# Laboratorio #4

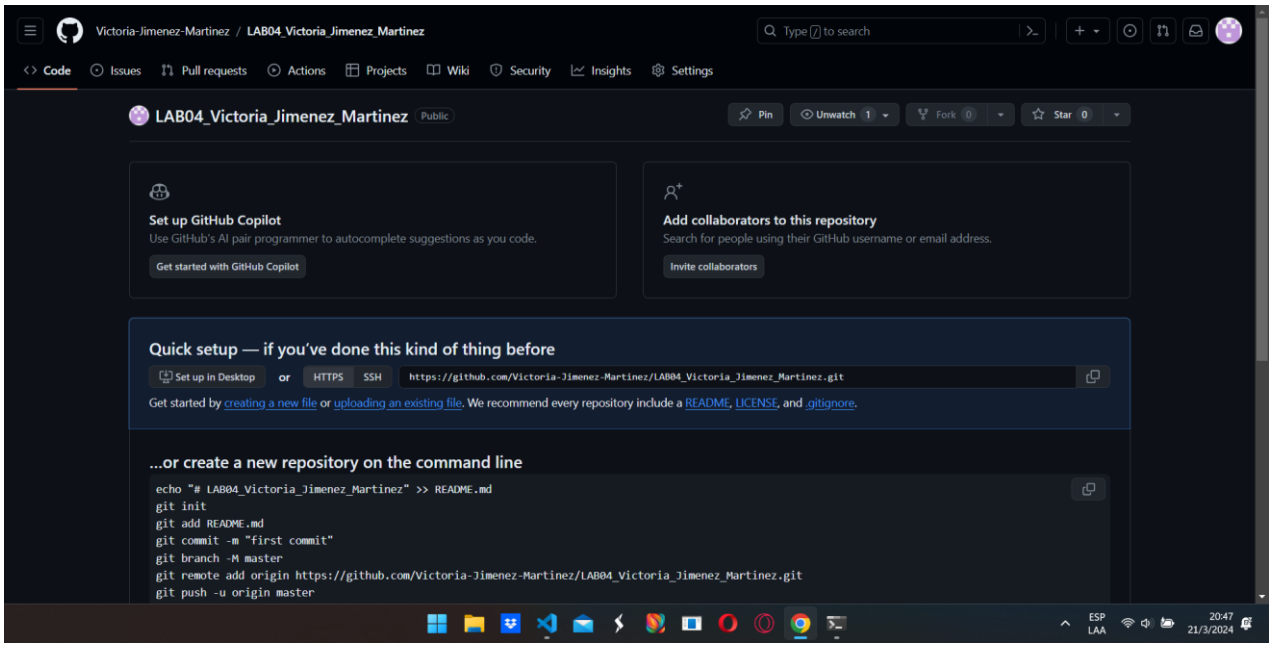
**Nombre: Victoria Jimenez Martinez**

**C.I.: 8610323 R.U.: 103334 Fecha: 21/03/2024**

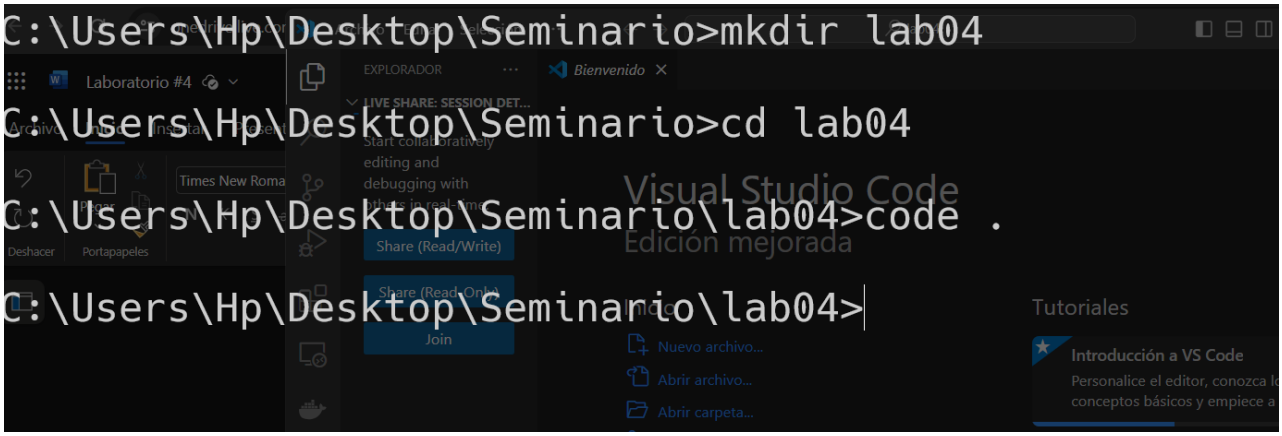
## Evaluación

Para el tema de la evaluación tan solo es necesario ir resolviendo las preguntas / desafíos planteados. Si se hacen uso de cosas nuevas, se darán información o pistas sobre ello.

1.- Crear un repo con el siguiente nombre “LAB04<Nombre>” donde el campo de “<Nombre>” deberá de ser reemplazado por su nombre.




2.- Una vez creado el repo, deberán crear una carpeta lab04 en donde ustedes deseen y abran esa carpeta desde VS Code. Una vez hecho vamos a la terminal e inserten el comando:

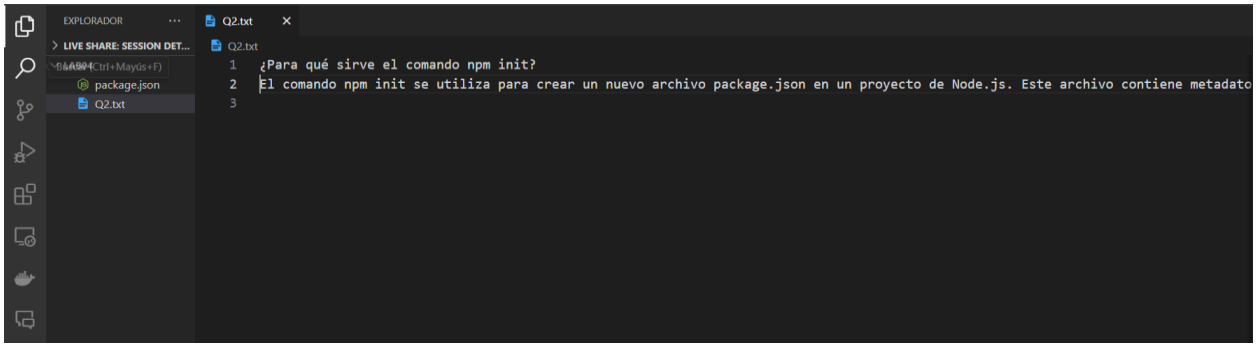


```
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help init` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (lab04) appLab
Sorry, name can no longer contain capital letters.
package name: (lab04) applab
version: (1.0.0)
description: Laboratorio04
entry point: (index.js)
test command:
git repository:
keywords:
author: vicky
license: (ISC)
About to write to C:\Users\Hp\Desktop\Seminario\lab04\package.json:

{
  "name": "applab",
  "version": "1.0.0",
  "description": "Laboratorio04",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "vicky",
  "license": "ISC"
}


Is this OK? (yes) Y

C:\Users\Hp\Desktop\Seminario\lab04>
```

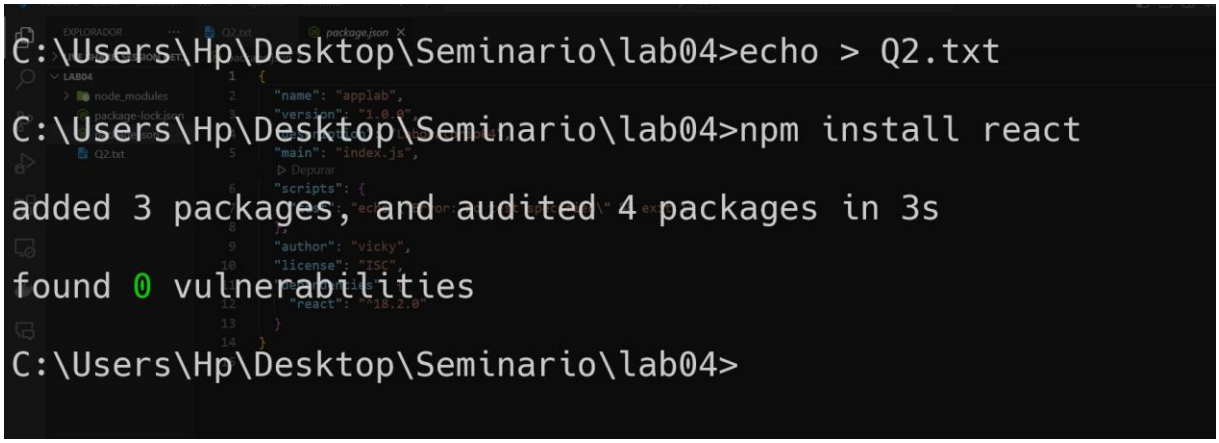
El cual contiene la siguiente información:

