

## Laboratorio N°9

**Estudiante:** Victoria Jimenez Martinez

**C.I. :** 8610323

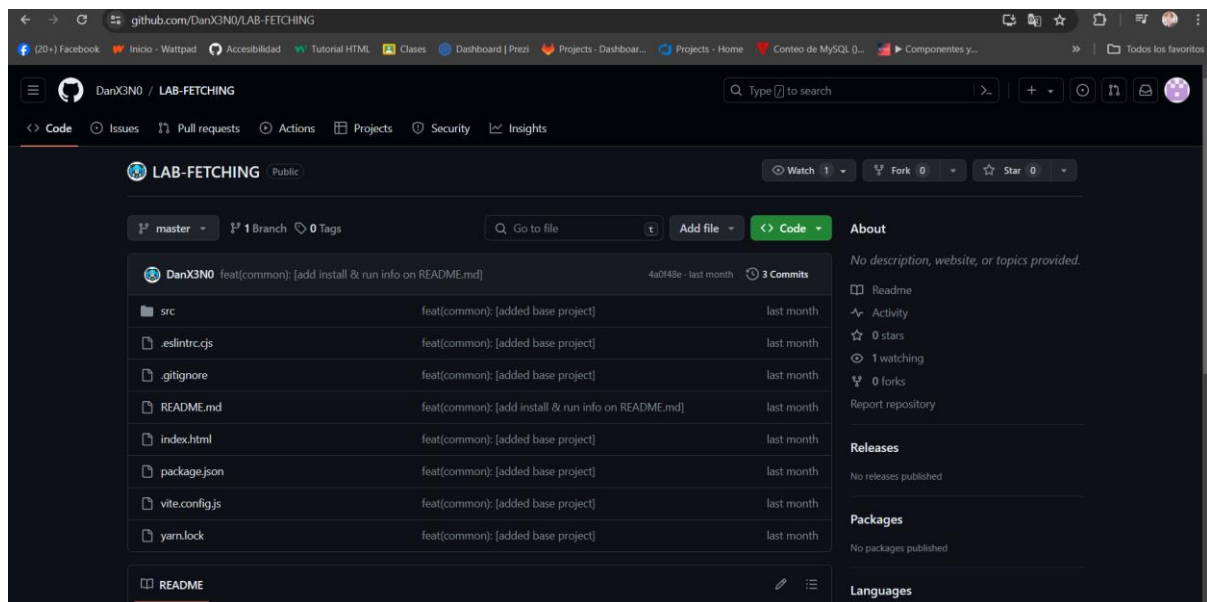
**R.U. :** 103334

## DATA FETCHING

### Evaluación

### PREGUNTA 1.

1.- Diríjase al siguiente enlace: <https://github.com/DanX3N0/LAB-FETCHING>



```
C:\Users\Hp\Desktop\Seminario>dir
El volumen de la unidad C: es Windows
El número de serie del volumen es: E2B7-2B78

Directorio de C:\Users\Hp\Desktop\Seminario

20/04/2024 01:55 <DIR> .
26/04/2024 10:32 <DIR> ..
16/03/2024 03:58 <DIR> hw-02
09/03/2024 19:46 <DIR> HW01
29/03/2024 19:30 <DIR> LAB-05
21/03/2024 10:37 <DIR> LAB03-01
21/03/2024 23:13 <DIR> lab04
07/04/2024 01:32 <DIR> LAB06
13/04/2024 11:05 <DIR> LAB07
20/04/2024 02:08 <DIR> LAB08
08/03/2024 15:04 <DIR> lab1
05/04/2024 15:32 <DIR> Laboratorios
05/04/2024 14:48 <DIR> LABS
0 archivos 0 bytes
13 dirs 143.877.742.592 bytes libres

C:\Users\Hp\Desktop\Seminario>mkdir LAB09

C:\Users\Hp\Desktop\Seminario>cd LAB09

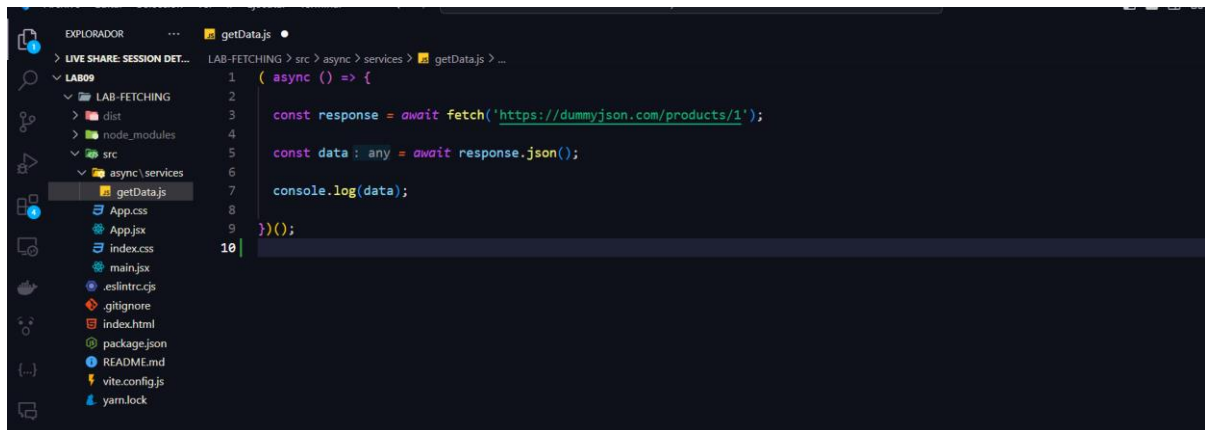
C:\Users\Hp\Desktop\Seminario\LAB09>git clone https://github.com/DanX3N0/LAB-FETCHING.git
Cloning into 'LAB-FETCHING'...
```

En el cual, si ejecutan el código, deberá de aparecer lo siguiente:

```
LAB DATA - FETCHING
```

```
[vite] connecting... client.ts:19
[vite] connected. client.ts:173
react-dom_client.js?v=7bc20356:21534
Download the React DevTools for a better development experience:
https://reactjs.org/link/react-devtools
gitData.js:17
{id: 1, title: 'iPhone 9', description: 'An apple mobile which is nothing like the apple', price: 549, discountPercentage: 12.96, ...}
```

Bien, tu primera tarea es encontrar que archivo, realiza el fetching de datos.



