



ESCUELA POLITÉCNICA NACIONAL
ESCUELA DE FORMACIÓN DE TECNÓLOGOS

DESARROLLO DE APLICACIONES WEB



ASIGNATURA:

Desarrollo de Aplicaciones Web

PROFESOR:

Ing. Loarte Byron

PERÍODO ACADÉMICO:

LABORATORIO - 2

TÍTULO:

CONSUMO DE UNA API



JSONPlaceholder

Fake Online REST API for Testing and Prototyping

PROPÓSITO DE LA PRÁCTICA

Familiarizar al estudiante con el lenguaje de intercambio de información y el consumo de una API FAKE

OBJETIVO GENERAL

Consumir una API pública mediante lenguajes como CSS, JS, HTML y Bootstrap

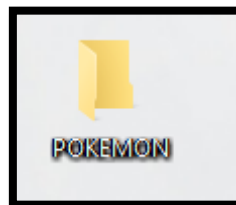
OBJETIVOS ESPECÍFICOS

- Configurar las herramientas para el laboratorio
- Elaborar un documento HTML para presentar los datos
- Implementar estilos mediante Bootstrap
- Elaborar un documento con toda la sintaxis necesaria para la lectura de datos en formato JSON
- Elaborar un documento .js con toda la sintaxis necesaria para la lectura de datos en formato JSON
- Visualizar los resultados obtenidos

