes

CMD en Windows, Terminal en Mac o Linux.

Disposa d'unes comandes
específiques per a cada acció:

- Afegir al repositori local.
- Preparar la versió a local.
- Enviar la versió al repositori remot.
- Etc.

Programes

Permeten realitzar les comandes de línia de comandes de manera visual.

A més, podem veure les diferències dels fitxers.

Exemples: Sourcetree, GitKraken



(Parèntesis)

CMD? What?

Repàs ràpid de com utilitzem
la línia de comandes

- Help : apareixen les comandes disponibles.
- Exit : Tanca el terminal / cmd
- Cls: limpiar el historial

Carpetes:

- Dir: mostra les carpetes que hi ha al directori.
- Cd: mostra en quina carpeta estem
- Cd “nom carpeta”: ens envia a la carpeta especificada
- Cd .. : tornar enrere
- Mkdir : Crear nova carpeta

- Move: per a moure arxius

- Ctrl+C : Per finalitzar algun procés que tarda massa.

Exercici 1 CMD

Crea una carpeta al escriptori o on vulguis.

Des de línia de comandes, crea dues carpetes “c1” i “c2”

Afegeix qualsevol fitxer a la carpeta “c1”

Des de línia de comandes, mou el fitxer de “c1” a “c2”

Neteja l’historial

Tanca el cmd amb “exit”



Què instal·lem?