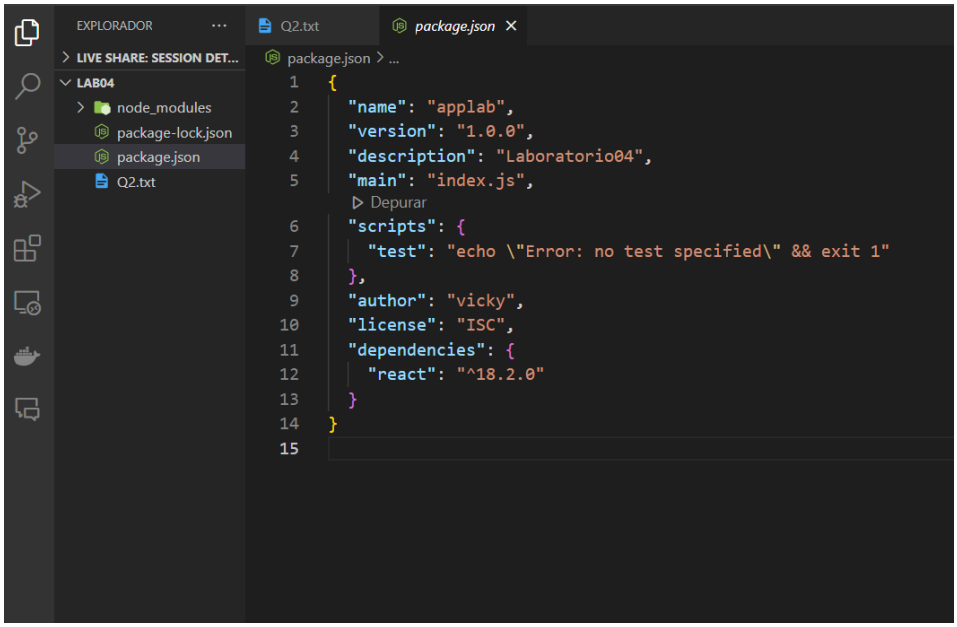
```
{
  "name": "applab",
  "version": "1.0.0",
  "description": "Laboratorio04",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "vicky",
  "license": "ISC"
}
```

Que son justamente los datos con los que rellenamos la información. Bien, una vez visto todo esto, responde lo siguiente. ¿Para qué sirve el comando npm init? Responde en un archivo llamado “Q2.txt” en la misma carpeta.



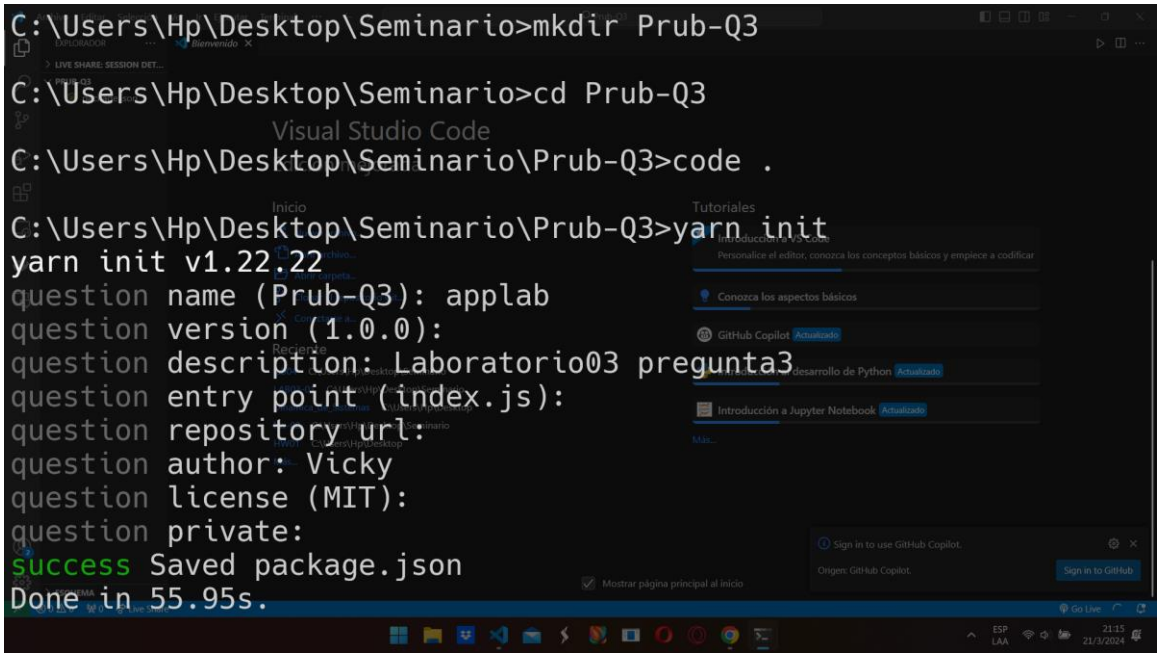

3.- Una vez realizado ello vamos a instalar react, pero antes asegúrate de estar en la ruta que apunte dentro de la carpeta donde estes trabajando, luego de eso escribe el siguiente comando:



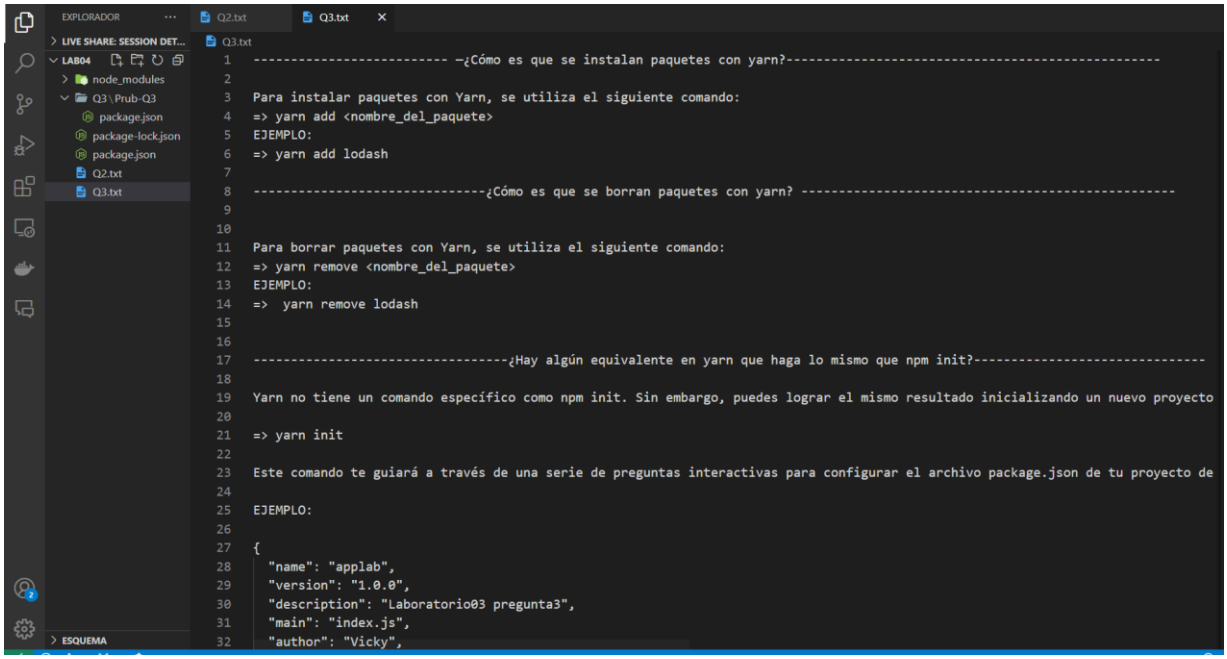



Bien ahora ya sabes como instalar paquetes y removerlos con NPM, es tu turno de investigar lo siguiente: —¿Cómo es que se instalan paquetes con yarn? —¿Cómo es que se borran paquetes con yarn?

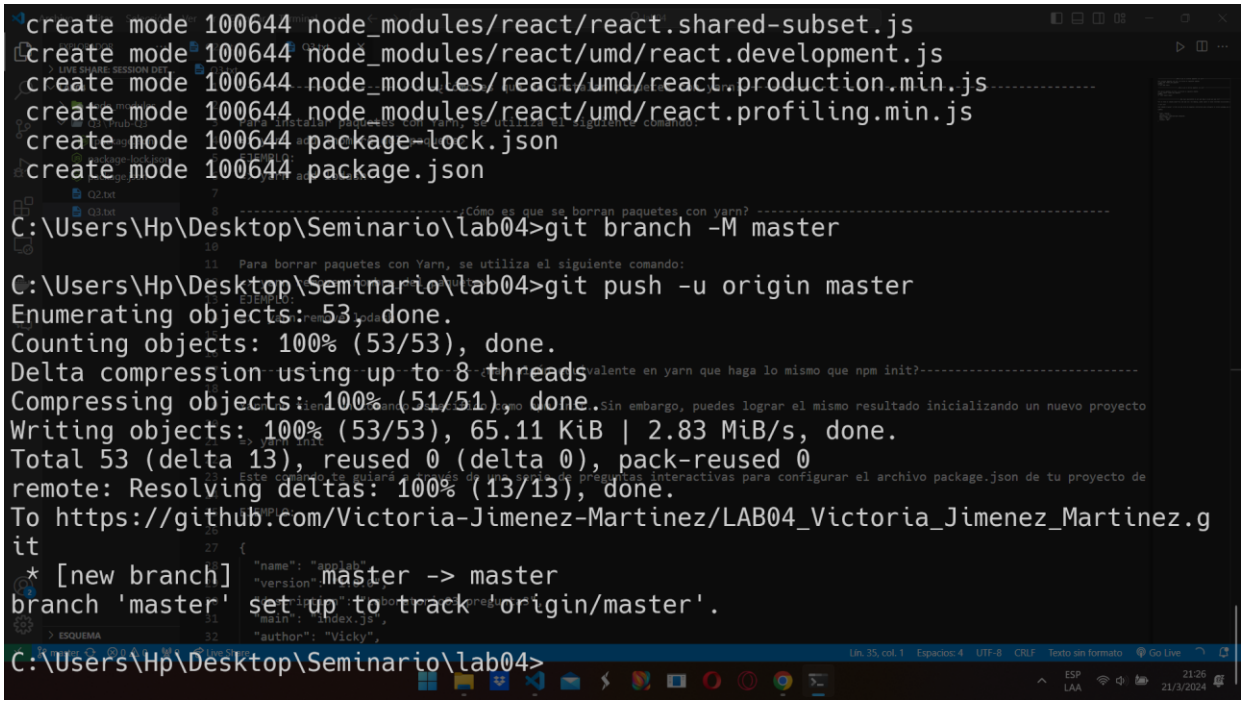
—¿Hay algún equivalente en yarn que haga lo mismo que npm init? Todas las Respuestas deben de estar en un archivo llamado “Q3.txt”