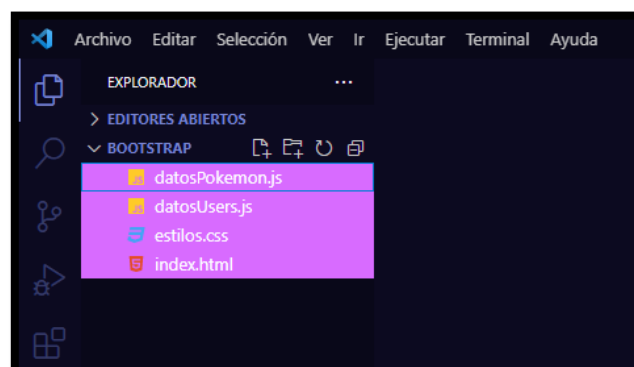
INSTRUCCIONES

1. Consumir una API de Pokemon, para lo cual se procede a crear una carpeta en el escritorio llamada “POKEMON”

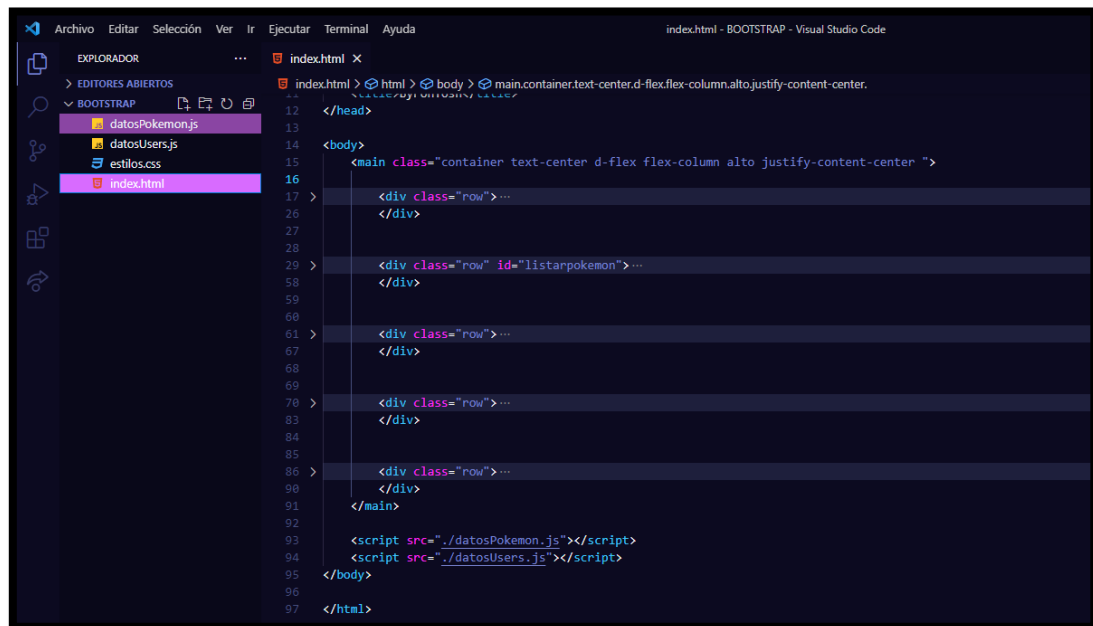
API DE POKEMON



2. Abrir la carpeta con Visual Studio Code y crear 3 archivos y un index.html

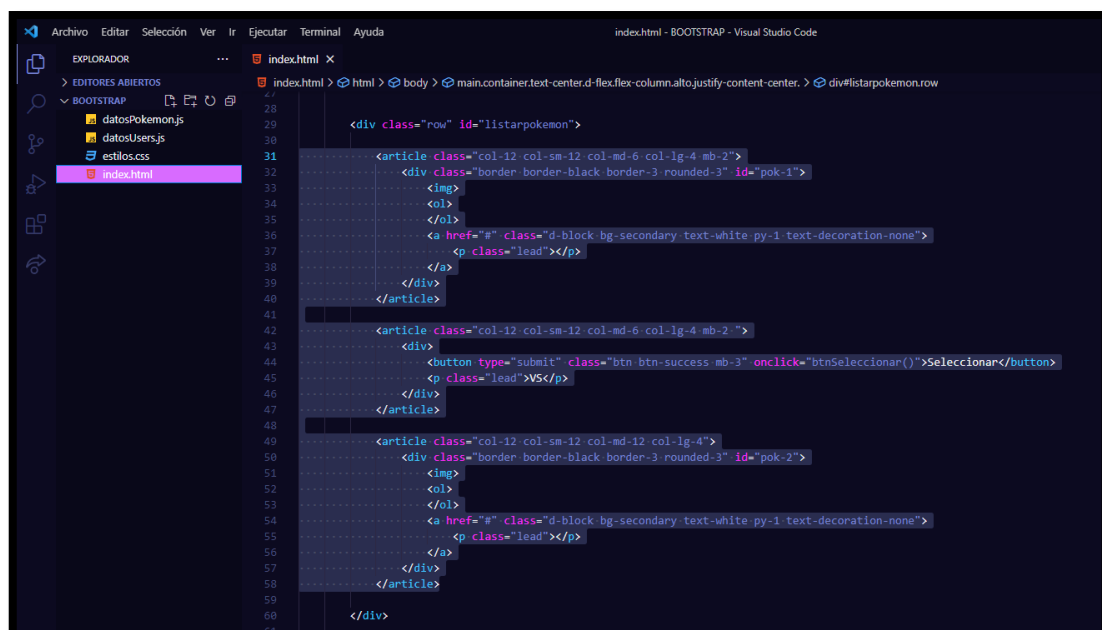


3. Se procede a trabajar en el documento HTML y crear la siguiente estructura



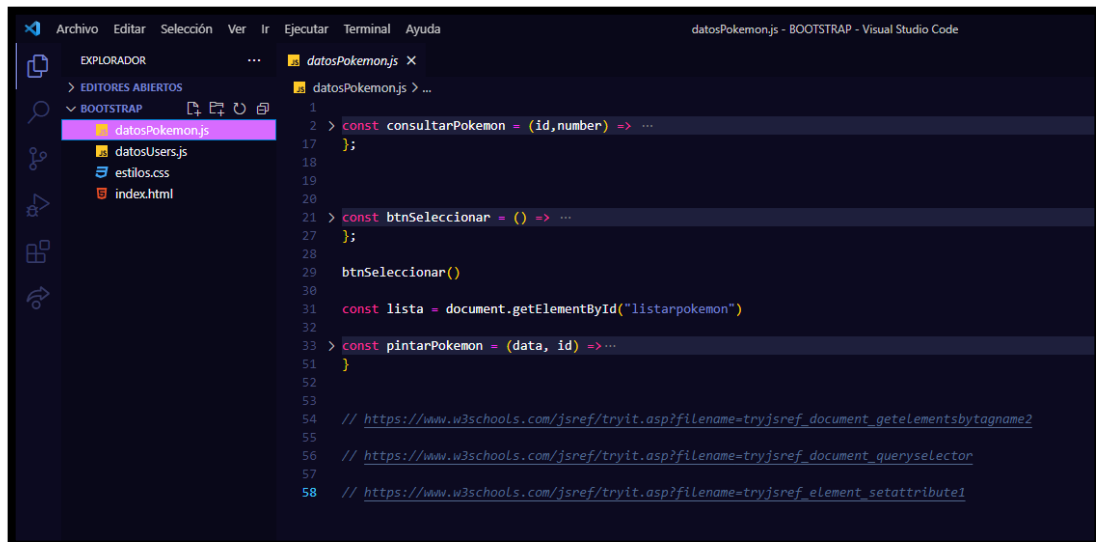
```
index.html - BOOTSTRAP - Visual Studio Code
11 </head>
12
13 <body>
14   <main class="container text-center d-flex flex-column align-items-center justify-content-center">
15     <div class="row">...
16   </div>
17
18   <div class="row id="listarpokemon">...
19 </div>
20
21   <div class="row">...
22 </div>
23
24   <div class="row">...
25 </div>
26
27   <div class="row">...
28 </div>
29
30   <div class="row">...
31 </div>
32
33   <script src="../datosPokemon.js"></script>
34   <script src="../datosUsers.js"></script>
35 </body>
36 </html>
```

