

0487: Entorns de desenvolupament

2.3_ACT01 : Control de versions - GIT

Objectius:

RA3 Verifica el funcionament de programes dissenyant i realitzant proves.

- 1.a. Identifica els diferents tipus de proves.
- 1.b. Defineix casos de prova.
- 1.c. Identifica les eines de depuració i prova d'aplicacions ofertes per l'entorn de desenvolupament.
- 1.d. Utilitza eines de depuració per definir punts de ruptura i seguiment.
- 1.e. Utilitza les eines de depuració per examinar i modificar el comportament d'un programa en temps d'execució.
- 1.f. Efectua proves unitàries de classes i funcions.
- 1.g. Implementa proves automàtiques.
- 1.h. Documenta les incidències detectades

RA4 Optimitza codi emprant les eines disponibles en l'entorn de desenvolupament.

- 2.a. Identifica els patrons de refacció més usuals.
- 2.b. Elabora les proves associades a la refacció.
- 2.c. Revisa el codi font utilitzant un analitzador de codi.
- 2.d. Identifica les possibilitats de configuració d'un analitzador de codi.
- 2.e. Aplica patrons de refacció amb les eines que proporciona l'entorn de desenvolupament.
- 2.f. Realitza el control de versions integrat en l'entorn de desenvolupament.
- 2.g. Utilitza eines de l'entorn de desenvolupament per documentar les classes

Requisits:

- ✓ IDE

Entrega:

- ✓ Arxiu .zip/.rar que inclogui els arxius creats per resoldre cada un dels exercicis.
- ✓ El nom de l'arxiu d'entrega, ha de seguir el següent format: **CognomNom_0487_ACT09**

Desenvolupament del Projecte Nadalenc Amb Git

Objectius

- Aprendre i practicar els comandaments bàsics de Git.
- Entendre el flux de treball en un entorn de control de versions.

Tasques

1. Configuració Inicial de Git

- Executa `git config` per establir el teu nom i correu electrònic.
- Obre l'editor de configuració i revisa les configuracions.

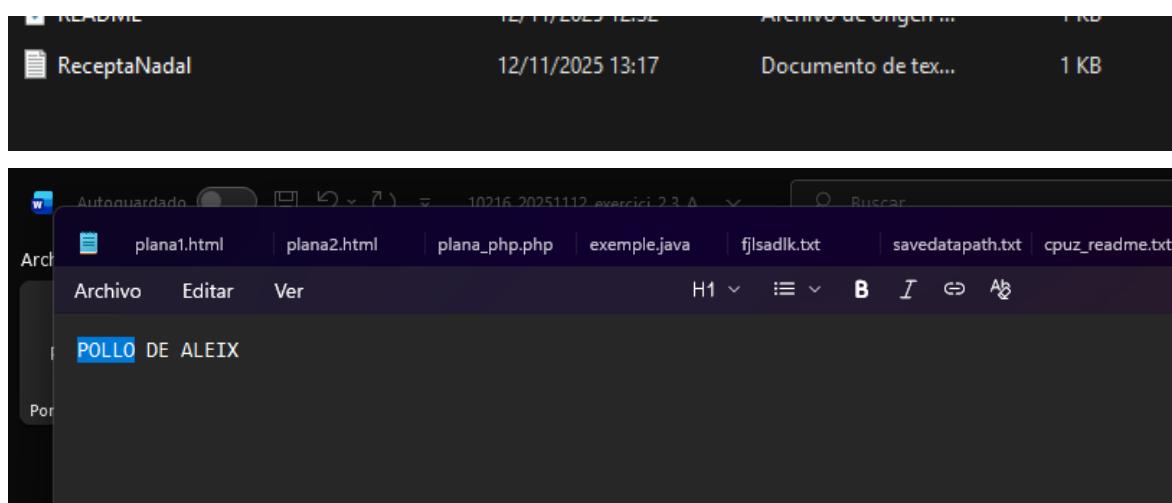
2. Inici del Repositori Git

- Crea un directori anomenat ProjecteNadal ○ Iniciar un nou repositori Git.