<http://www.git-scm.com/downloads>

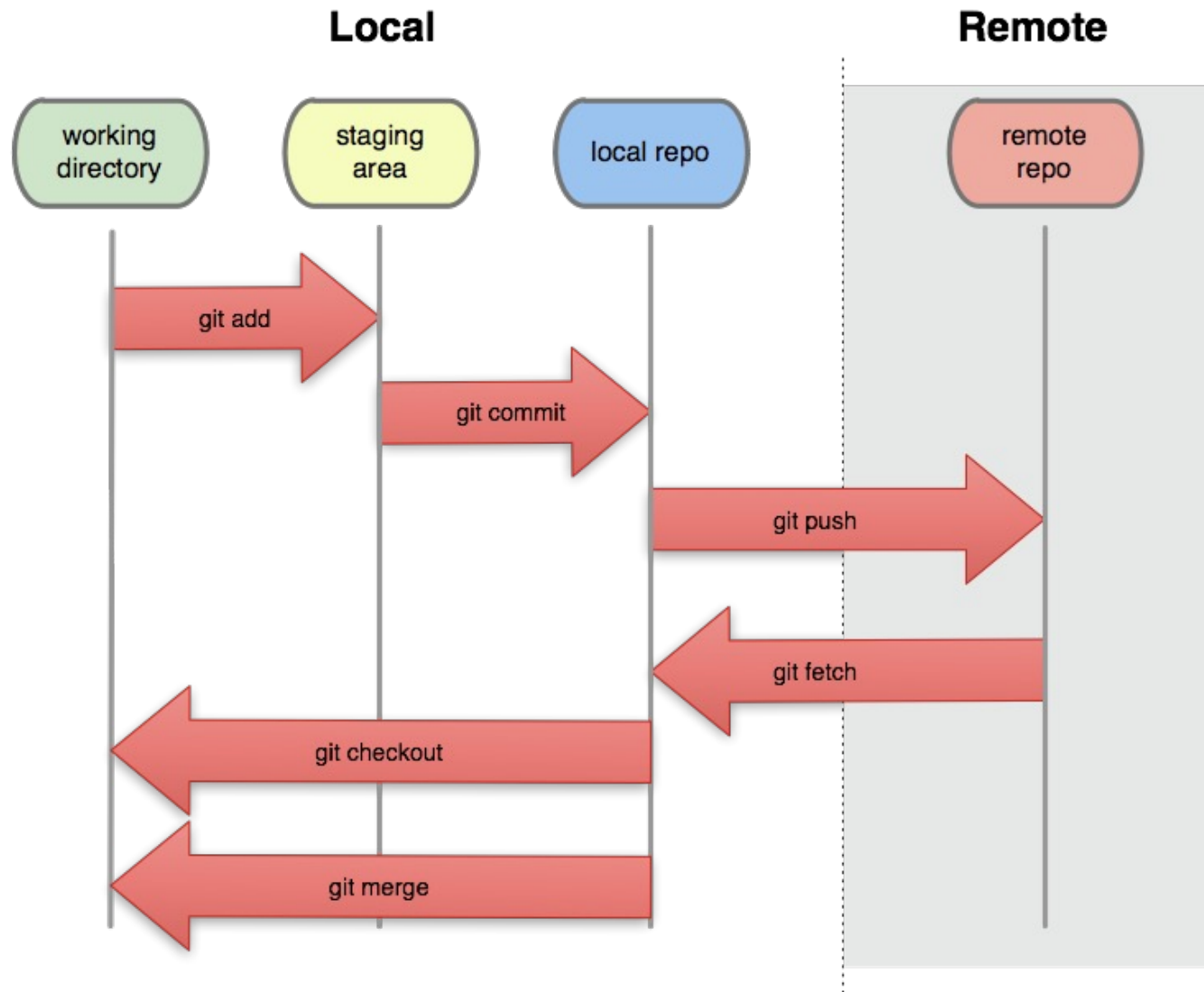


LOCAL

Crear compte a Github

<https://github.com/>

REMOT



Github – crear repositori

- 1- Accedeix al web de github, inicia sessió.
- 2- Selecciona “New” per a crear el nou repositori
- 3- Afegeix el nom del repositori.

Recent Repositories

 New

Find a repository...


Repository name *

/ git_test




Owner *

Repository name *



 Iriverauab ▼

 /

git_test 

Great repository names are short and memorable. Need inspiration? How about **ubiquitous-octo-waddle**?

Description (optional)

- ☒  **Public**
Anyone on the internet can see this repository. You choose who can commit.
- ☐  **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

- ☐ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore

Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: None ▼

Visible o no per tothom

Readme és un fitxer que apareix com a descripció del projecte

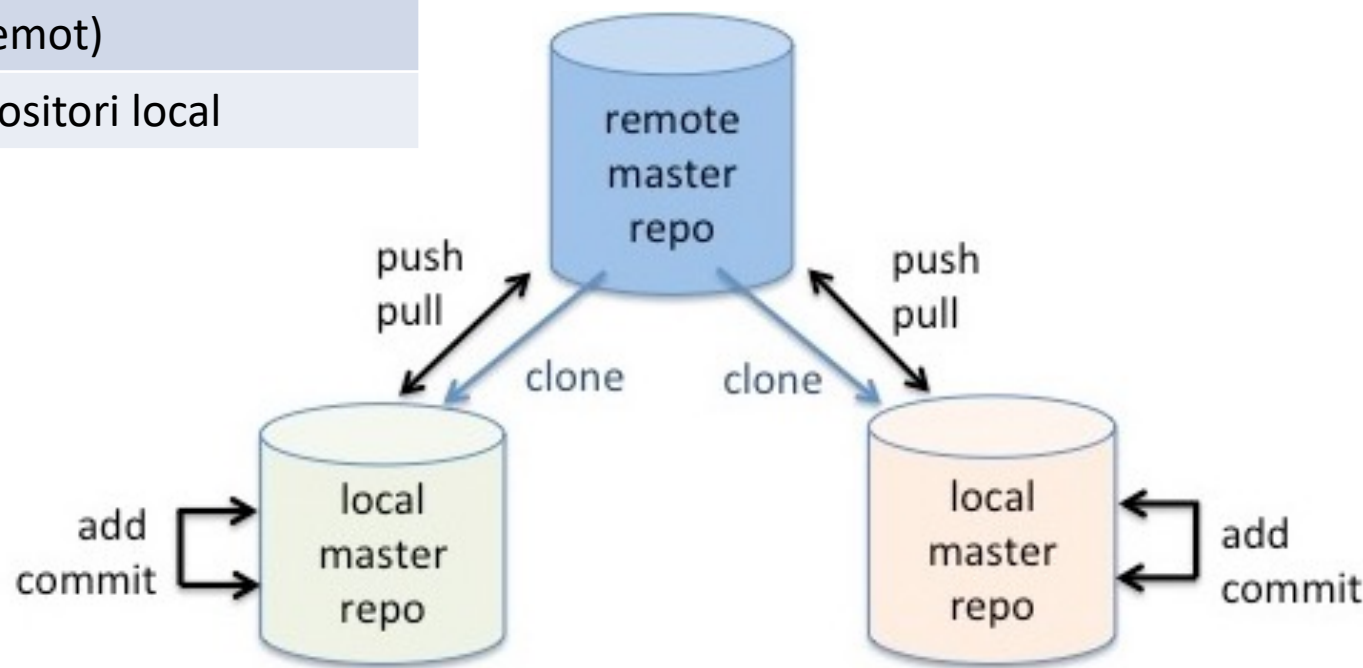
.gitignore per indica si volem que ignori fitxers o extensions