4. Se procede a trabajar en el div principal



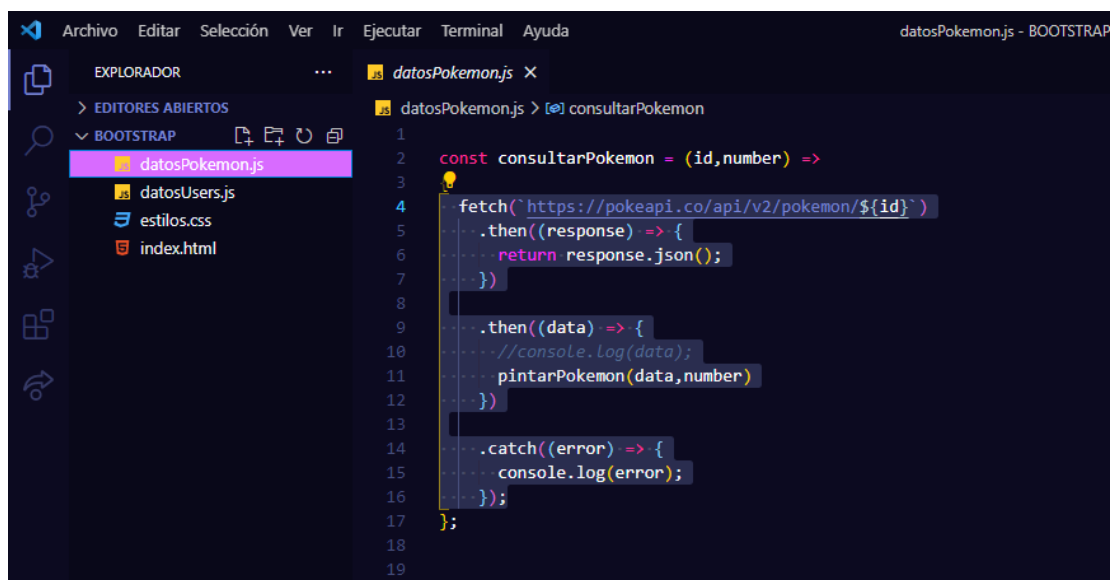
```
index.html - BOOTSTRAP - Visual Studio Code
31 <div class="row id="listarpokemon">
32   <article class="col-12 col-sm-12 col-md-6 col-lg-4 mb-2">
33     <div class="border border-black border-3 rounded-3 id="pok-1">
34       <img>
35       <ol>
36       </ol>
37       <a href="#" class="d-block bg-secondary text-white py-1 text-decoration-none">
38         <p class="lead"></p>
39       </a>
40     </div>
41   </article>
42   <article class="col-12 col-sm-12 col-md-6 col-lg-4 mb-2">
43     <div>
44       <button type="submit" class="btn btn-success mb-3" onclick="btnSeleccionar()">Seleccionar</button>
45       <p class="lead">V5</p>
46     </div>
47   </article>
48   <article class="col-12 col-sm-12 col-md-12 col-lg-4">
49     <div class="border border-black border-3 rounded-3 id="pok-2">
50       <img>
51       <ol>
52       </ol>
53       <a href="#" class="d-block bg-secondary text-white py-1 text-decoration-none">
54         <p class="lead"></p>
55       </a>
56     </div>
57   </article>
58 </div>
```

