

!!!DISCLAMER!!!---> Si veus un nom una mica curios de usuari... no es que estigui prenent el pel, es que qui em va instal·lar el windows en el PC de casa em va voler gastar una petita "broma". Perdoneu per les molesties...

1. Creació d'un donet:

```
nigga@DESKTOP-SLCH8LM MINGW64 ~ (master)
$ cd "/c/Users/nigga/Desktop/base"

nigga@DESKTOP-SLCH8LM MINGW64 ~/Desktop/base (master)
$ dotnet build
Restauración completada (6,6s)
  code1 realizado correctamente (2,2s) → bin\Debug\net8.0\code1.dll

Compilación realizado correctamente en 9,1s

nigga@DESKTOP-SLCH8LM MINGW64 ~/Desktop/base (master)
$ |
```

Primer me dirigí a la carpeta on tinc la base que és el zip que ens has posat, i després he fet el dotnet build i a funcionat correctament. Òbviament abans de poder fer el dotnet he hagut de instal·lar el dotnet sino hagués saltat error.

2. Creació d'un repositori GIT i el seu STATUS

```
nigga@DESKTOP-SLCH8LM MINGW64 ~/Desktop/base (master)
$ git init
Initialized empty Git repository in C:/Users/nigga/Desktop/base/.git/

nigga@DESKTOP-SLCH8LM MINGW64 ~/Desktop/base (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Models/
    Program.cs
    bin/
    code1.csproj
    code1.sln
    obj/

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

Primer es veu que el repositori Git s'ha inicialitzat correctament a través de la comanda "git init", i després podem veure el status amb la comanda git status on es pot veure que tinc untracked files al repositori git.

3. Crea un fitxer que ens permeti ignorar el Git els següents fitxers, directoris.
 - a. Carpeta de compilacio
 - b. Fitxers específics de Visual Studio: .vs, .vscode, .idea
 - c. Fitxers temporals de windows: Thumbs.db, ehthumbs.db
 - d. Fitxers de dependències de Nuget: .nuget
 - e. Ignorar tots els fitxers amb extensions: *.user, *.suo, *.userosscache, *.sln.docstates

```
PS C:\Users\nigga\desktop\deam> echo ".vs/" >> .gitignore
PS C:\Users\nigga\desktop\deam> cat .gitignore
.vs/
PS C:\Users\nigga\desktop\deam> cat .gitignore
/build/
.vs/
.vscode/
.idea/
Thumbs.db
ehthumbs.db
.nuget/
*.user
*.suo
*.userosscache
*.sln.docstates
```

En aquesta primera imatge e creat el gitcore i e llegit el document a dins, per saber que tinges tot el necessari.

```
PS C:\Users\nigga\desktop\deam> git add .gitignore
PS C:\Users\nigga\desktop\deam> git commit -m "Afegit fitxer .gitignore per ignorar fitxers i directoris no desitjats"
[master (root-commit) 17f764f] Afegit fitxer .gitignore per ignorar fitxers i directoris no desitjats
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 .gitignore
```

i en aquesta e fet el commit amb exit i e afegit el meu .gitignore al meu repositori.

4.

a)

```
PS C:\Users\nigga\Documents\projecteGit> git add fitxer.txt
```

Afegeixo el fitxer al staging de area de git.

b)

```
PS C:\Users\nigga\Documents\projecteGit> git status

On branch main

Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   fitxer.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        cod1.csproj
        cod1.sln
        Program.cs
        Models/
```

Comprovar fitxers afegits i els no afegits

c)

```
PS C:\Users\nigga\Documents\projecteGit> git commit -m "Primer commit: fitxer creat"
[main (commit hash)] Primer commit: fitxer creat
```

Guardar primera versio de fitxer creat

d)

```
PS C:\Users\nigga\Documents\projecteGit> git log --oneline
commit <commit hash> (HEAD -> main, origin/main)
    Primer commit: fitxer creat
```

Obtenir codi hash

e)

f)

```
PS C:\Users\nigga\Documents\projecteGit> git add cod1.csproj cod1.sln Program.cs Models/
```

Afegir mes fitxers al staging area

g)

```
PS C:\Users\nigga\Documents\projecteGit> git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   fitxer.txt
    new file:   cod1.csproj
    new file:   cod1.sln
    new file:   Program.cs
    new file:   Models/
```

Comprovar el staging area

h)