Git – comandes bàsiques

[Fetch vs Pull](#)

Comanda	Descripció
Init	Inicialitza un nou repositori
Add	Afegeix fitxers per a un nou commit (paquet)
Status	Ens permet veure els fitxers afegits al paquet
Commit	Prepara el paquet (encara estem a local)
Push	Envia els canvis al repositori del github (remot)
Pull	Recollir els canvis del github al nostre repositori local



Git – local (PC)

Opció 1: Ja teníem codi a local.

git init -> inicialitzar el repositori a local

git add -> afegir fitxer/s a un “paquet”

git commit -m “missatge” -> crear un primer commit “paquet” amb un missatge

git remote add origin <https://github.com/usuario/repo.git>

git push -u origin **master**



Exercici 2

- 1- Crea una carpeta amb fitxers al teu ordinador
- 2- Des de la línia de comandes inicialitza un repositori
- 3- Prepara un paquet (add + commit)
- 4- Crea un repositori buit al github
- 5- Sincronitza el repositori local amb el remot (add origin)
- 6- Fes push



Git – local (PC)

Opció 2: No tenim encara codi a local, esta al github.

`git clone url_repositori` -> per a clonar a local el repositori del github.

(amb això descarreguem el que ja tenim al github)

`git add` -> afegir fitxer/s a un “paquet”

`git commit -m “missatge”` -> crear un primer commit “paquet” amb un missatge

`git push`



Exercici 3

- 1- Crea un repositori al teu github amb Readme.
- 2- Fes clone a una carpeta del teu ordinador
- 3- Afegeix un fitxer a aquesta carpeta.
- 4- Envia els canvis al repositori remot (add + commit + push)
- 5- Modifica el readme desde github.
- 6- Fes pull.

(Resolució conflictes)



Exercici 4

Treballem en equips! Farem 5 equips.

Un alumne de l'equip entra en aquest link: <https://classroom.github.com/a/r9BBVS9>

Això crearà un repositori amb uns fitxers predefinits.

Convida als altres alumnes de l'equip Settings -> Teams -> Add people.

La idea és que tots els membres del equip modifiquen els fitxers i resolguen els conflictes.





És una aplicació d'escriptori que ens permet gestionar GIT de manera **visual**.

Més còmode, sobretot per a visualitzar les branques.

<https://www.sourcetreeapp.com/>

Fer l'exercici 3 amb SourceTree.



SourceTree

Commit, pull,
push

The screenshot shows the SourceTree application interface for a repository named "sessio070722-IriveraEjercicios (Git)". The top toolbar includes icons for "Anotar" (Commit), "Recibir" (Pull), "Push", "Recuperar" (Fetch), "Rama" (Branch), "Integrar" (Merge), and "Reservado" (Stash). The left sidebar contains navigation options: "Historial" (History), "Buscar" (Search), "RAMAS" (Branches) with "main" selected, "ETIQUETAS" (Tags), "REMOTOS" (Remotes) with "origin", "RESERVADOS" (Stashed), "SUBMODULOS" (Submodules), and "SUBARBOLES" (Subtrees).