## PREGUNTA 2.

2.- Una vez localizado el archivo, tomate tu tiempo para analizar el código.

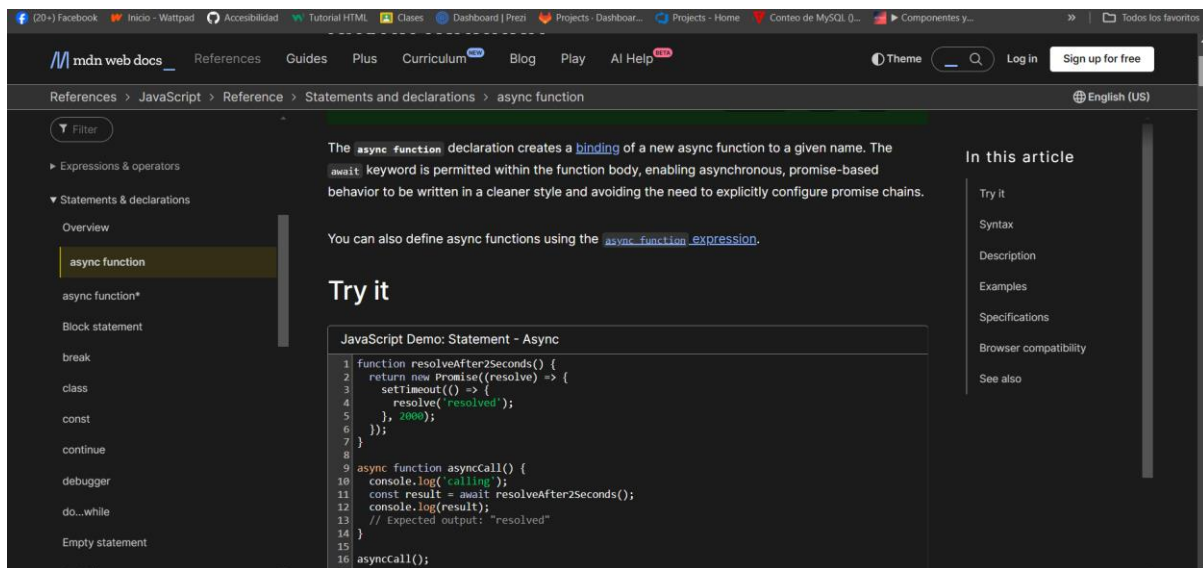


Lo importante es este fragmento de código. Primeramente, se realiza una petición asíncrona, la cual devuelve una respuesta, posterior a ello extraemos los datos que nos interesan de la respuesta y lo convertimos a un formato json, por último, imprimimos en consola lo obtenido.



Esa explicación era solo para dejar en claro que hace el código, puesto que te tocara a ti trabajar más adelante.

Si deseas revisar mas acerca de lo que es la asincronía en javascript, puedes dirigirte al siguiente enlace:



Este segundo paso tiene que ver con que tu repases un poco sobre la asincronía, y lo que es un fetch muy básico, el cual sirve de base para realizar fetch de maneras más complejas, o haciendo uso de librerías de terceros.

### PREGUNTA 3.

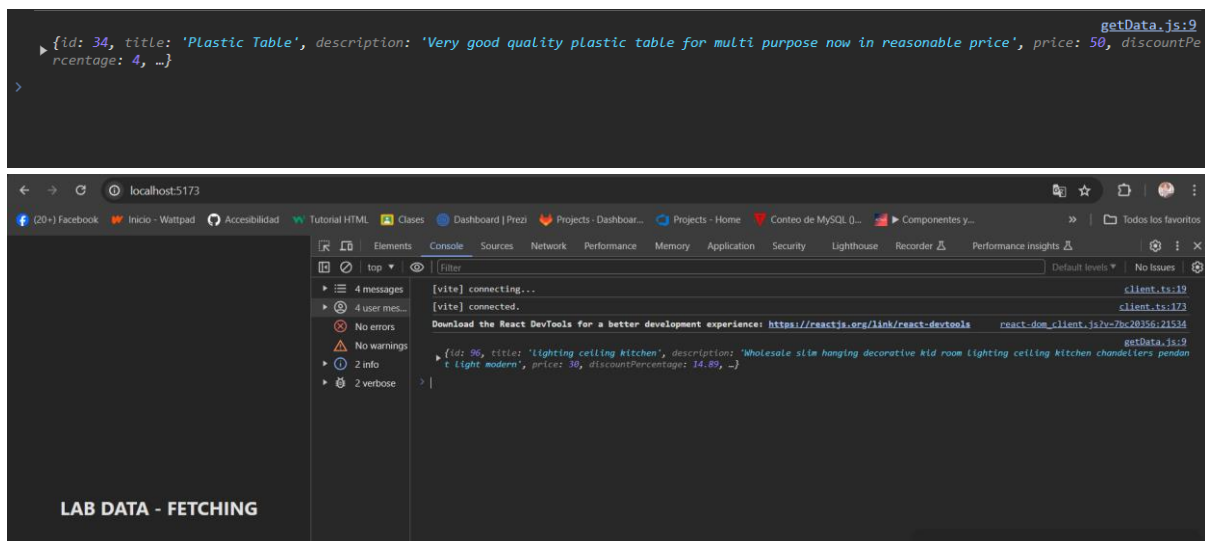
3.- El archivo que realiza el fetch de datos, si lo llegaste a analizar bien, te habrás dado cuenta, de que solo recupera un producto, pero ahora cambiemos eso, modifica la función para que retorne un producto, cuyo id este contenido en el rango [1 - 100]. Tal vez te preguntes, porque este rango y no uno mayor, o uno menor, esto se debe a que la documentación proporcionada, nos dice que son 100 productos los que están disponibles para acceder a esos datos.



Bueno, en este paso, debes modificar el archivo que hace el fetch, para recuperar de manera aleatoria un producto mediante su id. Para ello primero,

debes analizar el código, y como es que obtiene los datos mediante el fetch, luego deberás idear como obtener un numero aleatorio entre 1 y 100, y como este puede almacenarse en una variable, para luego obtener sus datos e imprimirlos en consola.