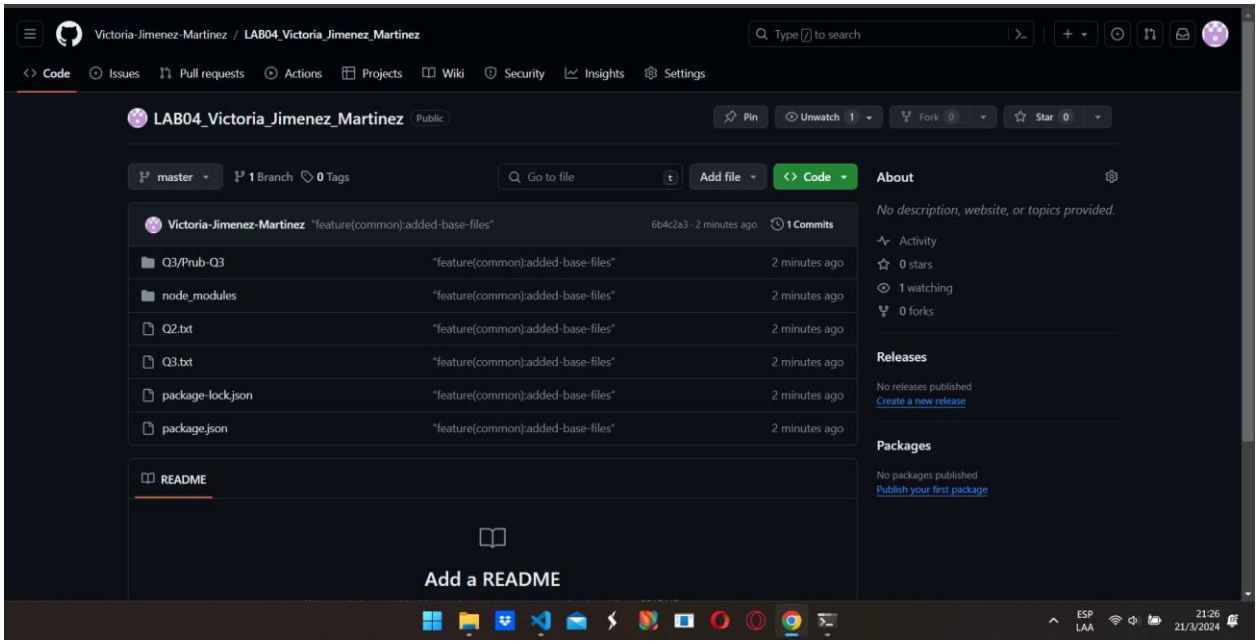


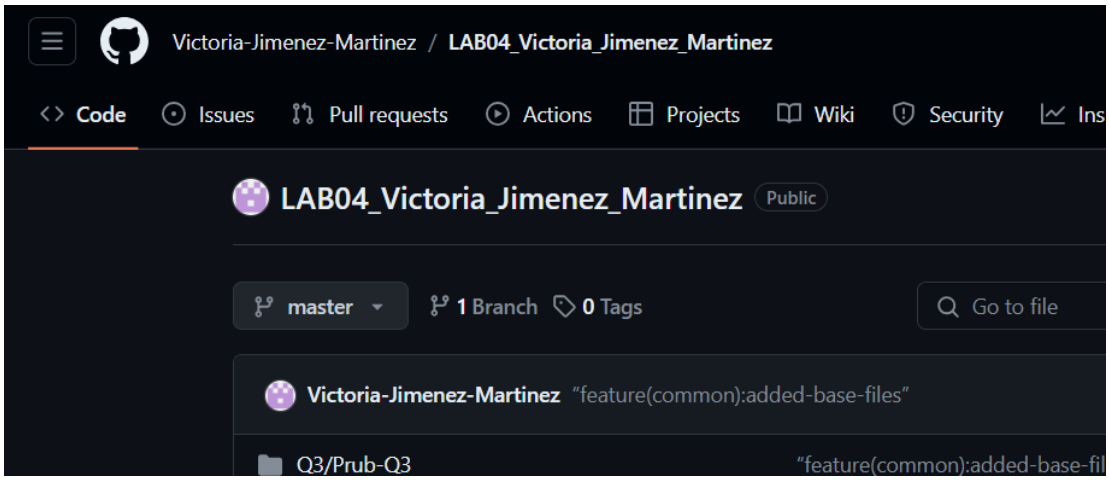
Todas las Respuestas deben de estar en un archivo llamado “Q3.txt”




4.- Una vez hechas las 3 preguntas anteriores, es hora de crear ramas. Para tal efecto, deberemos de subir todo lo que hicimos ahora al repo, el cual fue creado al comienzo. Para el commit use el siguiente texto: “feature(common): added base files”. Una vez subido el proyecto, en este apartado verán que solo aparece la rama main/master creada en un principio.






Creemos otra rama, con el siguiente comando:

```
C:\Users\Hp\Desktop\Seminario\lab04>git branch feature/LAB04-01

C:\Users\Hp\Desktop\Seminario\lab04>git branch
  feature/LAB04-01
* master

C:\Users\Hp\Desktop\Seminario\lab04>
```

El primer comando git branch feature/LAB04-01 es para la creación de ramas, y con el comando git branch se ve todas las ramas existentes, como se puede observar. Para el cambio de rama ejecutaremos el siguiente comando:

```
C:\Users\Hp\Desktop\Seminario\lab04>git branch feature/LAB04-01

C:\Users\Hp\Desktop\Seminario\lab04>git branch
  feature/LAB04-01
* master

C:\Users\Hp\Desktop\Seminario\lab04>git checkout feature/LAB04-01
Switched to branch 'feature/LAB04-01'

C:\Users\Hp\Desktop\Seminario\lab04>git branch
* feature/LAB04-01
  master

C:\Users\Hp\Desktop\Seminario\lab04>
```

Ahora en un archivo “Q4.txt” responde lo siguiente: —¿Cómo elimino una rama creada en git? Crea una nueva rama llamada feature/QA, para posteriormente borrarla.

BORRADO DE UNA RAMA

```
C:\Users\Hp\Desktop\Seminario\lab04>echo >Q4.txt

C:\Users\Hp\Desktop\Seminario\lab04>git branch feature/QA

C:\Users\Hp\Desktop\Seminario\lab04>git branch
* feature/LAB04-01
  feature/QA
  master

C:\Users\Hp\Desktop\Seminario\lab04>git branch -d  feature/QA
Deleted branch feature/QA (was 6b4c2a3).

C:\Users\Hp\Desktop\Seminario\lab04>git branch
* feature/LAB04-01
  master

C:\Users\Hp\Desktop\Seminario\lab04>
```

—¿Por qué debería de usar la ramificación en proyectos? —Usted recibió un ticket [SEM-005] y una vez terminado el trabajo usted decide crear una rama para un posterior PR. ¿Qué nombre pondría a la rama al momento de crearla?