The main area displays a commit history table with columns: "Gráfica", "Descripción", "Anotar", "Autor", and "Fecha". The commits are listed in descending order of age.

Gráfica	Descripción	Anotar	Autor	Fecha
	Upload exercise files	dc21065	Iriverauab <8270...	hoy, 8:32
	Delete git_ex4.txt	5bf0bbb	Iriverauab <8270...	hoy, 8:32
	Update README.md	04b6d64	Iriverauab <8270...	hoy, 8:30
	Update README.md	0936358	Iriverauab <8270...	hoy, 8:30
	Update README.md	2db3435	IriveraEjercicios...	ayer, 23:23
	canvis desde local	ec9c3d9	Laura Rivera <o...	ayer, 23:24
	Initial commit	cca8519	github-classroo...	ayer, 23:11

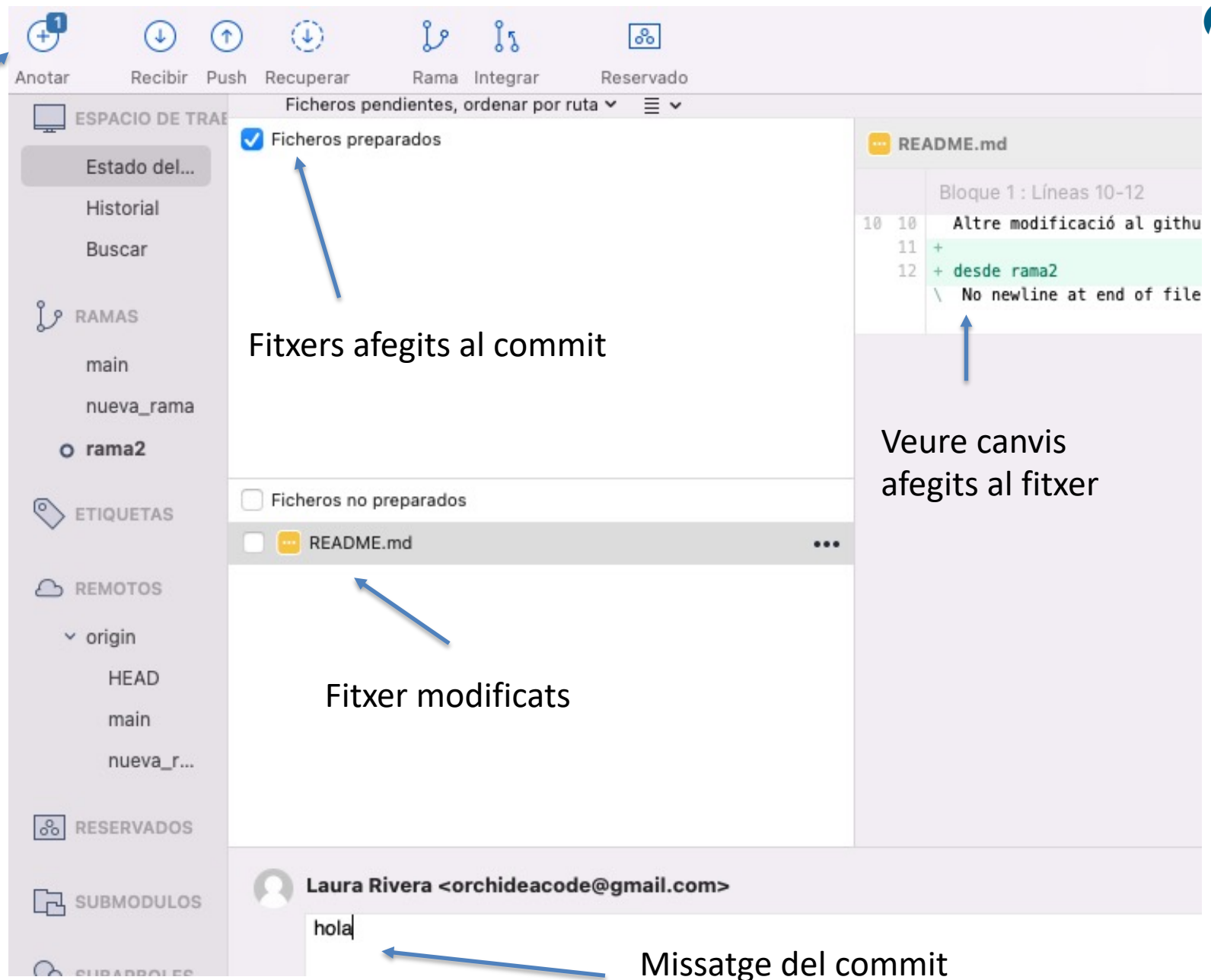
Below the commit history, the "canvis desde local" commit is selected, and the "README.md" file is shown in the diff view. The diff view displays the changes made in the selected commit, highlighting the addition of the line "adios" in the README.md file.