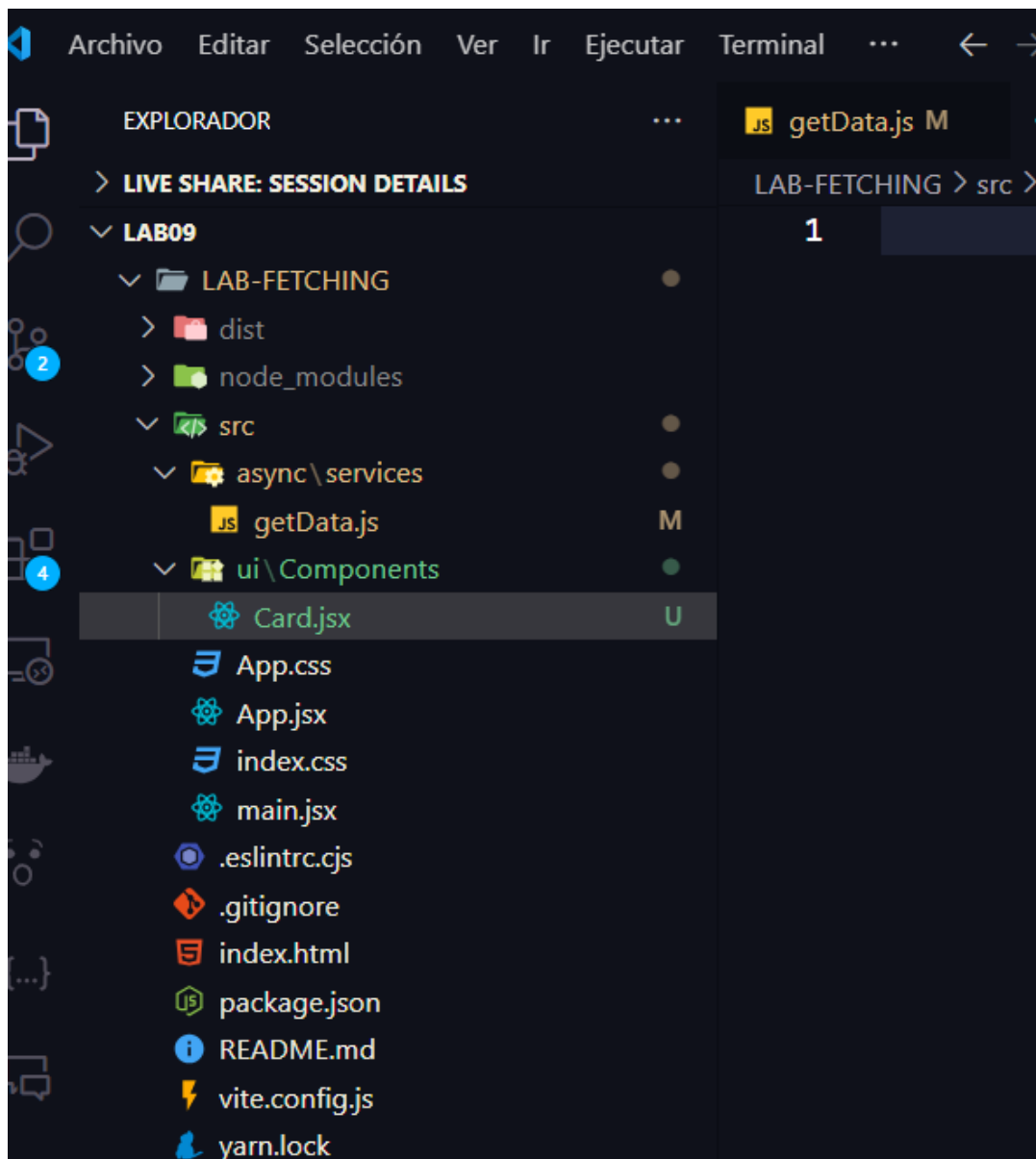
```
C:\Users\Hp\Desktop\Seminario\LAB09\LAB-FETCHING\src\async\services>node getData
{
  id: 58,
  title: 'Formal Office Shoes',
  description: 'Pattern Types Solid, (Material: PU, Toe Shape: Pointed Toe, Outsole Material: Rubber)',
  price: 57,
  discountPercentage: 12,
  rating: 4.41,
  stock: 68,
  brand: 'The Warehouse',
  category: 'mens-shoes',
  thumbnail: 'https://cdn.dummyjson.com/product-images/58/thumbnail.jpg',
  images: [
    'https://cdn.dummyjson.com/product-images/58/1.jpg',
    'https://cdn.dummyjson.com/product-images/58/2.jpg',
    'https://cdn.dummyjson.com/product-images/58/3.jpg',
    'https://cdn.dummyjson.com/product-images/58/4.jpg',
    'https://cdn.dummyjson.com/product-images/58/thumbnail.jpg'
  ]
}
C:\Users\Hp\Desktop\Seminario\LAB09\LAB-FETCHING\src\async\services>node getData
{
  id: 28,
  title: '3D Embellishment Art Lamp',
  description: '3D led lamp sticker Wall sticker 3d wall art light on/off button cell operated (included)',
  price: 20,
  discountPercentage: 16.49,
  rating: 4.82,
  stock: 54,
  brand: 'LED Lights',
  category: 'home-decoration'
}
```