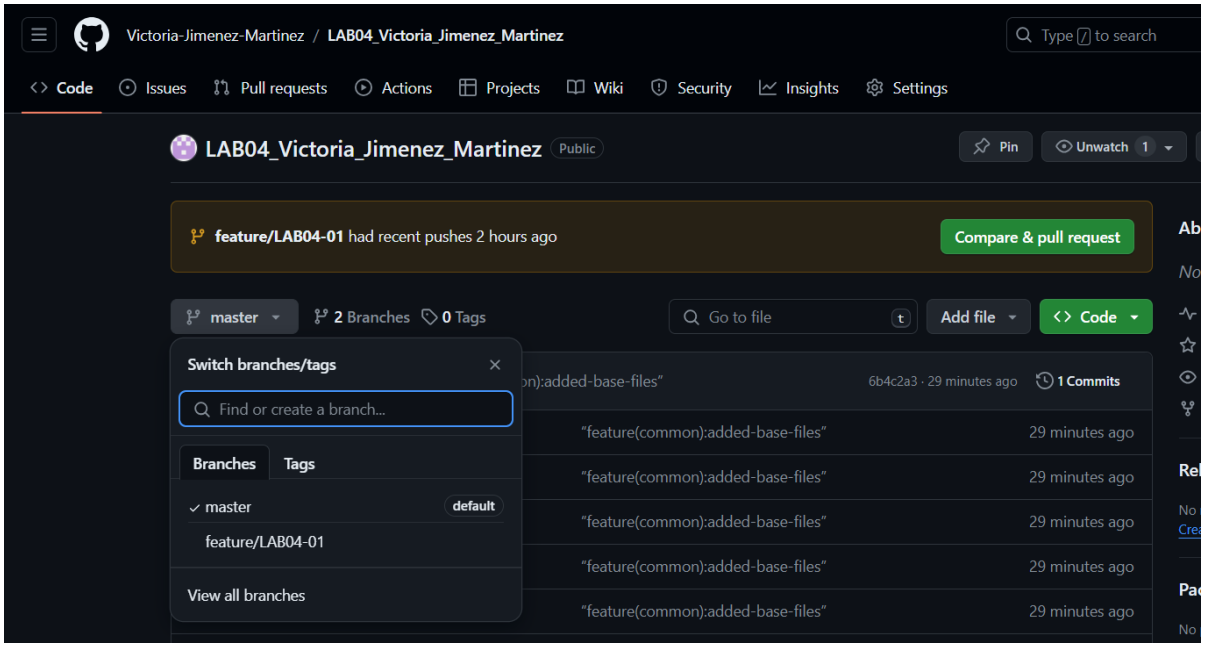
```
----------¿Cómo elimino una rama creada en git? Crea una nueva rama llamada feature/QA, para posteriormente bo


Para eliminar una rama en Git, se utiliza el siguiente comando:

=> git branch -d <nombre_de_la_rama>

EJEMPLO:

git branch -d feature/QA

=> git branch -d feature/QA
//// ver captura (BORRADO DE UNA RAMA)

---------- ¿Por qué debería de usar la ramificación en proyectos?----------

La ramificación en proyectos de Git proporciona una forma eficiente y segura de trabajar en nuevas características, arreglar erro

----------Usted recibió un ticket [SEM-005] y una vez terminado el trabajo usted decide crear una rama para un posterior

Para nombrar la rama, una práctica común es utilizar un prefijo que indique el tipo de trabajo que se está realizando, seguido de

=> git branch feature/SEM-005
```

Una vez realizado todo ello, actualice su repo subiendo todos los archivos. Para el commit use el siguiente texto: “feature(common): [added Question-4]”.

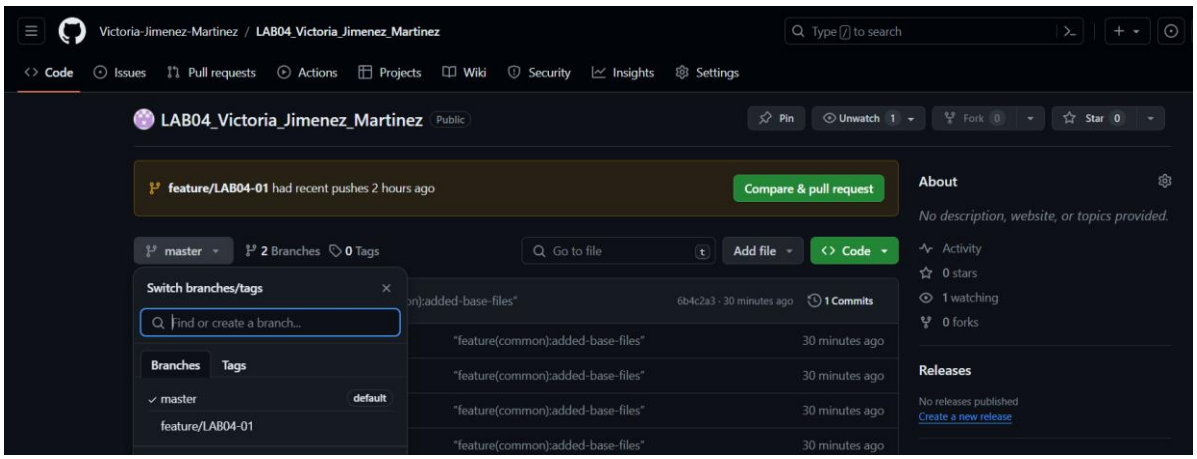
```
C:\Users\Hp\Desktop\Seminario\lab04>git status
On branch feature/LAB04-01
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Q4.txt

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

C:\Users\Hp\Desktop\Seminario\lab04>git add .

C:\Users\Hp\Desktop\Seminario\lab04>git commit -m “feature(common): [added Question-4]”
error: pathspec '[added' did not match any file(s) known to git
error: pathspec 'Question-4]”' did not match any file(s) known to git
C:\Users\Hp\Desktop\Seminario\lab04>git commit -m “feature(common):[a
dded_Question-4]”
[feature/LAB04-01 2fb8c5a] “feature(common):[added_Question-4]”
 1 file changed, 23 insertions(+)
 create mode 100644 Q4.txt

C:\Users\Hp\Desktop\Seminario\lab04>git push
fatal: The current branch feature/LAB04-01 has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin feature/LAB04-01


C:\Users\Hp\Desktop\Seminario\lab04>git push -u origin feature/LAB04-01
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
```

```
C:\Users\Hp\Desktop\Seminario\lab04>git push -u origin feature/LAB04-01
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 1.03 KiB | 1.03 MiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'feature/LAB04-01' on GitHub by visiting:
remote:      https://github.com/Victoria-Jimenez-Martinez/LAB04_Victoria_Jimenez_Martinez/pull/new/featu
re/LAB04-01
remote:
To https://github.com/Victoria-Jimenez-Martinez/LAB04_Victoria_Jimenez_Martinez.git
 * [new branch]      feature/LAB04-01 -> feature/LAB04-01
branch 'feature/LAB04-01' set up to track 'origin/feature/LAB04-01'.

C:\Users\Hp\Desktop\Seminario\lab04>
```

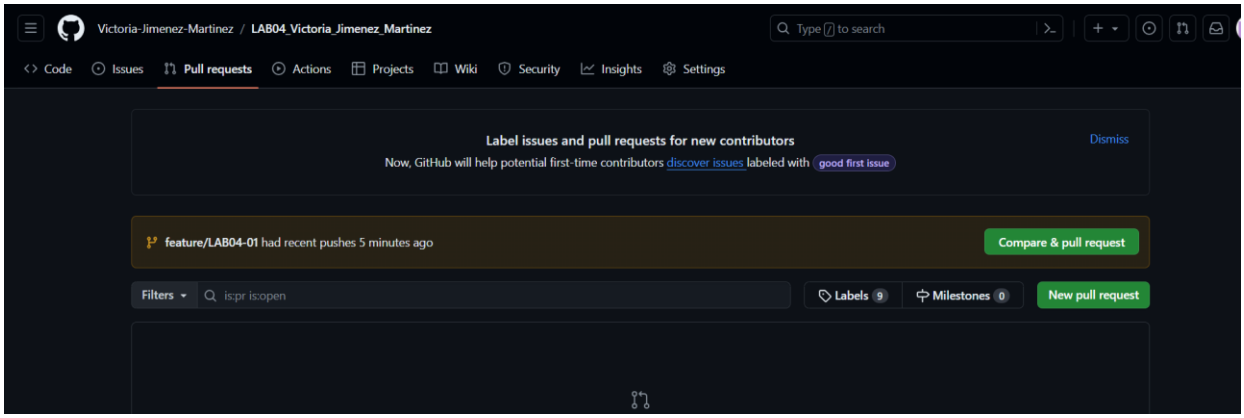
5.- Una vez realizado las 4 preguntas anteriores. Veras que tienes 2 ramas:



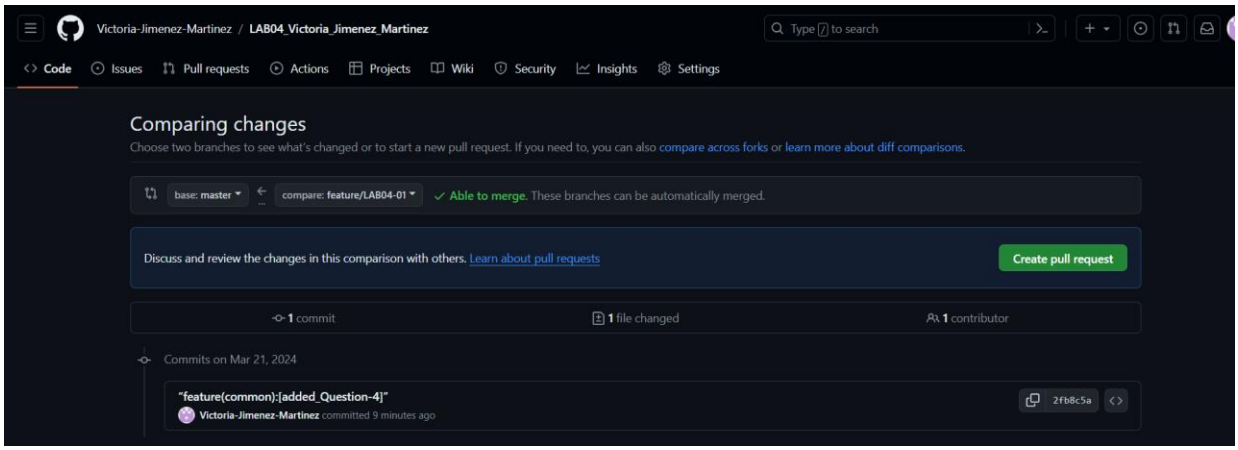

Es momento de hacer una PR y justamente, puedes observar que el siguiente mensaje apareció:



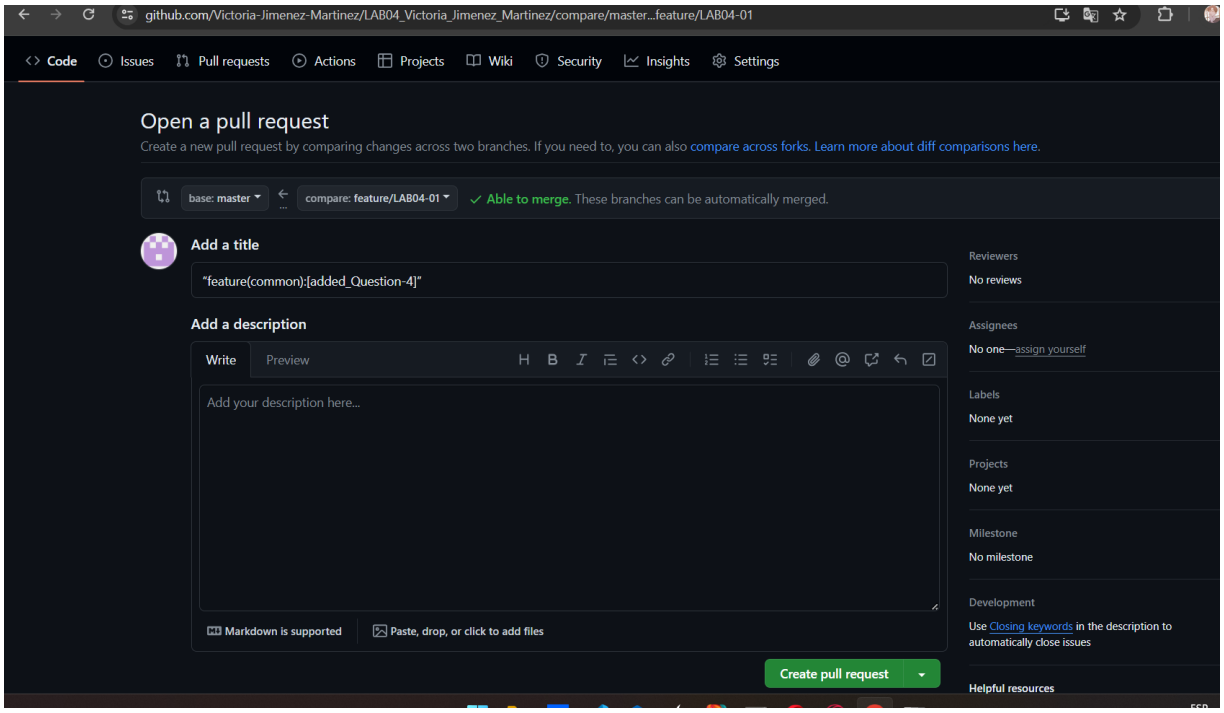

Pero hagamos las PR desde cero. Antes de eso aclaremos algunas cosas como lo es el término “PR” este se refiere a realizar una Pull Request y PR se pronuncia (pi ar) esto deletreando en el idioma inglés las letras P y R. De aquí en adelante, cada vez que veas PR se referirá a una Pull Request. Bien, para la PR vamos al siguiente apartado:

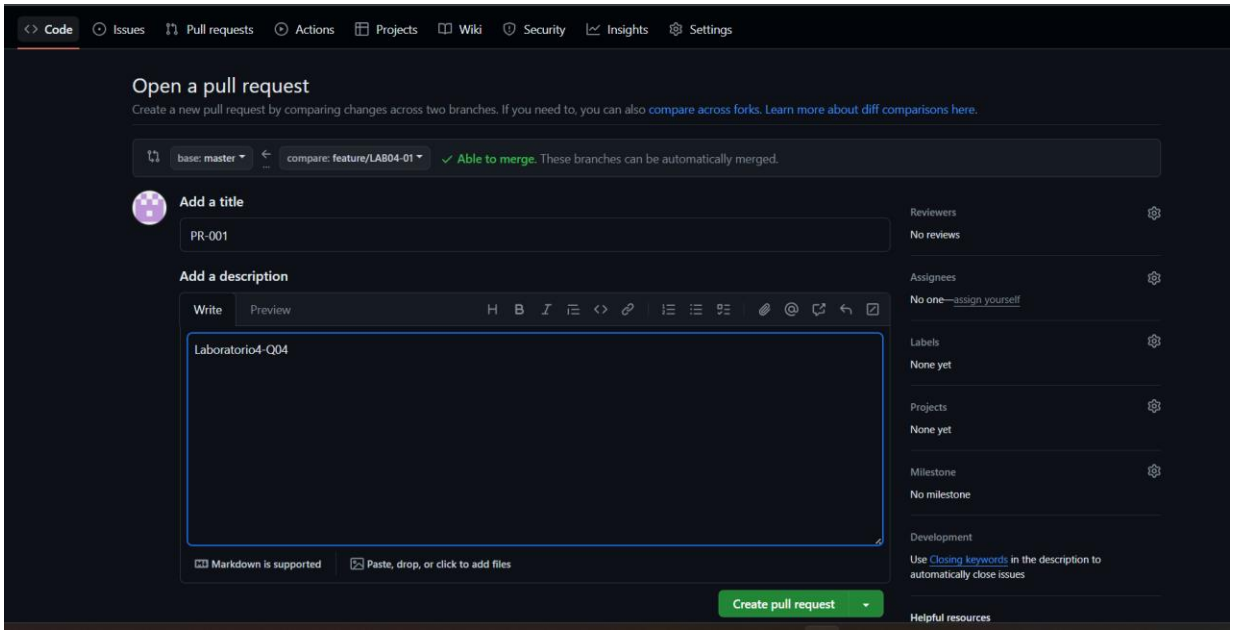



Tal como indica en la parte izquierda esta la base, y a la izquierda tenemos la rama a comparar. Seleccionemos la nueva rama que creamos para comparar con la base main.

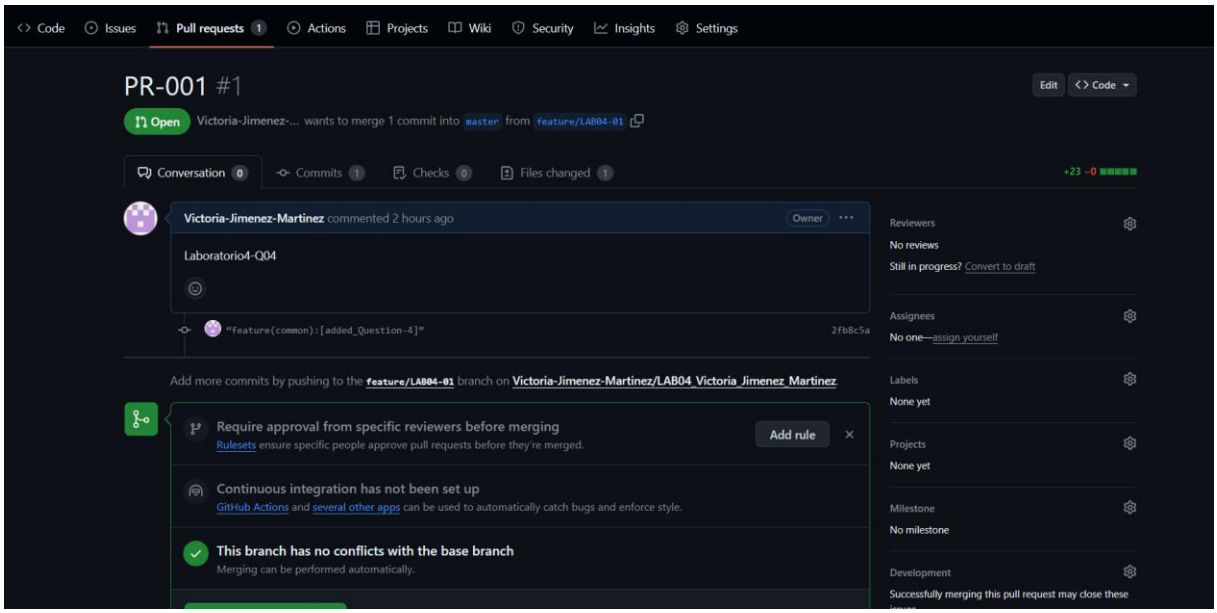



Verán que la PR ahora estará disponible y la creamos.






Deberán de colocar un título y la descripción que deseen. Y presionen Create pull request. Se mostrará lo siguiente.