Canvis

Commits ja realitzats

Canvis en el fitxer en el commit seleccionat

SourceTree Afegir canvis



The screenshot shows the SourceTree application interface with several annotations:

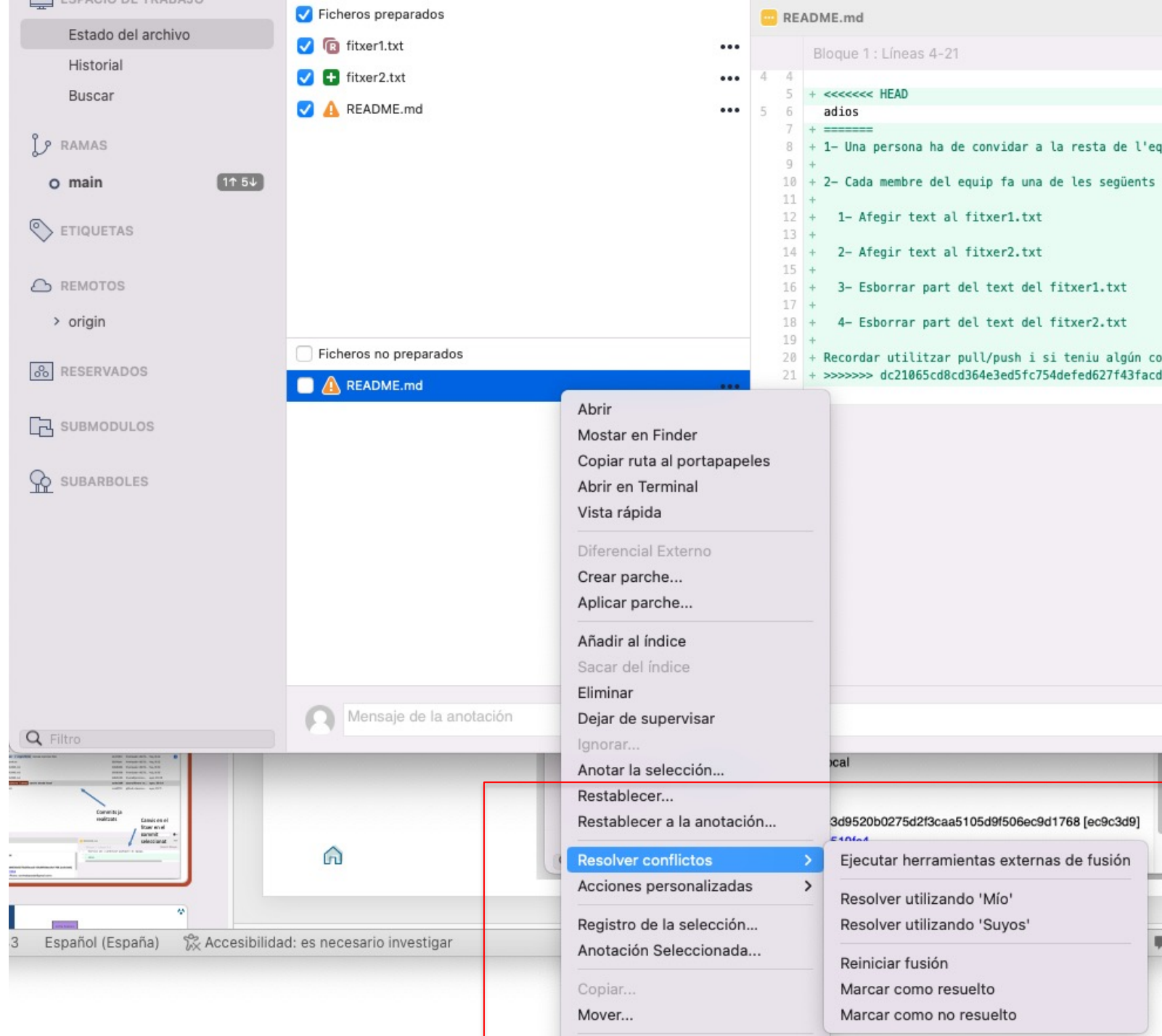
- Anotar**: An arrow points to the '+' icon in the top toolbar.
- Fitxers afegits al commit**: An arrow points to the 'Ficheros preparados' (Files ready for commit) section in the central pane.
- Fitxer modificat**: An arrow points to the 'README.md' file listed under 'Ficheros no preparados' (Files not ready for commit).
- Veure canvis afegits al fitxer**: An arrow points to the diff view for 'README.md' on the right, showing changes from 'rama2'.
- Missatge del commit**: An arrow points to the commit message input field at the bottom, which contains the text 'hola'.