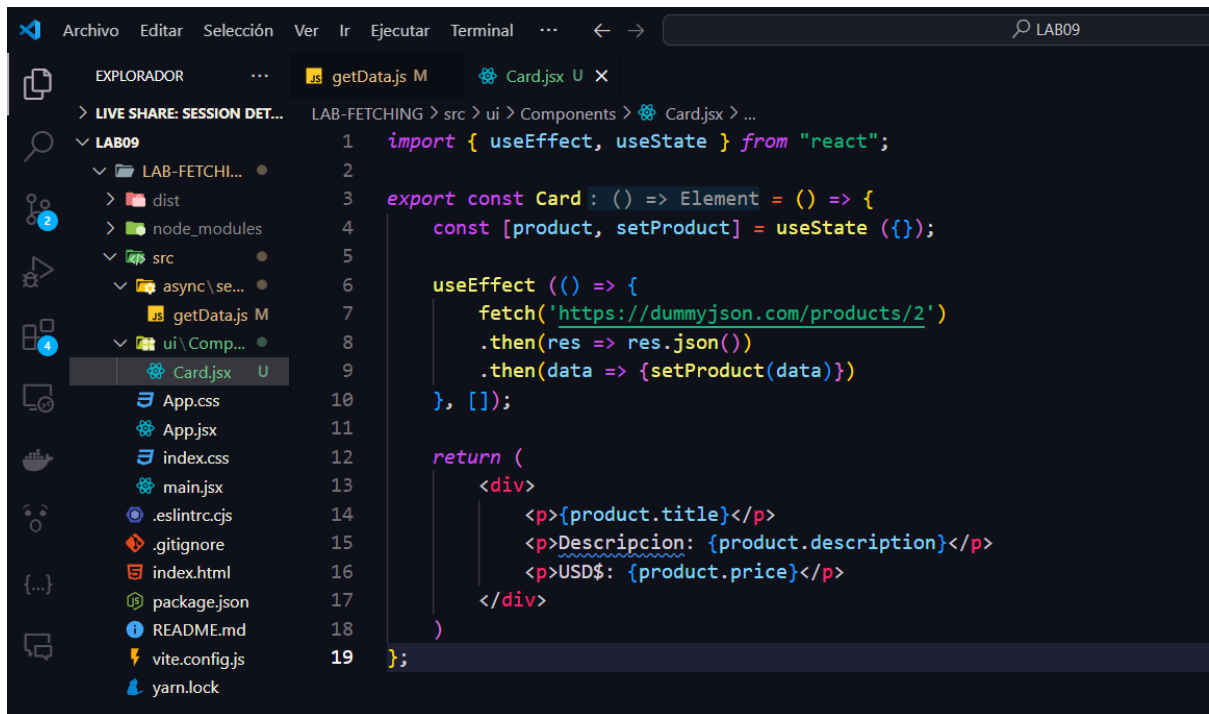
## PREGUNTA 4.

4.- Aprovechando el hecho, de que nuestro repo viene integrado con react, veamos cómo hacer la visualización de datos.

Crearemos los siguientes archivos y carpetas.

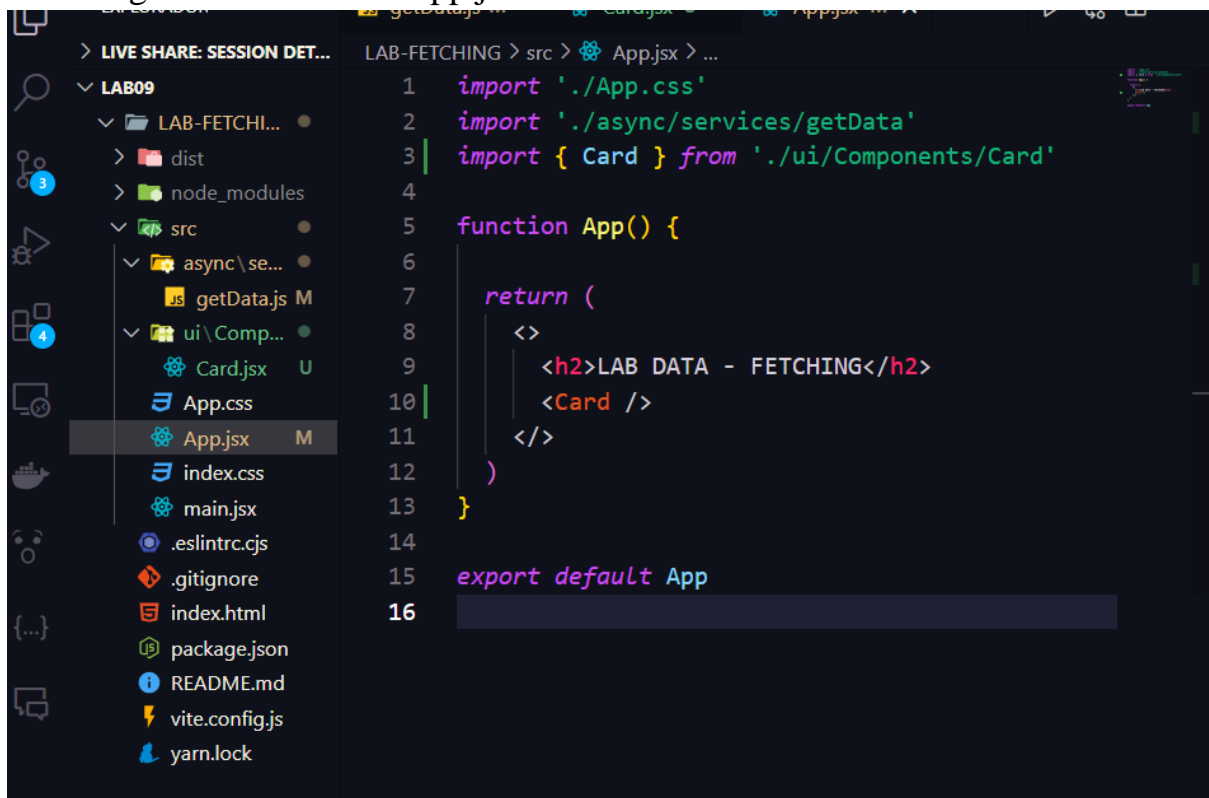


En Card.jsx haremos lo siguiente:



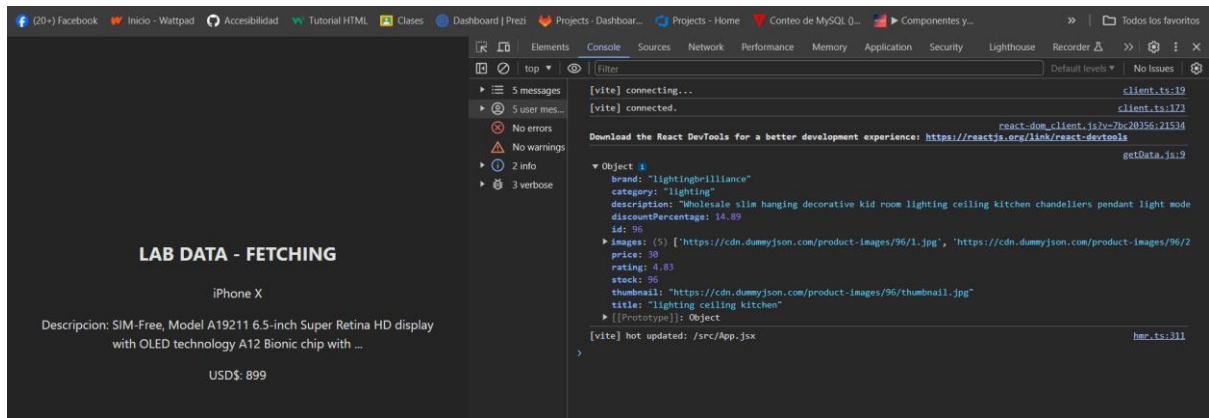
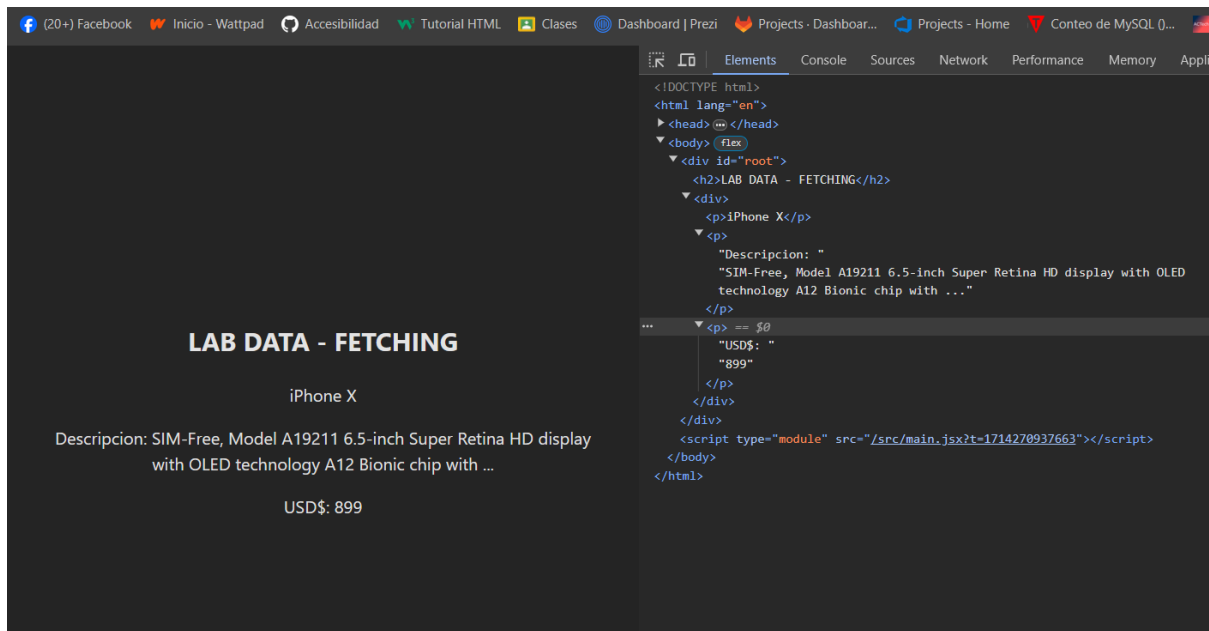
```
1 import { useEffect, useState } from "react";
2
3 export const Card : () => Element = () => {
4   const [product, setProduct] = useState ({});
5
6   useEffect (() => {
7     fetch('https://dummyjson.com/products/2')
8       .then(res => res.json())
9       .then(data => {setProduct(data)})
10   }, []);
11
12   return (
13     <div>
14       <p>{product.title}</p>
15       <p>Description: {product.description}</p>
16       <p>USD$: {product.price}</p>
17     </div>
18   )
19 };
```

Para luego llamarlo desde App.jsx



```
1 import './App.css'
2 import './async/services/getData'
3 import { Card } from './ui/Components/Card'
4
5 function App() {
6
7   return (
8     <>
9       <h2>LAB DATA - FETCHING</h2>
10       <Card />
11     </>
12   )
13 }
14
15 export default App
16
```

Con todo eso realizado, deberá de mostrarse lo siguiente:



De esa manera aprovecharemos la API, para mostrar los datos que se obtienen, a este proceso se le llama, consumo de API's. Por último, agrega lo siguiente, que obtenga de un producto aleatorio, el título, descripción, precio, y la marca. Puedes aprovechar la función que hiciste, pasos atrás.

