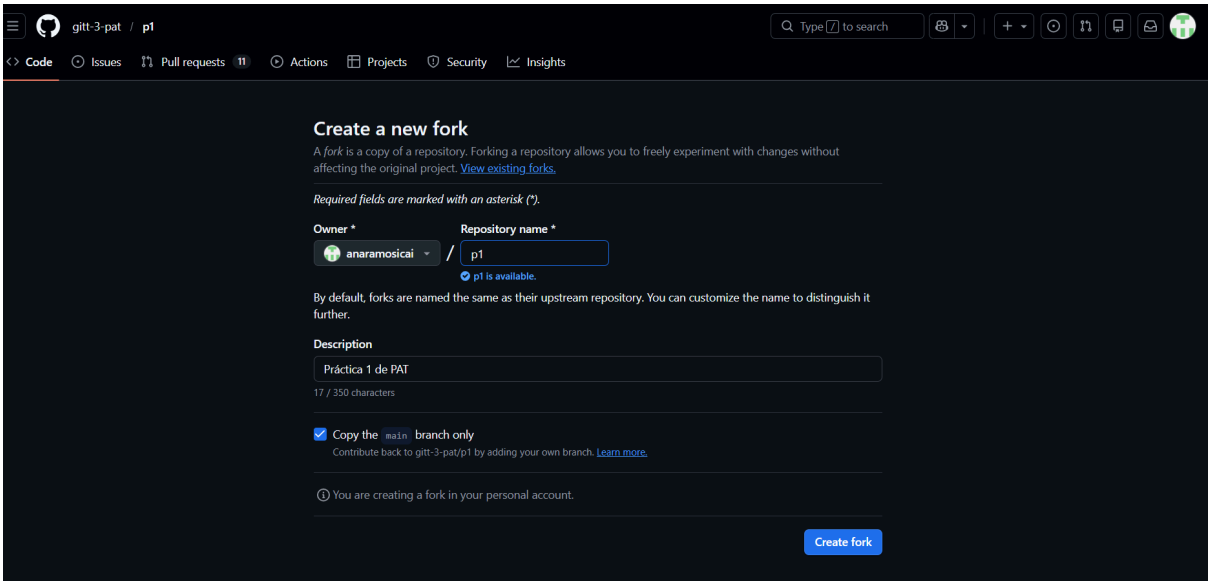
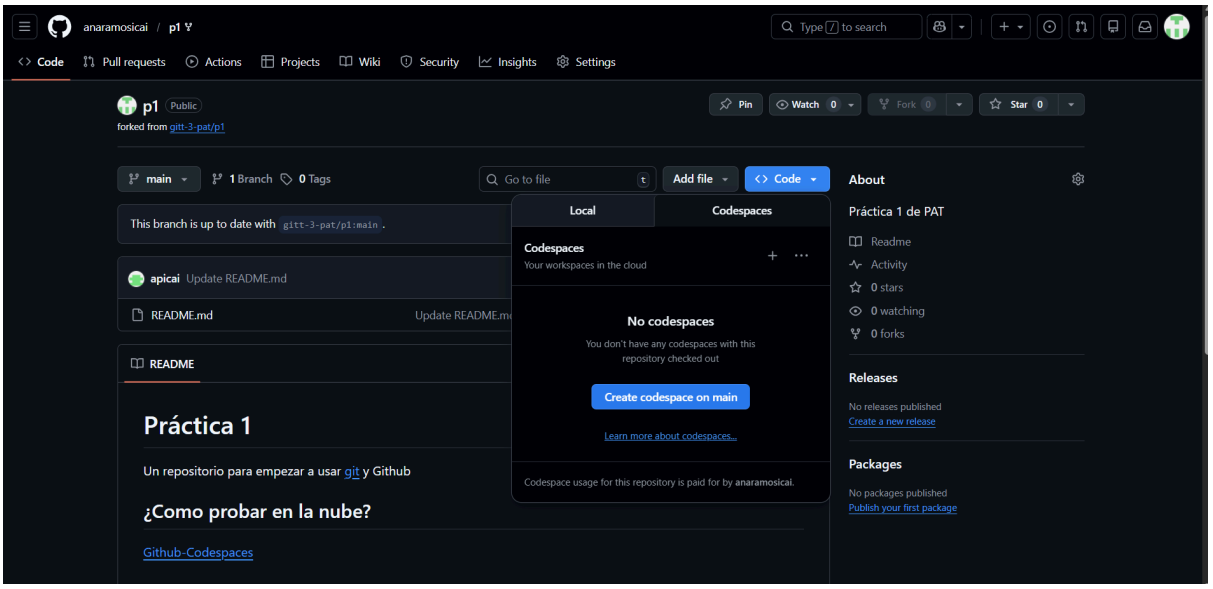
Práctica 1: Entorno de desarrollo

Desarrollo de la práctica

Git y Github:

Desde la cuenta de Github hacer un "fork" del repositorio: https://github.com/gitt-3-pat/p1.