Y con eso acaba la creación de la PR, te preguntaras y ¿para qué es esto? Generalmente en un entorno de trabajo se tiene a alguien a cargo, alguien con más experiencia que lidere el proyecto y vaya revisando las PR's y nombrando los errores a corregir. Y si todo está en orden dar un Aprove, y con ello hacer un Merge. Si desean pueden realizar el Merge para ver que sucede. Pero antes de realizar eso, primero hagamos lo siguiente:

Estando en la segunda rama cambia a la primera, ve los archivos, y nuevamente cambia a la segunda.

```
C:\Users\Hp\Desktop\Seminario\lab04>git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

C:\Users\Hp\Desktop\Seminario\lab04>git branch
  feature/LAB04-01
* master

C:\Users\Hp\Desktop\Seminario\lab04>
```

```
EXPLORADOR
LIVE SHARE: SESSION DET...
LAB04
node_modules
Q3\Prub-Q3
package-lock.json
package.json
Q2.txt
Q3.txt

Q2.txt
1 ¿Para qué sirve el comando npm init?
2 El comando npm init se utiliza para crear un nuevo archivo package.json en un proyecto de Node.js. Este archivo contiene metadato
3
```

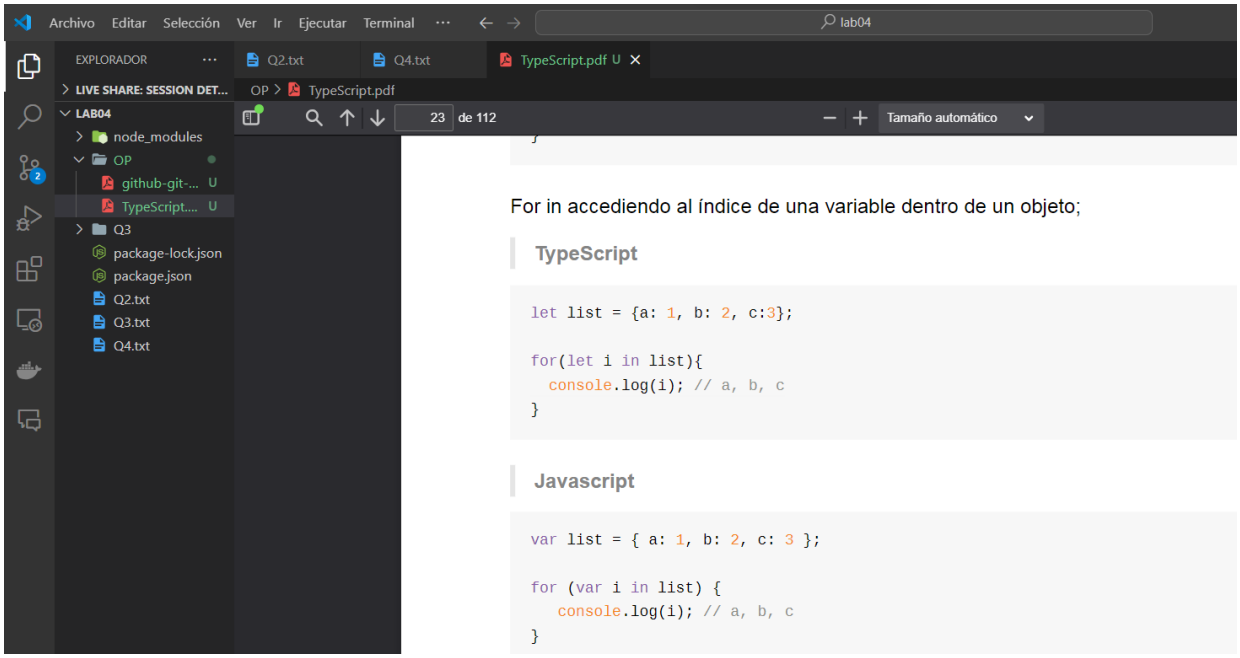
```
C:\Users\Hp\Desktop\Seminario\lab04>git checkout feature/LAB04-01
Switched to branch 'feature/LAB04-01'
Your branch is up to date with 'origin/feature/LAB04-01'.

C:\Users\Hp\Desktop\Seminario\lab04>git branch
* feature/LAB04-01
  master
Página 12 de 13   690 palabras   Español (alfab. internacional)   Sugerencias del editor: mostrando
C:\Users\Hp\Desktop\Seminario\lab04>
```

```
EXPLORADOR
LIVE SHARE: SESSION DET...
LAB04
node_modules
Q3
package-lock.json
package.json
Q2.txt
Q3.txt
Q4.txt

Q2.txt   Q4.txt
Q4.txt
1  |---------------------------¿Cómo elimino una rama creada en git? Crea una nueva rama llamada feature/QA, p
2
3
4  Para eliminar una rama en Git, se utiliza el siguiente comando:
5
6  => git branch -d <nombre_de_la_rama>
7
8  EJEMPLO:
9
10 git branch -d feature/QA
11
12 => git branch -d feature/QA
13 //// ver captura (BORRADO DE UNA RAMA)
```

Crea una tercera rama llamada “feature/Question-5” y sube los cambios. Realiza el commit con “feature(common): added question-5”.

```
C:\Users\Hp\Desktop\Seminario\lab04>git branch feature/Question-5

C:\Users\Hp\Desktop\Seminario\lab04>git branch
* feature/LAB04-01
  feature/Question-5
  master

C:\Users\Hp\Desktop\Seminario\lab04>git checkout feature/Question-5
Switched to branch 'feature/Question-5'

C:\Users\Hp\Desktop\Seminario\lab04>git branch
  feature/LAB04-01
* feature/Question-5
  master

C:\Users\Hp\Desktop\Seminario\lab04>
```

Crea una tercera rama llamada “feature/Question-5” y sube los cambios. Realiza el commit con “feature(common): added question-5”.




For in accediendo al índice de una variable dentro de un objeto;

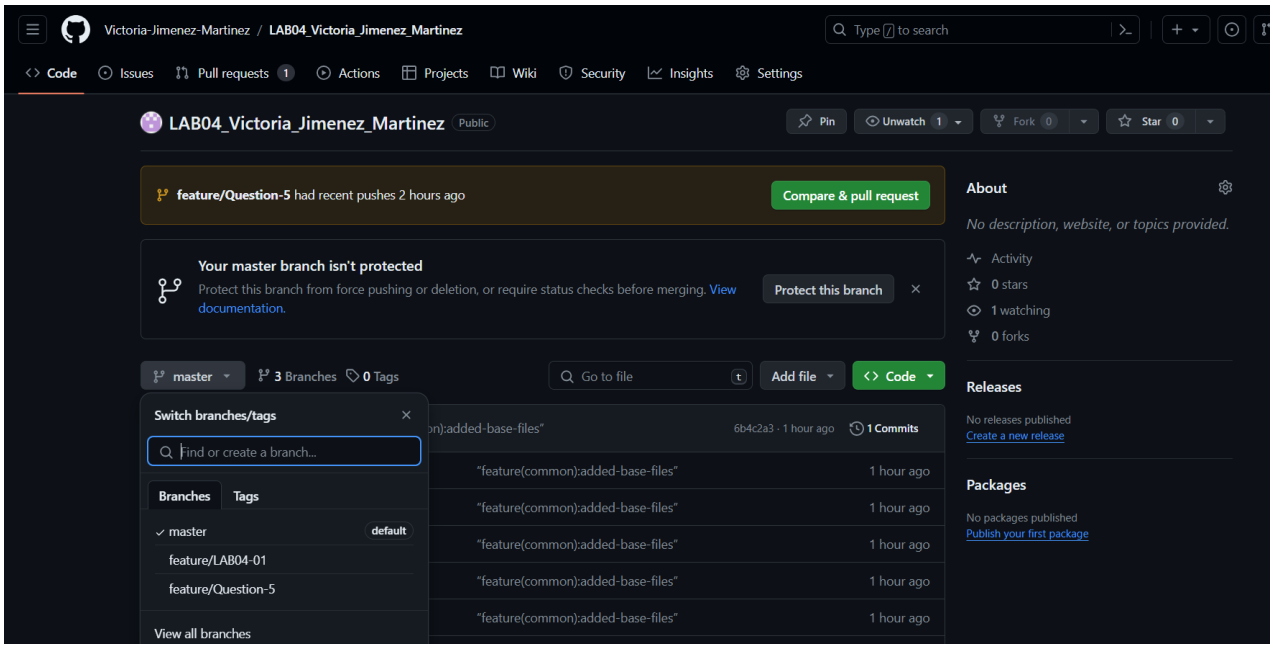
### TypeScript

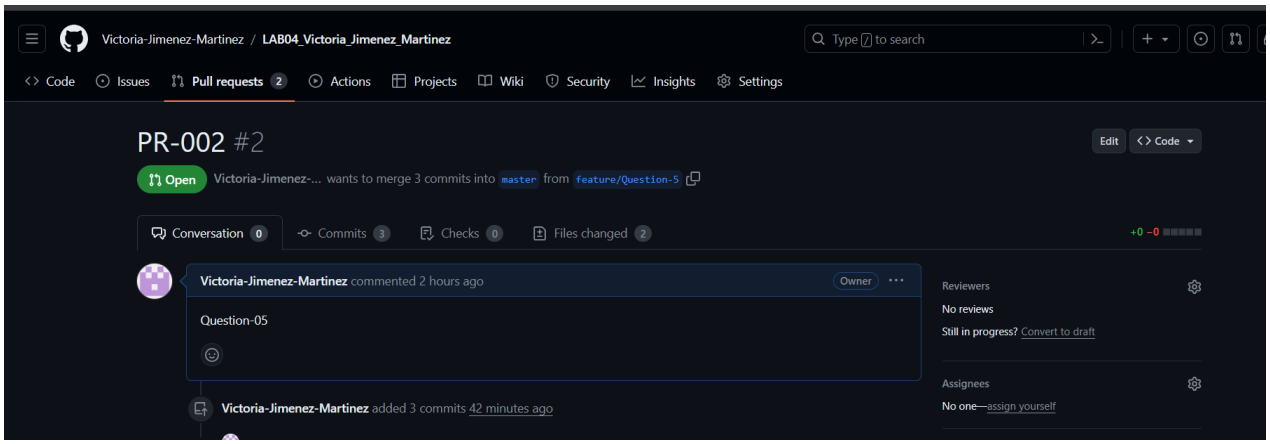
```typescript
let list = {a: 1, b: 2, c:3};

for(let i in list){
  console.log(i); // a, b, c
}
```