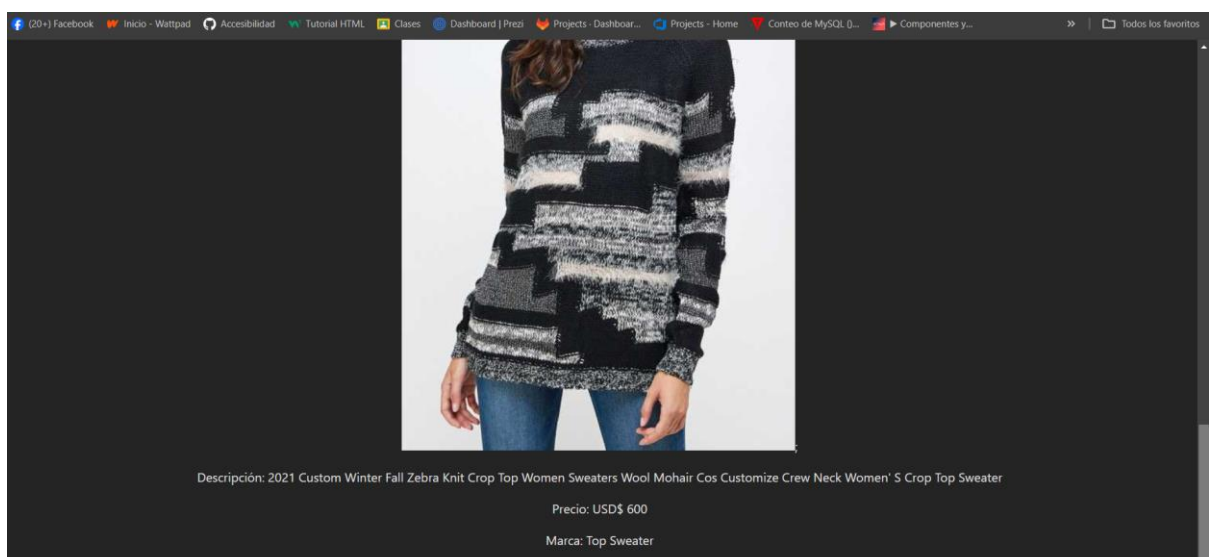
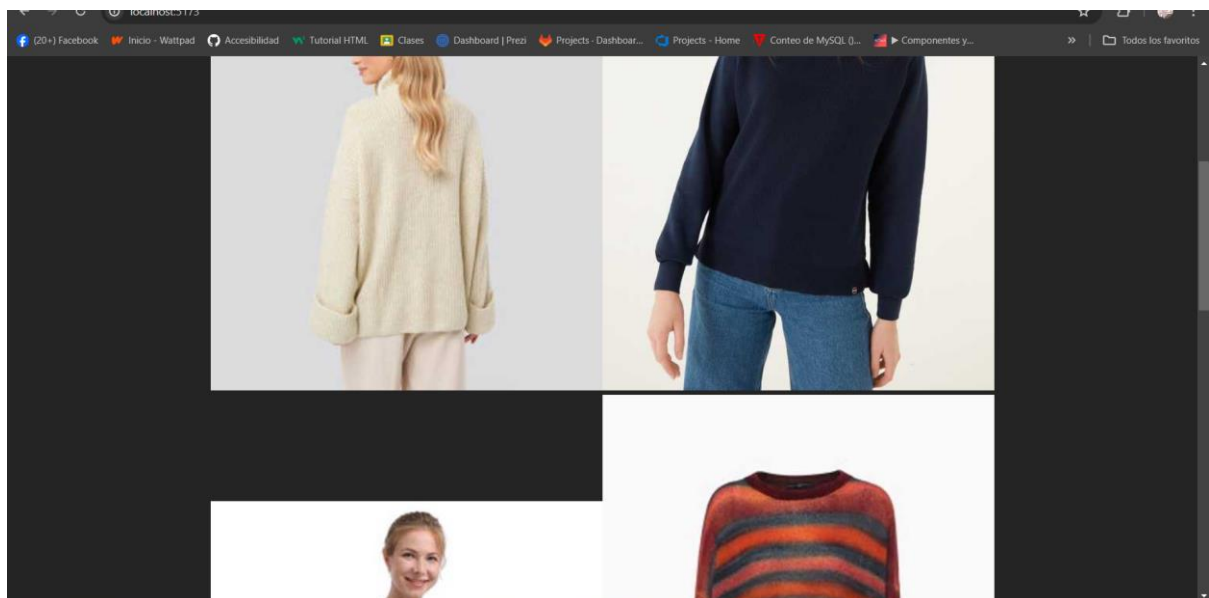


```
Archivo  Editor  Selección  Ver  Ir  Ejecutar  Terminal  ...  LAB09

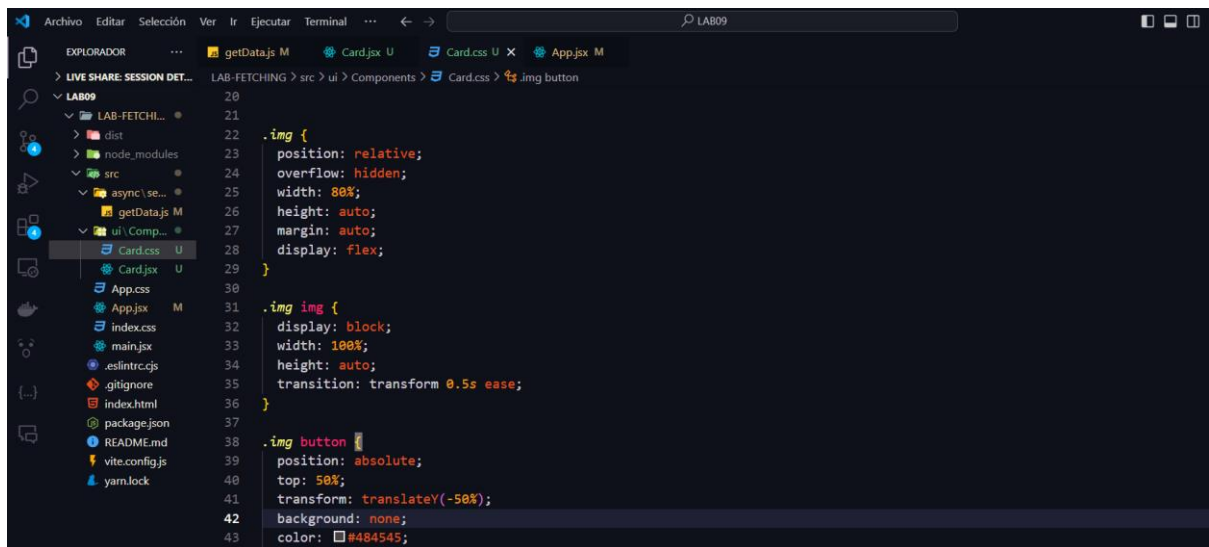
EXPLORADOR  ...  getDatajs M  Card.jsx U X  App.jsx M

> LIVE SHARE: SESSION DET...
LAB09
  > LAB-FETCHI...
    > dist
    > node_modules
    > src
      > async\se...
        > getDatajs M
        > ui\Comp...
          > Card.jsx U
            App.css
            App.jsx M
            index.css
            main.jsx
            .eslintrc.cjs
            .gitignore
            index.html
            package.json
            README.md
            vite.config.js
            yarn.lock

LAB-FETCHING > src > ui > Components > Card.jsx > Card
3  export const Card : () => Element = () => {
6
7    useEffect (() => {
8      const productId : number = Math.floor(Math.random() * 100) + 1;
9      fetch('https://dummyjson.com/products/${productId}')
10     .then(res => res.json())
11     .then(data => {setProduct(data)})
12   }, []);
13
14   return (
15     <div>
16       <h2>{product.title}</h2>
17       {product.images && product.images.length > 0 &&
18         product.images.map((image, index) => (
19           <img key={index} src={image} alt={` ${index + 1}`} />
20         ))
21       }
22       <p>Descripción: {product.description}</p>
23       <p>Precio: USD$ {product.price}</p>
24       <p>Marca: {product.brand}</p>
25     </div>
26   );
27 }
```