A continuación, prueba los siguientes comandos sobre el anterior repositorio:

```
git clone https://github.com/gitt-3-pat/p1
git status
git add .
git commit -m "Subo cambios"
git push
```

```
@anaramosicai ➜/workspaces/p1 (main) $ git clone https://github.com/gitt-3-pat/p1
Cloning into 'p1'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (1/1), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 5 (from 1)
Receiving objects: 100% (6/6), done.
@anaramosicai ➜/workspaces/p1 (main) $ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        p1/

nothing added to commit but untracked files present (use "git add" to track)
@anaramosicai ➜/workspaces/p1 (main) $ git add .
warning: adding embedded git repository: p1
hint: You've added another git repository inside your current repository.
hint: Clones of the outer repository will not contain the contents of
hint: the embedded repository and will not know how to obtain it.
hint: If you meant to add a submodule, use:
hint:
hint:   git submodule add <url> p1
hint:
hint: If you added this path by mistake, you can remove it from the
hint: index with:
hint:
hint:   git rm --cached p1
hint:
hint: See "git help submodule" for more information.
hint: Disable this message with "git config set advice.addEmbeddedRepo false"
@anaramosicai ➜/workspaces/p1 (main) $
```

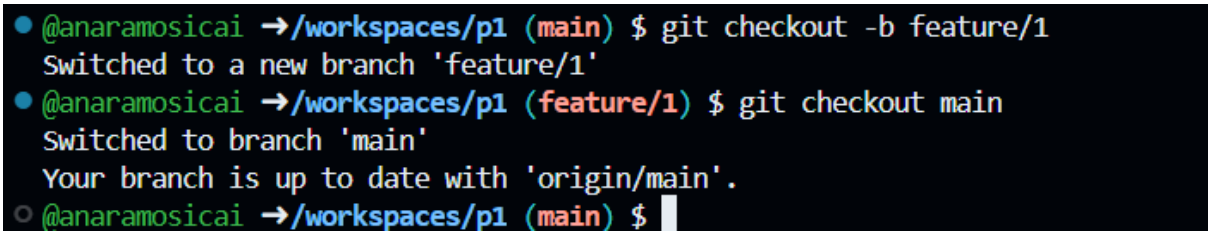
Como hay un .git dentro de la carpeta p1 creada, lo borro:

```
@anaramosicai ➜/workspaces/p1 (main) $ cd p1
@anaramosicai ➜/workspaces/p1/p1 (main) $ rm -rf .git
@anaramosicai ➜/workspaces/p1/p1 (main) $ cd ..
@anaramosicai ➜/workspaces/p1 (main) $ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   p1

@anaramosicai ➜/workspaces/p1 (main) $ git add .
@anaramosicai ➜/workspaces/p1 (main) $ git commit -m "Subo cambios"
[main cab022f] Subo cambios
 1 file changed, 1 insertion(+)
 create mode 160000 p1
@anaramosicai ➜/workspaces/p1 (main) $ git push origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 2 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 259 bytes | 259.00 KiB/s, done.
Total 2 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/anaramosicai/p1
   07720b5..cab022f  main -> main
@anaramosicai ➜/workspaces/p1 (main) $
```

```
git checkout -b feature/1
git checkout main
```



EXPLICACIÓN DE COMANDOS:

`git clone` - clonar - Hace una copia del repositorio mencionado al tuyo creado (el fork, nuestro caso)
`git status` - estado - Devuelve un "informe" sobre los cambios realizados en el directorio
`git add .` - añadir - Añade/Selecciona el repositorio actual
`git commit -m "` - cometer - Envía actualizaciones/cambias bajo un mensaje entrecomillado

`git checkout -b` - salir/revisar - Cambia a la rama que se cree (-b) con el nombre que indiques
`git checkout main` - `` ´´ - Cambia a la rama main