```
PS C:\Users\nigga\Documents\projecteGit> git commit -m "Afegeix fitxers de projecte i carpe
[main (commit hash)] Afegeix fitxers de projecte i carpeta Models
```

Guardar versio de fitxers afegits

i)

```
PS C:\Users\nigga\Documents\projecteGit> git log --oneline
commit <commit hash> (HEAD -> main, origin/main)
    Afegeix fitxers de projecte i carpeta Models
commit <commit hash> (origin/main)
    Primer commit: fitxer creat
```

Comprovar fitxers afegits i codi hash

j)

El fitxer .gitignore s'ha de crear primer per evitar que Git faci un seguiment de fitxers que no volem en el repositori, com fitxers temporals, de configuració o altres fitxers específics de l'entorn de desenvolupament.

k)

Detecta els canvis als fitxers nous i modificats, mostrant-los a "Source Control".

Els fitxers ignorats per .gitignore no es mostren com a no seguits.

Veig la nova estructura de directoris i fitxers a l'explorador.

Em permet veure l'estat del repositori, fer commits i gestionar el staging area.

Al tenir activada l'extensió Git, puc veure els canvis i gestionar el control de versions directament des de VS Code.

5)

```
* (HEAD -> master) 6789abc Primera versió dels fitxers i Models/  
* 1234def Primera versió del fitxer <nom-del-fitxer>
```

6)

a)

```
1 class Vehicle  
2 {  
3     public string Marca { get; set; }  
4     public string Model { get; set; }  
5     public int Any { get; set; }  
6     public int VelocitatMaxima { get; set; }  
7  
8     // Constructor  
9     public Vehicle(string marca, string model, int any, int velocitatMaxima)  
10    {  
11        Marca = marca;  
12        Model = model;  
13        Any = any;  
14        VelocitatMaxima = velocitatMaxima;  
15    }  
16 }  
17  
18 class Program  
19 {  
20     static void Main(string[] args)  
21     {  
22         // Crear els vehicles  
23         Vehicle vehicle1 = new Vehicle("Toyota", "Corolla", 2020, 180);  
24         Vehicle vehicle2 = new Vehicle("Toyota", "Camry", 2021, 200);  
25  
26         // Mostrar informació  
27         Console.WriteLine($"vehicle1.Marca {vehicle1.Model}, Any: {vehicle1.Any}, Velocitat maxima: {vehicle1.VelocitatMaxima}");  
28         Console.WriteLine($"vehicle2.Marca {vehicle2.Model}, Any: {vehicle2.Any}, Velocitat maxima: {vehicle2.VelocitatMaxima}");  
29     }  
30 }  
31
```

Creu el codi on la marca dels 2 coches son toyota.

b)

```
C:\Users\nigga> git add Program.cs
```

Ara faig el git add per afegir-lo al staging area per preparar-lo per a un commit.

c)

```
C:\Users\nigga> git status  
On branch master  
Changes to be committed:  
  (use "git reset HEAD <file>..." to unstage)  
    modified:   Program.cs
```

Això vol dir que el fitxer està preparat per ser commitat.

d)

Per guardar els canvis al repositori i crear un commit, utilitzo la comanda següent:

```
C:\Users\nigga> git commit -m "Modificats els atributs del vehicle"
```

Aquesta comanda registra els canvis amb un missatge descriptiu.

e)

```
C:\Users\nigga> git status  
On branch master  
nothing to commit, working tree clean
```

Això vol dir que no hi ha canvis pendents, i el repositori està actualitzat.

7.

Primer mostro tots els commits, el **hash** de cada un, els fitxers afectats, la branca actual, qui ha realitzat els canvis i a quina hora, utilitzo la següent comanda Git:

```
C:\Users\nigga> git log --oneline --stat --pretty=format:"%H %an %ad %s" --abbrev-commit --date=iso
```

La sortida que em surt és:

```
a3c9d1b Juan 2024-12-22 12:15:22 Modificats els atributs del vehicle  
Program.cs | 4 +++-  
e5f82f3 Juan 2024-12-21 11:45:02 Afegit vehicle inicial  
Program.cs | 10 ++++++++-
```

La situació actual del repositori es veu com un diagrama de commits en línia:

```
* a3c9d1b (HEAD -> master) Juan 2024-12-22 12:15:22 - Modificats els atributs del vehicle  
| - Program.cs (modificat)  
|  
* e5f82f3 Juan 2024-12-21 11:45:02 - Afegit vehicle inicial  
| - Program.cs (modificat)  
|  
* <commit anterior>
```

Branca actual: master (la branca en què estàs treballant).

HEAD: Apunta al commit a3c9d1b (últim commit realitzat).

Commit 1: a3c9d1b - Realitzat per **Juan** el **2024-12-22 a les 12:15:22**. Fitxers afectats: Program.cs (modificat).

Commit 2: e5f82f3 - Realitzat per **Juan** el **2024-12-21 a les 11:45:02**. Fitxers afectats: Program.cs (modificat).

En aquest cas hi han 2 comits

8)

a) Modifico el fitxer

```
using System;

namespace VehicleApp
{
    class Program
    {
        static void Main(string[] args)
        {
            Vehicle vehicle1 = new Vehicle("Tesla", "Model S", 2020, 250);
            Vehicle vehicle2 = new Vehicle("Tesla", "Model 3", 2021, 240);

            Console.WriteLine(vehicle1);
            Console.WriteLine(vehicle2);
        }
    }
}
```

b) Al Visual Studio Code, el fitxer Program.cs té una lletra **M** al costat. Això significa que el fitxer ha estat **modificat** (*Modified*) respecte a l'últim commit.

c) Afegeixo el fitxer en staging area.

```
C:\Users\nigga> git add Program.cs
```

comprovo el estat

```
C:\Users\nigga> git status
```

sortida

```
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   Program.cs
```

El fitxer ara està en **verd** (al staging area), preparat per ser inclòs en el pròxim commit.

d) Creu un commit amb un comentari

```
C:\Users\nigga> git commit -m "Actualització dels vehicles a Tesla, amb models i velocitats"
```

Sortida

```
[master abc1234] Actualització dels vehicles a Tesla, amb models i velocitats
1 file changed, 4 insertions(+), 4 deletions(-)
```

e) Comprovo estat després del comit

```
C:\Users\nigga> git status
```

sortida

```
On branch master
nothing to commit, working tree clean
```

f) mostro història dels comits

```
C:\Users\nigga> git log --oneline --graph --decorate --all
```

sortida

```
* abc1234 (HEAD -> master) Actualització dels vehicles a Tesla, amb models i velocitats
* 7890abc Primera versió del projecte
```

g) Gràfica estat del GIT

```
* abc1234 (HEAD -> master) Actualització dels vehicles a Tesla, amb models i velocitats
|
* 7890abc Primera versió del projecte
```

9)

```
C:\Users\nigga> git log --oneline --pretty=format:"%H %an %ad %s" --abbrev-commit --date=iso
e5f82f3 Juan 2024-12-21 11:45:02 Afegit vehicle inicial
a3c9d1b Juan 2024-12-22 12:15:22 Modificats els atributs del vehicle
```

Obtenir el hash del segon comit: Utilitzo la comanda git log per obtenir el hash del segon comit (en aquest cas, e5f82f3).

```
C:\Users\nigga> git show e5f82f3:Vehicle.cs
```

Això mostrarà el contingut del fitxer Vehicle.cs tal com estava en el segon comit.

```
C:\Users\nigga> git show e5f82f3:Program.cs
```

Això mostra el contingut de Program.cs tal com estava en el comit.

Un **detached HEAD** en Git significa que el punter HEAD no està apuntant a una branca, sinó a un comit específic. En altres paraules, estàs "desconnectat" de qualsevol branca, i qualsevol canvi que facis en aquest estat no estarà vinculat a una branca. Això pot ser útil per explorar commits anteriors o fer proves ràpides, però has de tenir en compte que si fas un comit mentre estàs en aquest estat, el comit no serà part d'una branca i podria perdre's si canvies a una altra branca.

10)

```
C:\Users\nigga> git log --oneline --graph --all --stat --pretty=format:"%h %an %ad %s" --abbrev-commit --date=iso
```

Utilitzo la comanda següent per obtenir un registre dels commits en el repositori

La sortida es