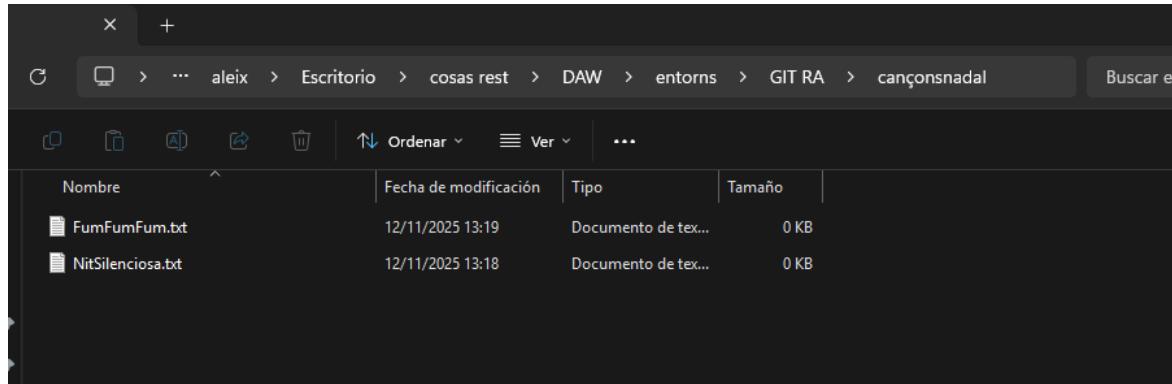
```
C:\Users\aleix>cd C:\Users\aleix\Desktop\cosas rest\DAW\entorns\GIT RA  
C:\Users\aleix\Desktop\cosas rest\DAW\entorns\GIT RA>git init  
Initialized empty Git repository in C:/Users/aleix/Desktop/cosas rest/DAW/entorns/GIT RA/.git/
```

3. Creació dels Fitxers del Projecte

- Crea un fitxer de text `git config`
- text anomenat `ReceptaNadal.txt` i escriu una recepta fictícia per a un plat nadalenc.



- Finalment, crea un directori anomenat [CançonsNadal](#) i dins d'aquest, crea diversos fitxers de text, cada un anomenat per una cançó nadalenca diferent (per exemple, [NitSilenciosa.txt](#), [FumFumFum.txt](#)). Escriu les cançons a dins.



4. Estat i Afegiment de Fitxers

- Revisa l'estat dels fitxers (git status)

```
C:\Users\aleix\Desktop\cosas rest\DAW\entorns\GIT RA>git status
On branch main
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    ACT 1/
    Git-2.51.2-64-bit.exe
    "can\303\247onsnadal/"
      inicio.png
```

- Afegeix els fitxers al seguiment de Git un a un menys el directori de [CançonsNadal](#) que ha de ser tot de cop amb els fitxers. (git add filename.txt)

```
C:\Users\aleix\Desktop\cosas rest\DAW\entorns\GIT RA>git add ReceptaNadal.txt
warning: in the working copy of 'ReceptaNadal.txt', LF will be replaced by CRLF the next time Git touches it
C:\Users\aleix\Desktop\cosas rest\DAW\entorns\GIT RA>git commit -m "Añadir ReceptaNadal.txt"
[main 31b3c6d] Añadir ReceptaNadal.txt
 1 file changed, 3 insertions(+)
 create mode 100644 ReceptaNadal.txt
```

- Fes els commits de cada part. (git commit -m filename.txt)

```
C:\Users\aleix\Desktop\cosas rest\DAW\entorns\GIT RA>git add cançonsnadal/
C:\Users\aleix\Desktop\cosas rest\DAW\entorns\GIT RA>git commit -m "Añadir cançonsnadal"
[main c018bcb] Añadir cançonsnadal
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 "can\303\247onsnadal\FumFumFum.txt.txt"
 create mode 100644 "can\303\247onsnadal\NitSilenciosa.txt.txt"
```

5. Modificacions i Correccions

- Realitza algunes modificacions als fitxers.

```
C:\Users\aleix\Desktop\cosas rest\DAW\entorns\GIT RA>git status
On branch main
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    ACT 1/
    Git-2.51.2-64-bit.exe
    inicio.png

nothing added to commit but untracked files present (use "git add" to track)

C:\Users\aleix\Desktop\cosas rest\DAW\entorns\GIT RA>git diff
```

- Utilitza la comanda per descartar els canvis per descartar canvis no desitjats. (git restore filename.txt)

```
C:\Users\aleix\Desktop\cosas rest\DAW\entorns\GIT RA>git diff
C:\Users\aleix\Desktop\cosas rest\DAW\entorns\GIT RA>git restore ReceptaNadal.txt
```

- Revisa l'històrial de commits. (git log)

```
C:\Users\aleix\Desktop\cosas rest\DAW\entorns\GIT RA>git log
commit c018bcbc1f901692191305b5564d5a966a83ad62 (HEAD --> main)
Author: aleix <almaar@jviladoms.cat>
Date:   Wed Nov 12 13:24:40 2025 +0100

  Añadir cançonsnadal

commit 31b3c6dabac8a6697f00ddad3f335ab5308aa782
Author: aleix <almaar@jviladoms.cat>
Date:   Wed Nov 12 13:22:55 2025 +0100

  Añadir ReceptaNadal.txt

commit 424aafc03398cc6ea109b33ea8f3fcf1ba9e896c
Author: aleix <almaar@jviladoms.cat>
Date:   Wed Nov 12 12:35:45 2025 +0100

  first commit
```