### Javascript

```javascript
var list = { a: 1, b: 2, c: 3 };

for (var i in list) {
   console.log(i); // a, b, c
}
```

```
C:\Users\Hp\Desktop\Seminario\lab04>git status
On branch feature/Question-5
Untracked files:
  (use "git add <file>..." to include in what will be committed)

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

C:\Users\Hp\Desktop\Seminario\lab04>git add .

C:\Users\Hp\Desktop\Seminario\lab04>git commit -m "feature(common):added
_question-5"
[feature/Question-5 de882fa] "feature(common):added_question-5"
 2 files changed, 28999 insertions(+)
 create mode 100644 OP/TypeScript.pdf
 create mode 100644 OP/github-git-cheat-sheet.pdf

Enumerating objects: 6, done.\lab04>git push -u origin feature/Question-5
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 1.86 MiB | 1.30 MiB/s, done.
Total 5 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'feature/Question-5' on GitHub by visiting:
remote:      https://github.com/Victoria-Jimenez-Martinez/LAB04_Victoria_Jimenez_Martinez/pull/new/feature/Question-5
remote:
To https://github.com/Victoria-Jimenez-Martinez/LAB04_Victoria_Jimenez_Martinez.git
 * [new branch]      feature/Question-5 -> feature/Question-5
branch 'feature/Question-5' set up to track 'origin/feature/Question-5'.

C:\Users\Hp\Desktop\Seminario\lab04>
```

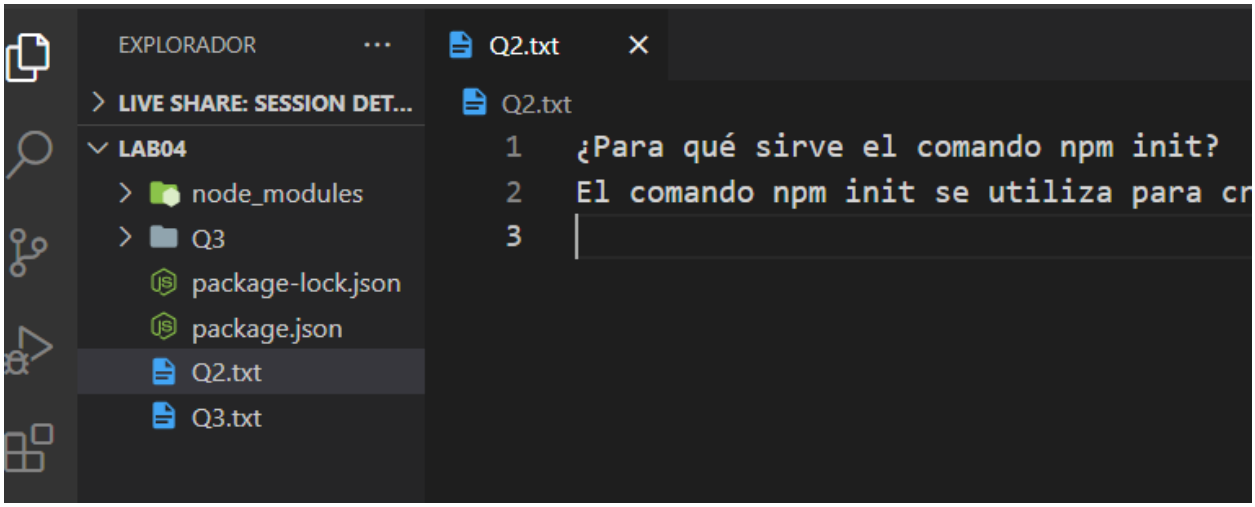



MERGE:

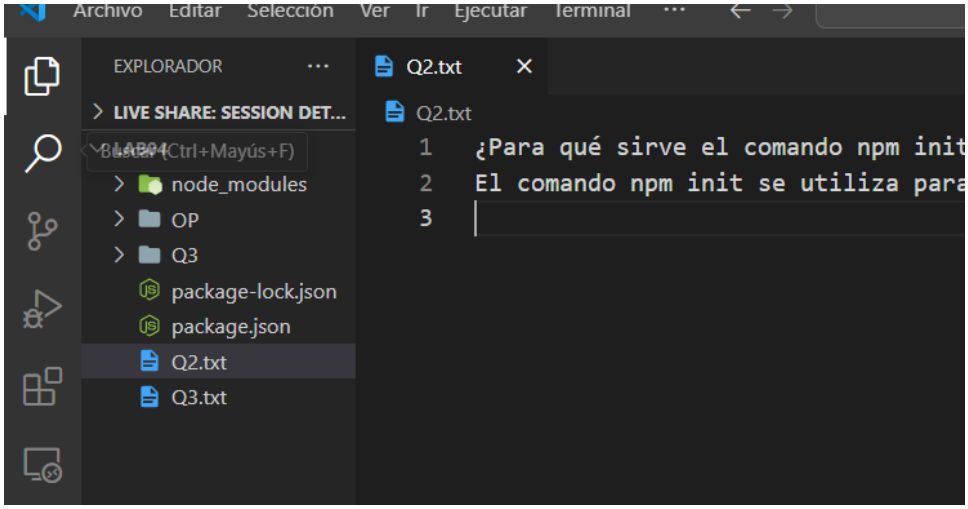
RAMA MASTER

```
C:\Users\Hp\Desktop\Seminario\lab04>git branch
* feature/LAB04-01
  feature/Question-5
  master

C:\Users\Hp\Desktop\Seminario\lab04>git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

C:\Users\Hp\Desktop\Seminario\lab04>git branch
  feature/LAB04-01
  feature/Question-5
* master

C:\Users\Hp\Desktop\Seminario\lab04>
```