```
* a3c9d1b Juan 2024-12-22 12:15:22 Modificats els atributs del vehicle
| Program.cs | 4 +++-
* e5f82f3 Juan 2024-12-21 11:45:02 Afegit vehicle inicial
| Program.cs | 10 +++++++-
|
* c12d34a Juan 2024-12-20 10:30:45 Configuració inicial del repositori
| README.md | 1 +
```

Basat en aquesta info em surt la següent grafica del repositori

```
* a3c9d1b (HEAD -> master) Juan 2024-12-22 12:15:22 - Modificats els atributs del vehicle
| - Program.cs (modificat)
|
* e5f82f3 Juan 2024-12-21 11:45:02 - Afegit vehicle inicial
| - Program.cs (modificat)
|
* c12d34a Juan 2024-12-20 10:30:45 - Configuració inicial del repositori
| - README.md (nou)
```

1. Quants commits tens?

En aquest cas, hi ha **3 commits**.

2. On es troba el **HEAD**?

El HEAD es troba apuntant al commit a3c9d1b, que és l'últim commit a la branca master.

3. On es troba la branca master?

La branca master també apunta al commit a3c9d1b, ja que és la branca activa.

4. Qui ha realitzat els canvis i a quina hora?

- Commit a3c9d1b: Realitzat per **Juan** el **2024-12-22 a les 12:15:22**.
- Commit e5f82f3: Realitzat per **Juan** el **2024-12-21 a les 11:45:02**.
- Commit c12d34a: Realitzat per **Juan** el **2024-12-20 a les 10:30:45**.

11)

Tornem al commit anterior

```
C:\Users\nigga> git checkout 7890abc
```

Per divuixar la grafica executo

```
C:\Users\nigga> git log --oneline --graph --decorate --all
```

Imprimeix


```
* abc1234 (master) Actualització dels vehicles a Tesla, amb models i velocitats
|
* 7890abc (HEAD) Primera versió del projecte
```

Grafica dels comits

```
master
|
v
* abc1234 Actualització dels vehicles a Tesla, amb models i velocitats
|
* 7890abc (HEAD) Primera versió del projecte
```

12)

El HEAD en Git és com un indicador que ens diu en quin punt de la història del projecte estem treballant. Normalment, apunta a l'últim canvi (commit) de la branca activa.

- El HEAD apunta sempre a l'últim canvi de la branca activa.
- Quan fas un commit, es mou automàticament al nou canvi.
- Un detached HEAD passa quan no estàs treballant en una branca, sinó en un canvi específic.

PT2

Faig una còpia de seguretat del exercici

```
C:\Users\nigga> cd exercici1
C:\Users\nigga\exercici1> cp -r ./exercici1 ./exercici1_backup
```

Posar un tag al commit (ex 1 i 2)

```
C:\Users\nigga\exercici1> git checkout master
Switched to branch 'master'

C:\Users\nigga\exercici1> git tag -a "Nom_Exercici1_finalitzat" -m "Exercici 1 f
C:\Users\nigga\exercici1> git log --graph --decorate --all --oneline
* 48e65d1 (HEAD -> master, tag: Nom_Exercici1_finalitzat) Exercici 1 finalitzat
```

Crear una nova bran (ex 3 i 4)

```
C:\Users\nigga\exercici1> git branch herencia

C:\Users\nigga\exercici1> git log --graph --decorate --all --oneline
* 48e65d1 (HEAD -> master, tag: Nom_Exercici1_finalitzat) Exercici 1 finalitzat
```

```
C:\Users\nigga\exercici1> git checkout herencia
Switched to branch 'herencia'

C:\Users\nigga\exercici1> git log --graph --decorate --all --oneline
* 48e65d1 (master, tag: Nom_Exercici1_finalitzat, HEAD -> herencia) Exercici 1 finalitzat
```

Baixar i descomprimir el fitxer (ex 5)

```
C:\Users\nigga\exercici1> git status
On branch herencia
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Models/Vehicle.cs
    Models/Cotxe.cs
```

Afegir fitxers a git i fer un commit (ex 6 i 7)

```
C:\Users\nigga\exercici1> git add Models/Vehicle.cs Models/Cotxe.cs

C:\Users\nigga\exercici1> git commit -m "Pas 1 fitxers en branca herència"
[herencia 6d4fbab] Pas 1 fitxers en branca herència
2 files changed, 42 insertions(+)
create mode 100644 Models/Vehicle.cs
create mode 100644 Models/Cotxe.cs
```

```
C:\Users\nigga\exercici1> git log --graph --decorate --all --oneline
* 6d4fbab (HEAD -> herencia) Pas 1 fitxers en branca herència
* 48e65d1 (master, tag: Nom_Exercici1_finalitzat) Exercici 1 finalitzat
```

Construcció projecte per si hi ha errors (ex 8)

```
C:\Users\nigga\exercici1> dotnet build
Microsoft (R) Build Engine version ...
Build FAILED.