6. Comparació de Canvis

- Fes més canvis i utilitza la comanda per veure les diferències. (git diff)

```
C:\Users\aleix>git diff

C:\Users\aleix>git add receptanadal.txt
fatal: pathspec 'receptanadal.txt' did not match any files

C:\Users\aleix>git add ReceptaNadal.txt
fatal: pathspec 'ReceptaNadal.txt' did not match any files

C:\Users\aleix>cd C:\Users\aleix\Desktop\cosas rest\DAW\entorns\GIT RA

C:\Users\aleix\Desktop\cosas rest\DAW\entorns\GIT RA>git add ReceptaNada
l.txt

C:\Users\aleix\Desktop\cosas rest\DAW\entorns\GIT RA>
```

7. Correcció de Commit

- Fes un commit amb un missatge incorrecte

```
C:\Users\aleix\Desktop\cosas rest\DAW\entorns\GIT RA>git commit -m "Mens
aje mal" --allow-empty
[main a8bf232] Mensaje mal

C:\Users\aleix\Desktop\cosas rest\DAW\entorns\GIT RA>
```

- Corregeix aquest missatge. (git commit --amend -m "nuevo mensaje")

```
C:\Users\aleix\Desktop\cosas rest\DAW\entorns\GIT RA>git commit -m "Mens
aje mal" --allow-empty
[main a8bf232] Mensaje mal

C:\Users\aleix\Desktop\cosas rest\DAW\entorns\GIT RA>git commit --amend
-m "Mensaje corregido y más claro"
```

8. Ús de Reset

- Investiga i practica l'ús de `git reset` amb les diferents opcions (`--soft`, `--mixed`, `--hard`) usant tant `HEAD^` com un hash de commit específic. (`git reset --soft HEAD~1`)