The interface includes a sidebar on the left with sections: ESPACIO DE TRABAJO, RAMAS (main, nueva_rama, rama2), ETIQUETAS, REMOTOS (origin, HEAD, main, nueva_r...), RESERVADOS, and SUBMODULOS. The top toolbar contains icons for Anotar, Recibir, Push, Recuperar, Rama, Integrar, and Reservado.

Conflictes en SourceTree

Permet escollir:

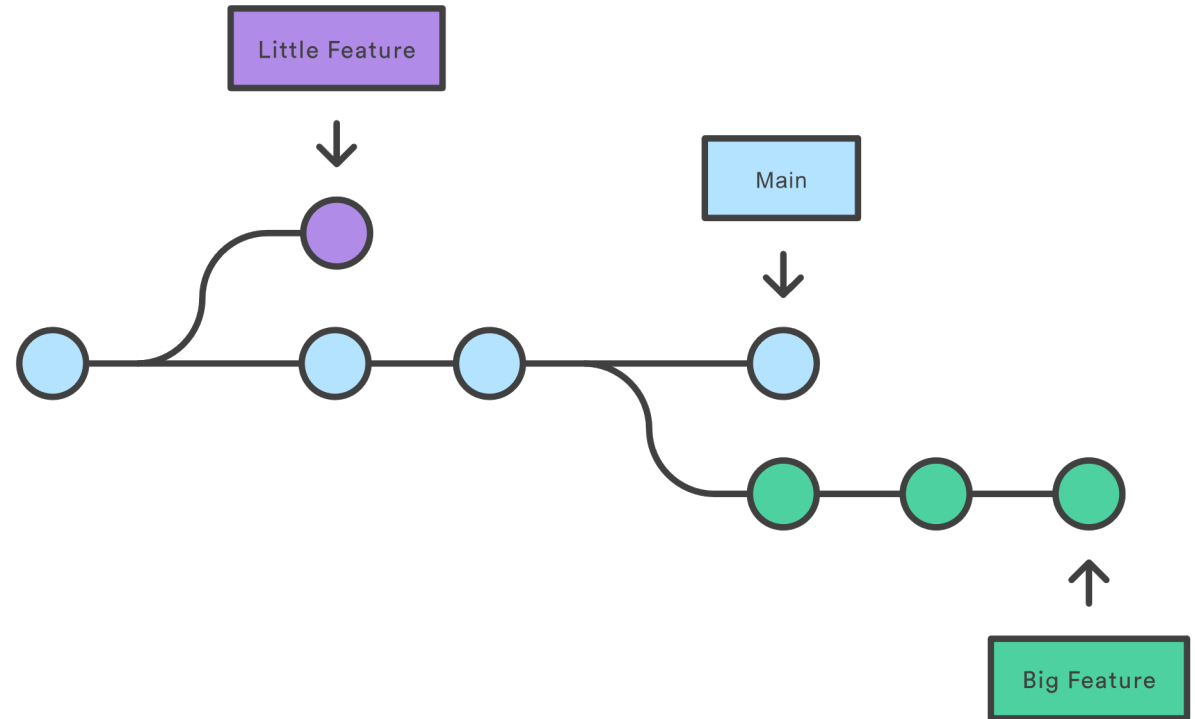
- El codi local "Mio"
- El codi remot "Suyos"
- Solucionar manualment i seleccionar "Marcar como resuelto"