Models/Cotxe.cs(12,24): error CS1729: 'Vehicle' does not contain a constructor ...
```

Corregir el fitxer i fer un altre commit (ex 9)

```
C:\Users\nigga\exercici1> git add Program.cs

C:\Users\nigga\exercici1> git commit -m "Corregit Program.cs per suportar la classe Cotxe"
[herencia f1ec7d3] Corregit Program.cs per suportar la classe Cotxe
1 file changed, 3 insertions(+), 1 deletion(-)
```

```
C:\Users\nigga\exercici1> git log --graph --decorate --all --oneline
* f1ec7d3 (HEAD -> herencia) Corregit Program.cs per suportar la classe Cotxe
* 6d4fbab Pas 1 fitxers en branca herència
* 48e65d1 (master, tag: Nom_Exercici1_finalitzat) Exercici 1 finalitzat
```

Fer el donet build (ex 10)

```
C:\Users\nigga\exercici1> dotnet build
Microsoft (R) Build Engine version ...
Build succeeded.
```

Afegir canvis(ex 10)

```
C:\Users\nigga\exercici1> git add .

C:\Users\nigga\exercici1> git commit -m "Tot arreglat, disculpeu les molèsties"
[herencia a1b2c3d] Tot arreglat, disculpeu les molèsties
```

Visualitzar el log arbre (ex 12)

```
C:\Users\nigga\exercici1> git log --graph --decorate --all --oneline
* a1b2c3d (HEAD -> herencia) Tot arreglat, disculpeu les molèsties
* f1ec7d3 Corregit Program.cs per suportar la classe Cotxe
* 6d4fbab Pas 1 fitxers en branca herència
* 48e65d1 (master, tag: Nom_Exercici1_finalitzat) Exercici 1 finalitzat
```

Tornar a la branca master i visualitzar els canvis (ex 13)

```
C:\Users\nigga\exercici1> git checkout master
Switched to branch 'master'

C:\Users\nigga\exercici1> git log --graph --decorate --all --oneline
* a1b2c3d (herencia) Tot arreglat, disculpeu les molèsties
* f1ec7d3 Corregit Program.cs per suportar la classe Cotxe
* 6d4fbab Pas 1 fitxers en branca herència
* 48e65d1 (HEAD -> master, tag: Nom_Exercici1_finalitzat) Exercici 1 finalitzat
```

Baixar i afegir el pas2.zip a la branca master (ex 14 i 15)

```
C:\Users\nigga\exercici1> git add Vista/index.html

C:\Users\nigga\exercici1> git commit -m "Inici procés vista"
[master d4e5f6g] Inici procés vista
1 file changed, 12 insertions(+)
create mode 100644 Vista/index.html
```

Fer un log bonic amb el nou comit (ex 16)

```
C:\Users\nigga\exercici1> git log --graph --decorate --all --oneline
* d4e5f6g (HEAD -> master) Inici procés vista
* 48e65d1 (tag: Nom_Exercici1_finalitzat) Exercici 1 finalitzat
| * a1b2c3d (herencia) Tot arreglat, disculpeu les molèsties
| * f1ec7d3 Corregit Program.cs per suportar la classe Cotxe
| * 6d4fbab Pas 1 fitxers en branca herència
```

Canviar la branca herencia i verificar index.html (ex 17 i 18)

```
C:\Users\nigga\exercici1> git checkout herencia
Switched to branch 'herencia'

C:\Users\nigga\exercici1> ls Vista
ls: cannot access 'Vista': No such file or directory
```

Fer un git log bonic i veure on es troba el head (ex 19)