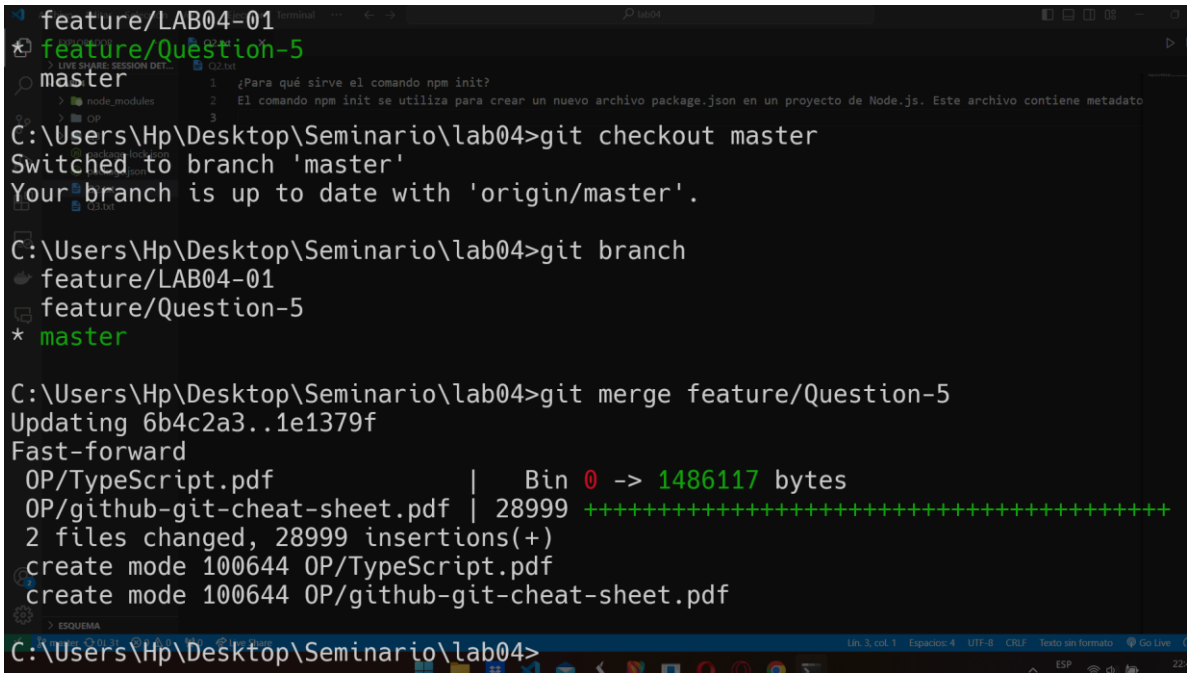



RAMA feature/Question-5

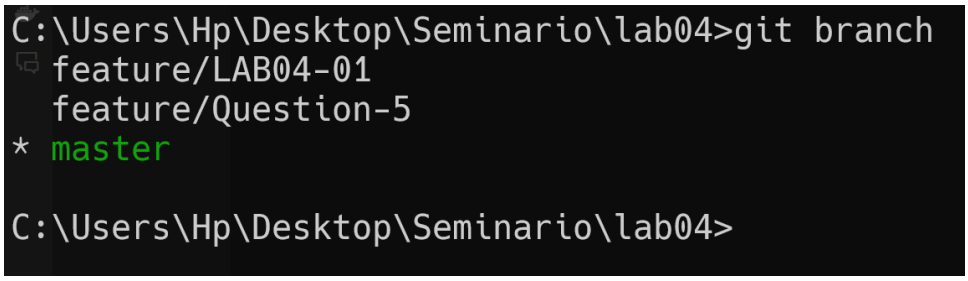
```
C:\Users\Hp\Desktop\Seminario\lab04>git branch
  feature/LAB04-01
  feature/Question-5
* master

C:\Users\Hp\Desktop\Seminario\lab04>git checkout feature/Question-5
Switched to branch 'feature/Question-5'
Your branch is up to date with 'origin/feature/Question-5'.

C:\Users\Hp\Desktop\Seminario\lab04>git branch
  feature/LAB04-01
* feature/Question-5
  master

C:\Users\Hp\Desktop\Seminario\lab04>
```

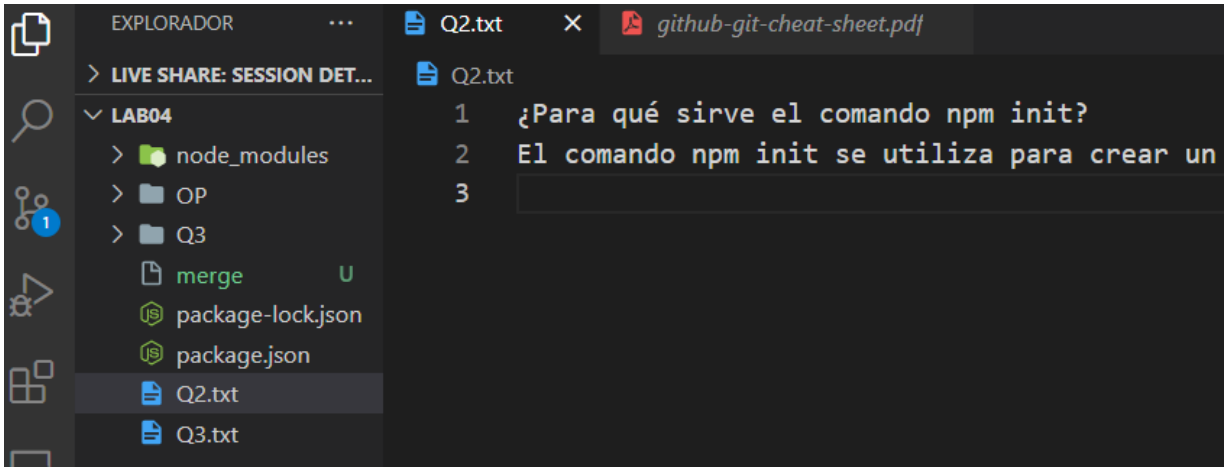



MERGE CON RAMA  feature/Question-5

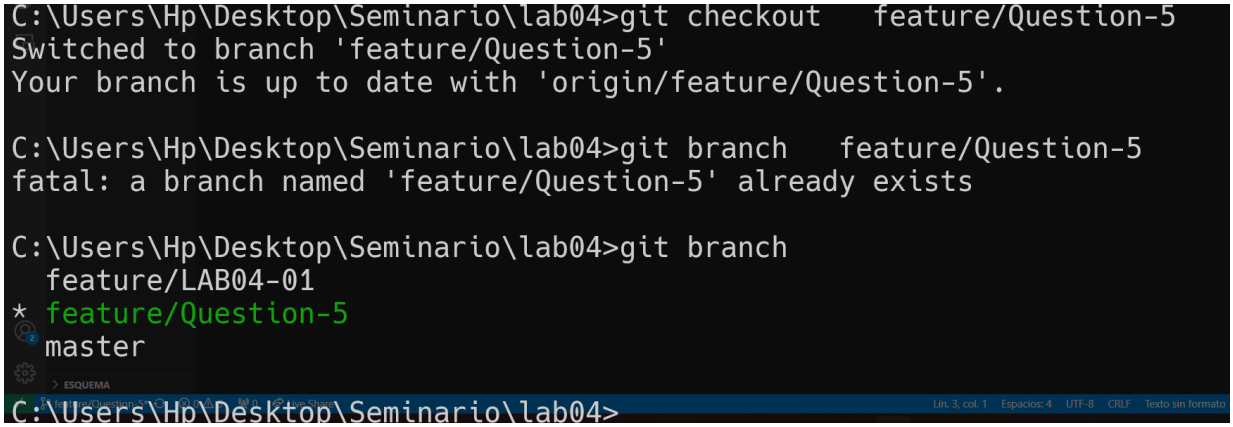


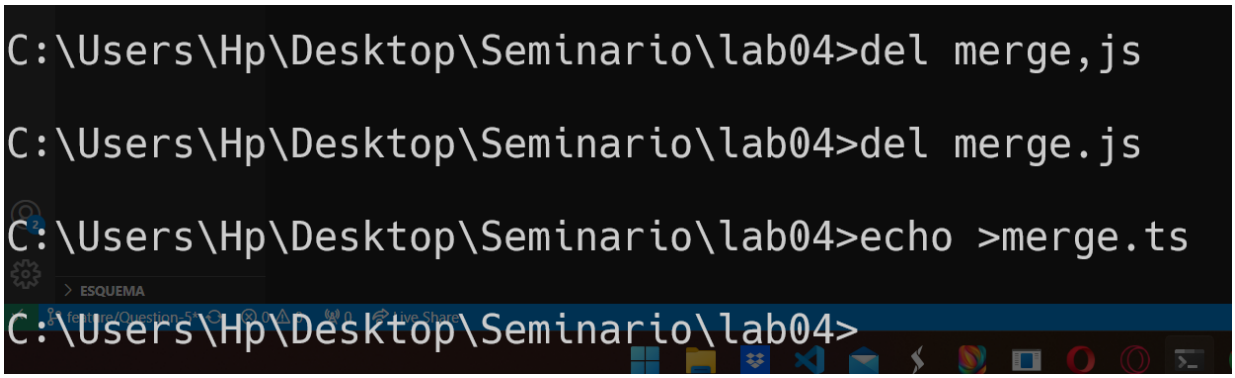

RAMA MASTER

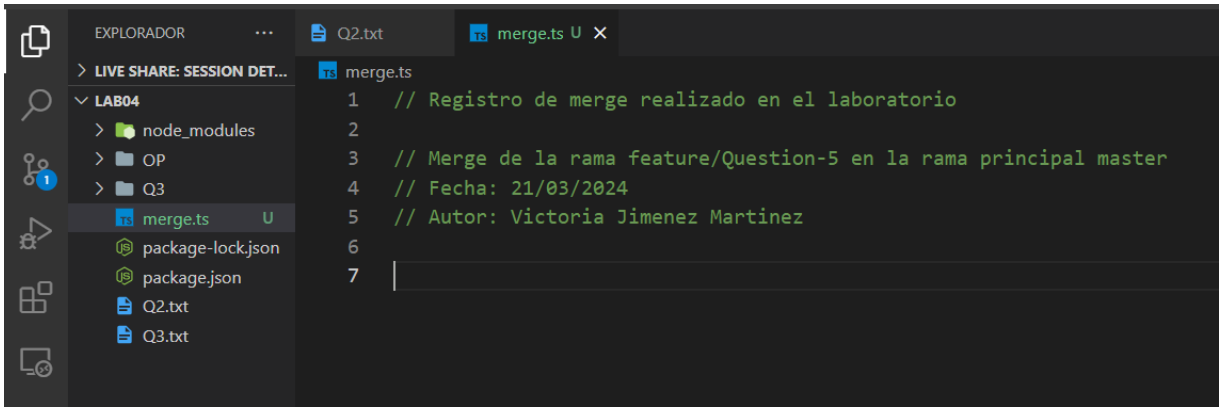





Todas las ramas hechas en el laboratorio deberán de estar en el repo, si se hizo merge de alguna especificar en un archivo aparte “merge.ts”.



```
C:\Users\Hp\Desktop\Seminario\lab04>git add .

C:\Users\Hp\Desktop\Seminario\lab04>git commit -m “feature(common):merge”
[feature/Question-5 b737c85] “feature(common):merge”
 1 file changed, 6 insertions(+)
 create mode 100644 merge.ts

C:\Users\Hp\Desktop\Seminario\lab04>git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 444 bytes | 444.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Victoria-Jimenez-Martinez/LAB04_Victoria_Jimenez_Martine
z.git
   1e1379f..b737c85  feature/Question-5 -> feature/Question-5

C:\Users\Hp\Desktop\Seminario\lab04>
```

ENLACE DE REPO

https://github.com/Victoria-Jimenez-Martinez/LAB04_Victoria_Jimenez_Martinez.git