## 5.- Como último paso aplica css a Card.jsx

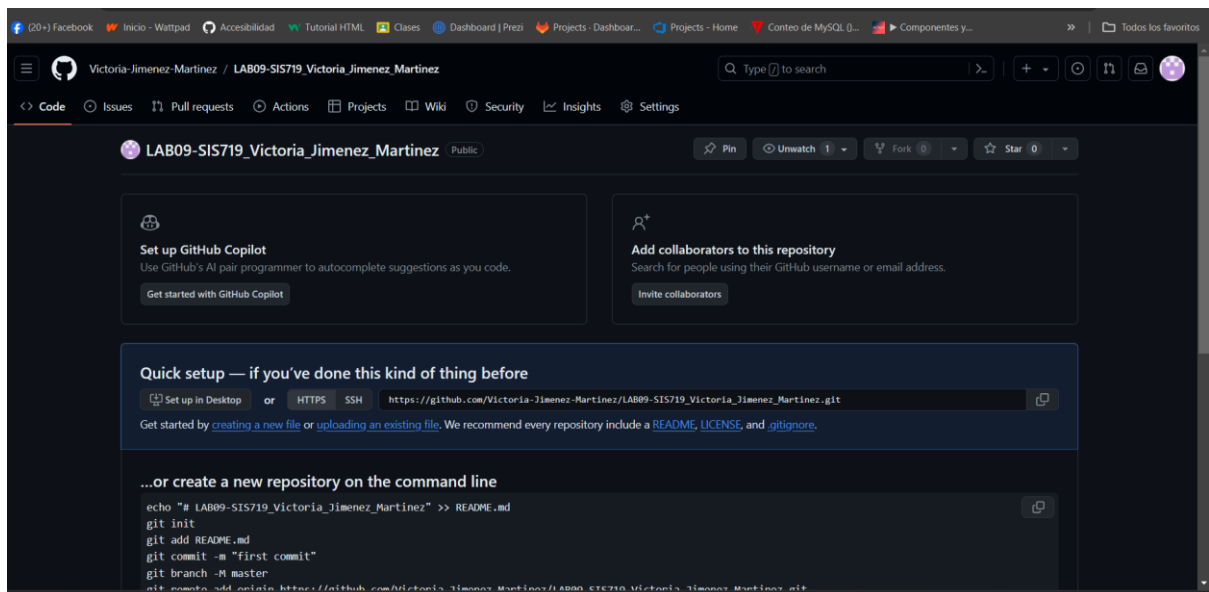
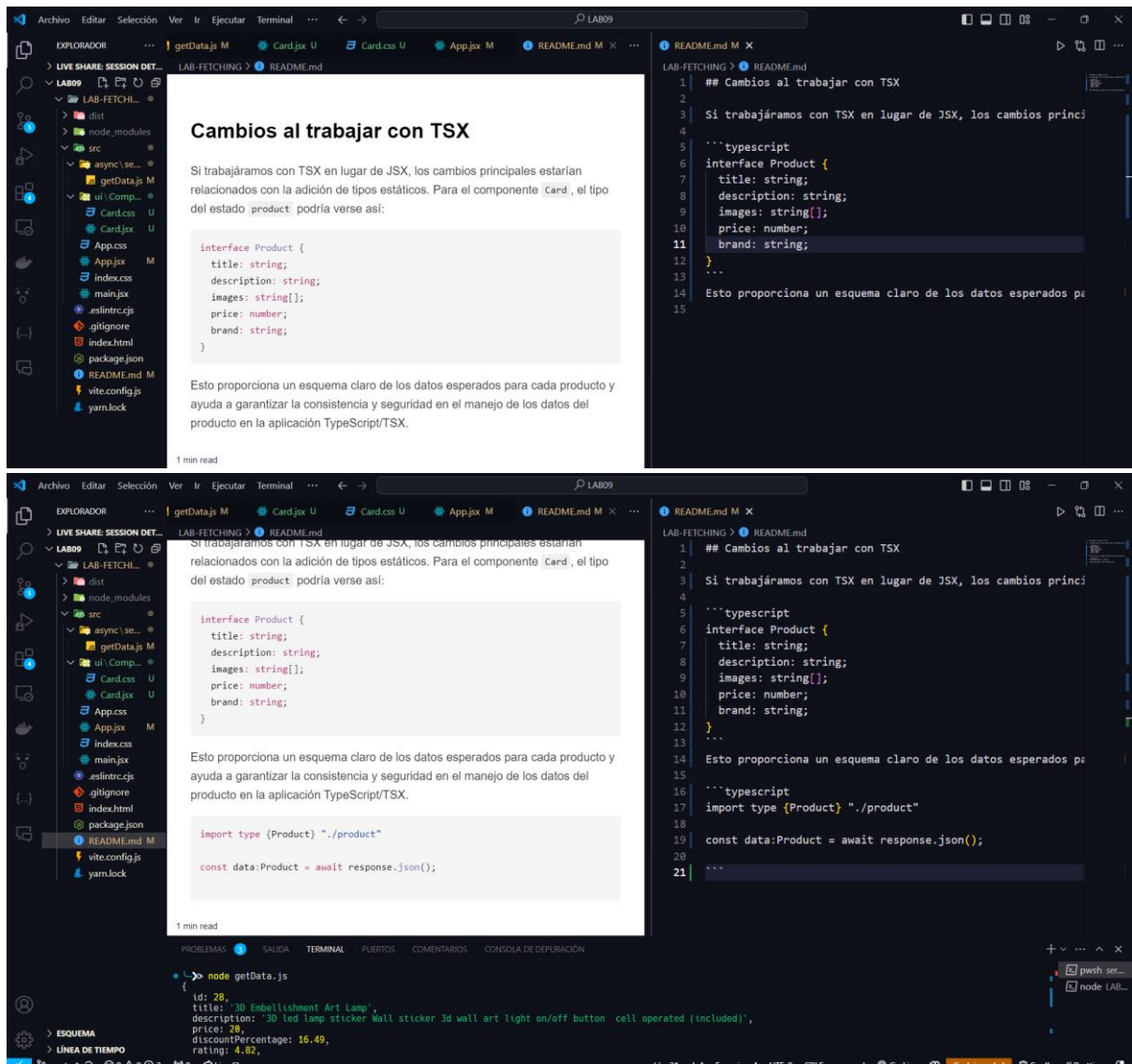