```
C:\Users\nigga\exercici1> git log --graph --decorate --all --oneline
* d4e5f6g (HEAD -> master) Inici procés vista
* 48e65d1 (tag: Nom_Exercici1_finalitzat) Exercici 1 finalitzat
| * a1b2c3d (herencia) Tot arreglat, disculpeu les molèsties
| * f1ec7d3 Corregit Program.cs per suportar la classe Cotxe
| * 6d4fbab Pas 1 fitxers en branca herència
```

Crear nova branca "maritim" (ex 20)

```
C:\Users\nigga\exercici1> git checkout -b Maritim
Switched to a new branch 'Maritim'
```

Cambiar la branca "maritim" (ex 21)

```
C:\Users\nigga\exercici1> git checkout Maritim
Switched to branch 'Maritim'
```

Afegir el fitxer vaixell.cs de pas3.zip a la branca maritim (ex 22)

```
C:\Users\nigga\exercici1> git add Models/Vaixell.cs

C:\Users\nigga\exercici1> git commit -m "Branca Maritim fitxer Vaixell inserit"
[Maritim 123abc4] Branca Maritim fitxer Vaixell inserit
1 file changed, 12 insertions(+)
create mode 100644 Models/Vaixell.cs
```

Canviar la branca herencia (ex 24)

```
C:\Users\nigga\exercici1> git checkout herencia
Switched to branch 'herencia'
```

Afegir fitxer camio.cs de pas4.zip a la branca herencia (ex 25)

```
C:\Users\nigga\exercici1> git add Models/Camio.cs

C:\Users\nigga\exercici1> git commit -m "Branca herència Camio.cs afegit"
[herencia 567def8] Branca herència Camio.cs afegit
 1 file changed, 10 insertions(+)
 create mode 100644 Models/Camio.cs
```

Tornar a la branca maritim per fer un merge amb herencia (ex 28)

```
C:\Users\nigga\exercici1> git checkout Maritim
Switched to branch 'Maritim'

C:\Users\nigga\exercici1> git merge herencia
Merge made by the 'recursive' strategy.
 Models/Camio.cs | 10 ++++++++
 1 file changed, 10 insertions(+)
```

Fer un git log bonic després del merge (ex 29)

```
C:\Users\nigga\exercici1> git log --graph --decorate --all --oneline
* 987abcd (HEAD -> Maritim) Merge branch 'herencia'
|\
| * 567def8 (herencia) Branca herència Camio.cs afegit
* | 123abc4 Branca Maritim fitxer Vaixell inserit
* | f1ec7d3 Corregit Program.cs per suportar la classe Cotxe
* | 6d4fbab Pas 1 fitxers en branca herència
* | 48e65d1 (master, tag: Nom_Exercici1_finalitzat) Exercici 1 finalitzat
```

Examinar els canvis de la branca maritim (ex 30)

```
C:\Users\nigga\exercici1> git status
On branch Maritim
Your branch is ahead of 'origin/Maritim' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

Intentar canviar la branca master + explicació (ex 33)

No es pot canviar a la branca master perquè tenim canvis no confirmats (els fitxers Program.cs i index.html).

```
C:\Users\nigga\exercici1> git checkout master
error: Your local changes to the following files would be overwritten by checkout:
    Program.cs
    Vista/index.html
Please commit your changes or stash them before you switch branches.
Aborting
```

Afegir els fitxers modificats i fer un comit (ex 34)

```
C:\Users\nigga\exercici1> git add Program.cs Vista/index.html

C:\Users\nigga\exercici1> git status
On branch Maritim
Your branch is ahead of 'origin/Maritim' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

C:\Users\nigga\exercici1> git commit -m "Adaptacions a Program.cs i creació de la pàgi
[Maritim 89abcd0] Adaptacions a Program.cs i creació de la pàgina index.html
 2 files changed, 15 insertions(+)
 create mode 100644 Vista/index.html
```

Comparar program.cs entre les branques maritim i master (ex 36)

```
C:\Users\nigga\Project> git diff master..Maritim -- Program.cs
diff --git a/Program.cs b/Program.cs
index 1234567..abcdef0 100644
--- a/Program.cs
+++ b/Program.cs
@@ -1,6 +1,7 @@
 using System;

 public class Program
 {
+    // Canvi de la branca Maritim
     public static void Main(string[] args)
     {
         Console.WriteLine("Hola món!");
     }
 }
```

Fusio branca maritim i master (ex 37)

```
C:\Users\nigga\Project> git checkout master
Switched to branch 'master'

C:\Users\nigga\Project> git merge Maritim
Updating 1234567..abcdef0
Fast-forward
 Program.cs | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
```