Branques (Branch)

Git ens permet dividir la feina en branques, de manera que tenim diferents versions de manera simultània.

Aquestes branques després es poden unir.



Món laboral

A les empreses és força comú utilitzar una branca per cada funcionalitat o tasca.

Per exemple: Crear la funcionalitat de Login, arreglar un error en la comprovació de dades...

Branques per línia comandes

git branch -> veure les branques que tenim

git **checkout** nom_branca -> canviar de branca

git checkout -b nom_branca -> crear nova branca i canviar a aquesta

Treballar sobre aquesta branca i pujar els canvis.

Per a fer el push, ens demanarà més informació per a crear la branca a remot:

git push --set-upstream origin nueva_rama



Credencials

<https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/creating-a-personal-access-token#creating-a-token>

Crear el token i utilitzar-ho com a password si ho demana.



Integrar els canvis a la branca principal

git **checkout main** -> tornem a la branca principal (main/master)

git **merge nom_branca** -> integrem els canvis de la branca en la que estem (main)

Si hi ha algun conflicte. Resoldre com ja hem vist.



Exercici 5

- Crea un repositori al github amb readme.
- Fer clone per linea de comandes.
- Crea una branca.
- Des de la branca modifica el fitxer readme.
- Comprova al github els canvis
- Integra (merge) els canvis d'aquesta branca a la principal (main/master).



Branques al SourceTree

(Afegir un projecte local)

Nuevo -> Añadir
repositorio local existente

Selecciona el repositori de
l'exercici anterior.

Permet crear una branca
a partir de la que estem
(negreta)

Permet integrar
una branca a dins
de la que estem

Gràfica	Descripción
	nueva_ra... origin/nueva_ra... canvi a la branca
	origin/main origin/HEAD main arreglat el conflicte
	Update README.md
	modificat
	Update README.md
	afegit text al readme
	Initial commit

Branques que tenim

Exercici 6

- Selecciona la branca main (doble-click)
- Crea una nova branca (branca2)
- Modifica el fitxer
- Ara torna a la branca main
- Modifica també el fitxer
- Integra la branca2 en la actual (main)
- Soluciona el conflicte.



Exercici final

Treballarem tots junts!

1. Us convidaré a tots a un repositori.
2. Un membre de l'equip crea una branca "equip1", "equip2".
3. Cada membre de l'equip tindreu unes modificacions a fer en una branca dins de la branca del seu equip.
4. Integrar primer els canvis a la branca d'equip.
5. Integrar els canvis a la branca "main" (junts).

https://github.com/lriverauab/git_final.git

gitignore

Permet indicar quins fitxers / carpetes GIT hauria d'ignorar i no afegir al repositori.

Normalment esta a l'arrel del projecte [.gitignore](#).

Es pot afegir al crear el repositori al Github.

Per exemple:

.DS_Store -> per a que ignori els fitxers de Mac

*.txt -> utilitzem * per indicar que volem que ignori tots els fitxers amb extensió txt

Nom_carpeta -> podem ignorar carpetes senceres.

Útil per ignorar fitxers de configuració que depenen de la màquina.

Barcel
Activ



Ajuntament de
Barcelona



Barcelona
Activa

barcelonactiva.cat