The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with files like Card.css, Card.jsx, App.css, App.jsx, index.css, main.jsx, .eslintrc.cjs, .gitignore, index.html, package.json, README.md, vite.config.js, and yarn.lock. The code editor shows the following CSS styles:

```
20
21
22 .img {
23   position: relative;
24   overflow: hidden;
25   width: 80%;
26   height: auto;
27   margin: auto;
28   display: flex;
29 }
30
31 .img img {
32   display: block;
33   width: 100%;
34   height: auto;
35   transition: transform 0.5s ease;
36 }
37
38 .img button {
39   position: absolute;
40   top: 50%;
41   transform: translateY(-50%);
42   background: none;
43   color: #484545;
```

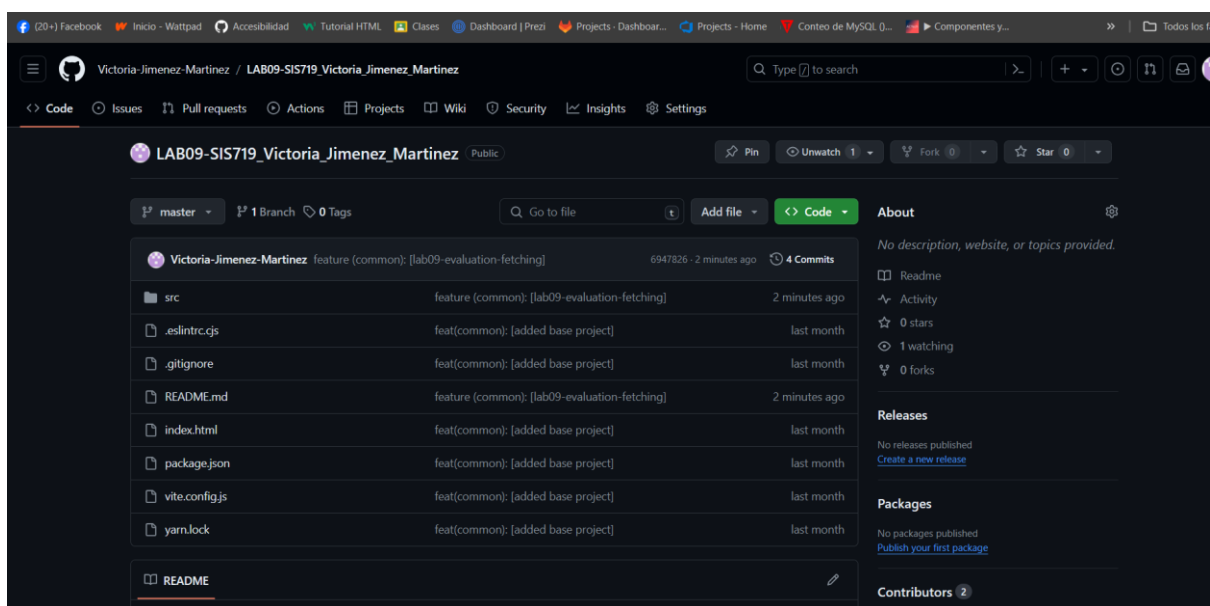


sube todo el proyecto a un repo además en el archivo [Readme.md](#) especifica que cambiaría en el lab, si se trabajara con tsx y como se vería el tipado de product.



```
C:\Users\Hp\Desktop\Seminario\LAB09\LAB-FETCHING>git remote add origin https://github.com/Victoria-Jimenez-Martinez-Martinez.git
error: remote origin already exists.

...or create a new repository on the command line
C:\Users\Hp\Desktop\Seminario\LAB09\LAB-FETCHING>git remote add new-origin https://github.com/Victoria-Jimenez-Martinez-Martinez.git
C:\Users\Hp\Desktop\Seminario\LAB09\LAB-FETCHING>git add .
C:\Users\Hp\Desktop\Seminario\LAB09\LAB-FETCHING>git commit -m "feature (common): [lab09-evaluation-fetching]"
[master 6947826]: feature (common): [lab09-evaluation-fetching]
5 files changed, 129 insertions(+), 15 deletions(-)
rewrite README.md (99%)
create mode 100644 src/ui/Components/Card.css
create mode 100644 src/ui/Components/Card.jsx
C:\Users\Hp\Desktop\Seminario\LAB09\LAB-FETCHING>git branch -M master
C:\Users\Hp\Desktop\Seminario\LAB09\LAB-FETCHING>git push -u new-origin master
Enumerating objects: 35, done.
Counting objects: 100% (35/35), done.
Delta compression using up to 8 threads
Compressing objects: 100% (26/26), done.
Writing objects: 100% (35/35), 47.53 KiB | 9.50 MiB/s, done.
Total 35 (delta 7), reused 20 (delta 4), pack-reused 0
remote: Resolving deltas: 100% (7/7), done.
To https://github.com/Victoria-Jimenez-Martinez-Martinez/LAB09-SIS719_Victoria_Jimenez_Martinez.git
 * [new branch]      master -> master
branch 'master' set up to track 'new-origin/master'.
```



URL de Repositorio: [https://github.com/Victoria-Jimenez-Martinez-Martinez/LAB09-SIS719\\_Victoria\\_Jimenez\\_Martinez.git](https://github.com/Victoria-Jimenez-Martinez-Martinez/LAB09-SIS719_Victoria_Jimenez_Martinez.git)