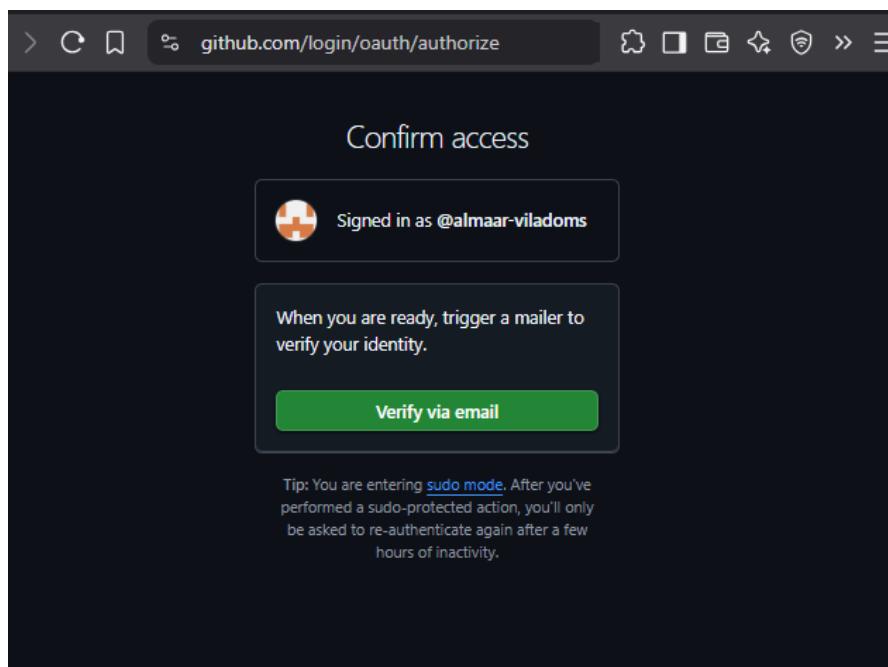
9. Branques

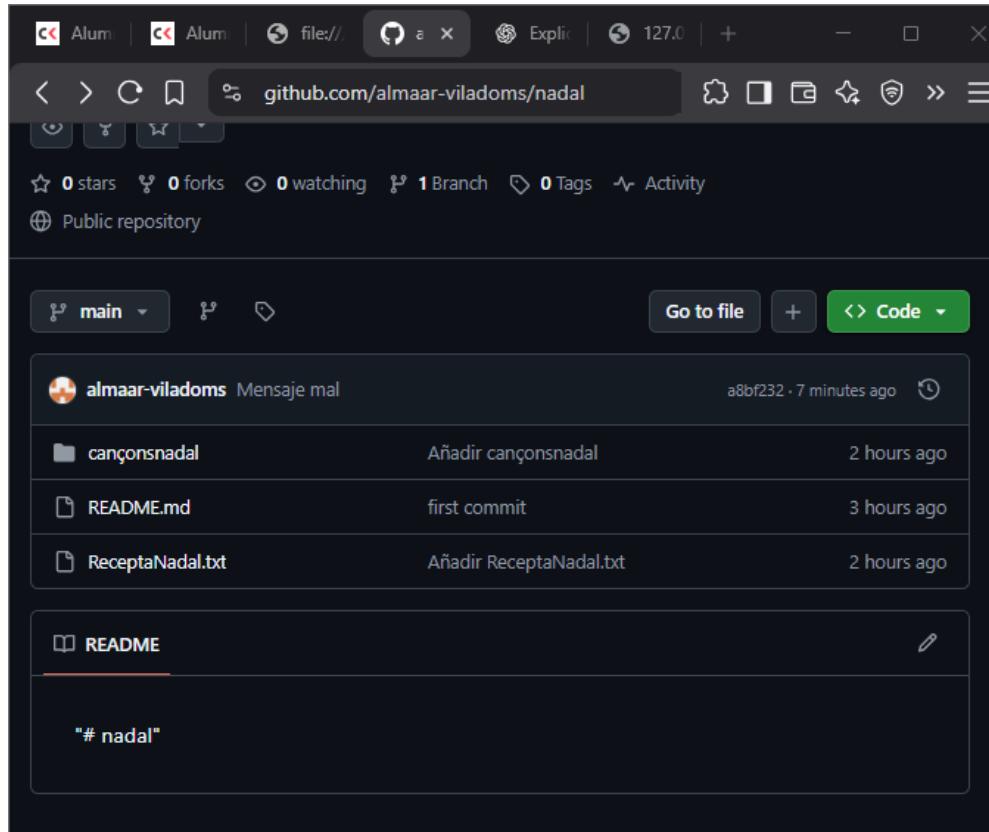
- Investiga i mostra l'ús de les branques, crea una branca, fes una modificació i ajunta-la amb la principal novament.

10. GitHub

- Connectat amb el GitHub del núvol i puja el teu projecte públic i comparteix-lo

```
C:\Users\aleix\Desktop\cosas rest\DAW\entorns\GIT RA>git remote add orig  
in https://github.com/almaar-viladoms/nadal.git  
  
C:\Users\aleix\Desktop\cosas rest\DAW\entorns\GIT RA>git branch -M main  
  
C:\Users\aleix\Desktop\cosas rest\DAW\entorns\GIT RA>git push -u origin  
main  
info: please complete authentication in your browser...
```





ALEIX GITHUB

INSTRUCCIONS AJUDA

1. **Crear una nueva rama:** Para crear una nueva rama, utiliza el comando: bash
Copiar código

```
git branch nombre_de_la_rama
```

Este comando solo crea la rama, pero no te cambia a ella automáticamente. Para trabajar en esta nueva rama, debes cambiarte a ella con git checkout o git switch.
2. **Cambiar a una rama (checkout):** Para moverte a la nueva rama y empezar a trabajar en ella, usa:

```
git checkout nombre_de_la_rama
```

Alternativamente, desde Git 2.23, puedes usar git switch para cambiar de rama:

```
git switch nombre_de_la_rama
```
3. **Crear y cambiar a una nueva rama directamente:** Para crear una nueva rama y moverte a ella en un solo paso, puedes usar:

```
git checkout -b nombre_de_la_rama  
O bien: bash Copiar código git  
switch -c nombre_de_la_rama  
Esto crea la rama y te coloca automáticamente en ella.
```

4. Ver las ramas existentes

Para ver todas las ramas en tu repositorio, usa:

```
bash Copiar  
código git  
branch
```

Esto mostrará una lista de todas las ramas, con un asterisco (*) junto a la rama en la que estás trabajando actualmente.

5. Combinar ramas (merge)

Cuando termines de trabajar en una rama y quieras incorporar los cambios en otra rama (por ejemplo, la rama main), debes hacer un **merge**:

1. Cambia a la rama en la que quieras integrar los cambios (normalmente main): bash Copiar código git checkout main
2. Luego, combina la rama donde trabajaste usando el comando merge:
bash Copiar código
git merge nombre_de_la_rama

6. Eliminar una rama

Si ya no necesitas una rama, puedes eliminarla:

```
bash Copiar código  
git branch -d nombre_de_la_rama
```