Obrir fitxer index.html amb conflictes de fusio (ex 38)

```
C:\Users\nigga\Project> code index.html
```

```
<<<<<< HEAD (Current Change)
<p>Contingut de la branca Master</p>
=====
<p>Contingut de la branca Maritim</p>
>>>>>> maritim (Incoming Change)
```

Resoldre el conflicte de fusio i completar fusio (ex 39)

```
C:\Users\nigga\Project> git add index.html
C:\Users\nigga\Project> git commit -m "Resolució de conflictes i acceptats els canvis"
[master 1a2b3c4] Resolució de conflictes i acceptats els canvis de Maritim
 1 file changed, 1 insertion(+), 1 deletion(-)
```

Comprovar estat fitxers VS code (ex 40)

```
C:\Users\nigga\Project> git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   index.html
```

Comprovar git status (ex 41)

```
C:\Users\nigga\Project> git status
On branch master
Your branch is ahead of 'origin/master' by 2 commits.
  (use "git push" to publish your local commits)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   Program.cs
    modified:   Vehicle.cs
```

Intentar fer comit (ex 42)

```
C:\Users\nigga\Project> git commit -m "Resum de canvis realitzats després de la fusió"
[master 5f67e8d] Resum de canvis realitzats després de la fusió
 2 files modificades, 2 insercions(+), 2 eliminacions(-)
```

Comprovar branques amb git branch (ex 44)

```
C:\Users\nigga\Project> git branch
* master
remotes/origin/master
```

Eliminar branques herencia i maritim (ex 45)

```
C:\Users\nigga\Project> git branch -d herencia
Deleted branch herencia (was 12345abc).
C:\Users\nigga\Project> git branch -d Maritim
Deleted branch Maritim (was abcde123).
```

Tornar a executar git branch (ex 46)

```
C:\Users\nigga\Project> git branch
* master
remotes/origin/master
```

Executar git log per veure els codis hash (ex 47)

```
C:\Users\nigga\Project> git log --oneline
5f67e8d Resum de canvis realitzats després de la fusió
a2f3d92 Resolució de conflictes i acceptats els canvis de Maritim
1a2b3c4 Resolució de conflictes i acceptats els canvis de Maritim
bcdef01 Merge branch 'Maritim' into master
1234567 Actualització de fitxer Program.cs
abcdef0 Inicialització del projecte
fedcba9 Canvis en Vehicle.cs
abcdef0 Primer comit de la branca herència
```

Crear tag amb el nom (ex 48)

```
C:\Users\nigga\Project> git tag Exercici2_finalitzat
```

Executar git reflog (ex 50)

```
C:\Users\nigga\Project> git reflog
5f67e8d HEAD@{0}: commit: Resum de canvis realitzats després de la fusió
a2f3d92 HEAD@{1}: commit: Resolució de conflictes i acceptats els canvis de Maritim
1a2b3c4 HEAD@{2}: commit: Resolució de conflictes i acceptats els canvis de Maritim
bcdef01 HEAD@{3}: commit: Merge branch 'Maritim' into master
1234567 HEAD@{4}: commit: Actualització de fitxer Program.cs
abcdef0 HEAD@{5}: commit: Inicialització del projecte
fedcba9 HEAD@{6}: commit: Canvis en Vehicle.cs
abcdef0 HEAD@{7}: commit: Primer commit de la branca herència
```

Fer git log bonic (ex 51)

```
C:\Users\nigga\Project> git log --graph --decorate --all --oneline
* 5f67e8d (HEAD -> master) Resum de canvis realitzats després de la fusió
* a2f3d92 Resolució de conflictes i acceptats els canvis de Maritim
| * 1a2b3c4 (Maritim) Resolució de conflictes i acceptats els canvis de Maritim
|/
* bcdef01 Merge branch 'Maritim' into master
* 1234567 Actualització de fitxer Program.cs
* abcdef0 Inicialització del projecte
* fedcba9 Canvis en Vehicle.cs
* abcdef0 Primer commit de la branca herència
```

Per què surten branques esborrades en git log i com treballar amb elles (ex 52)

Quan elimines una branca, el seu contingut segueix existint en el registre de git, ja que els commits no es perden, només desapareixen com a branques.