5. Ahora se procede a trabajar en la etiqueta Script y crear 3 funciones y agregar una referencia al documento HTML



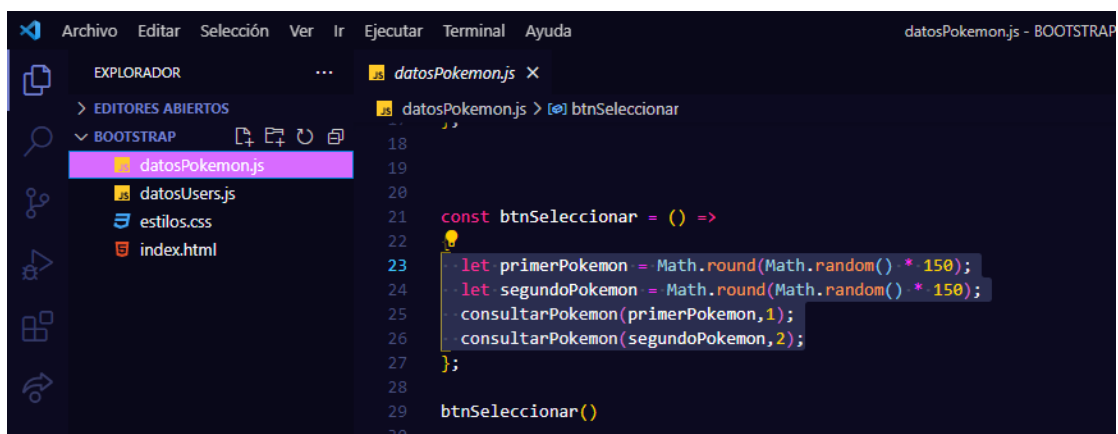
```
1 > const consultarPokemon = (id,number) => ...
17 };
18
19
20
21 > const btnSeleccionar = () => ...
27 };
28
29 btnSeleccionar()
30
31 const lista = document.getElementById("listarpokemon")
32
33 > const pintarPokemon = (data, id) =>...
51 }
52
53
54 // https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref_document_getelementsbytagname2
55
56 // https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref_document_queryselector
57
58 // https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref_element_setattribute1
```

6. Se procede a trabajar con la primera función



```
1
2 const consultarPokemon = (id,number) =>
3
4   fetch(`https://pokeapi.co/api/v2/pokemon/${id}`)
5   .then((response) => {
6     return response.json();
7   })
8
9   .then((data) => {
10     //console.log(data);
11     pintarPokemon(data,number)
12   })
13
14   .catch((error) => {
15     console.log(error);
16   });
17
18
19
```

7. Se procede a trabajar con la segunda función

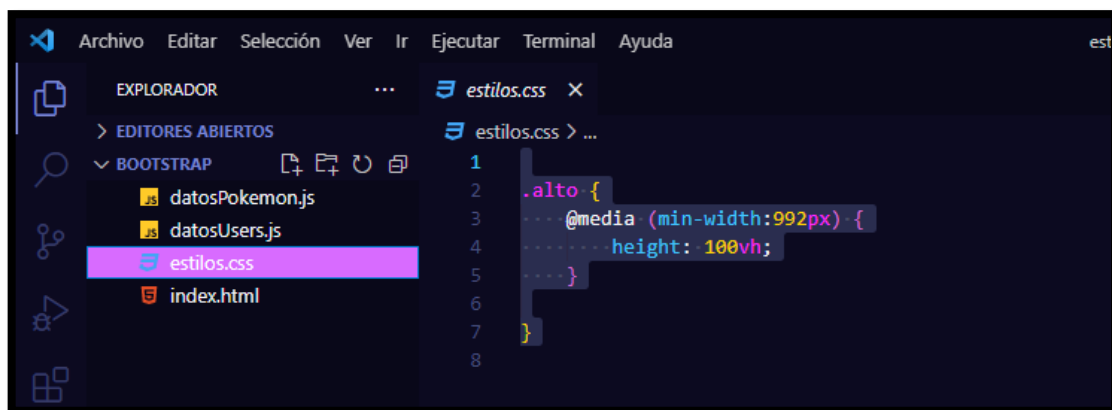


```
18
19
20
21 const btnSeleccionar = () =>
22
23   let primerPokemon = Math.round(Math.random() * 150);
24   let segundoPokemon = Math.round(Math.random() * 150);
25   consultarPokemon(primerPokemon,1);
26   consultarPokemon(segundoPokemon,2);
27 };
28
29 btnSeleccionar()
30
```

8. Se procede a trabajar con la tercera función



9. Se procede a trabajar en la estructura del CSS



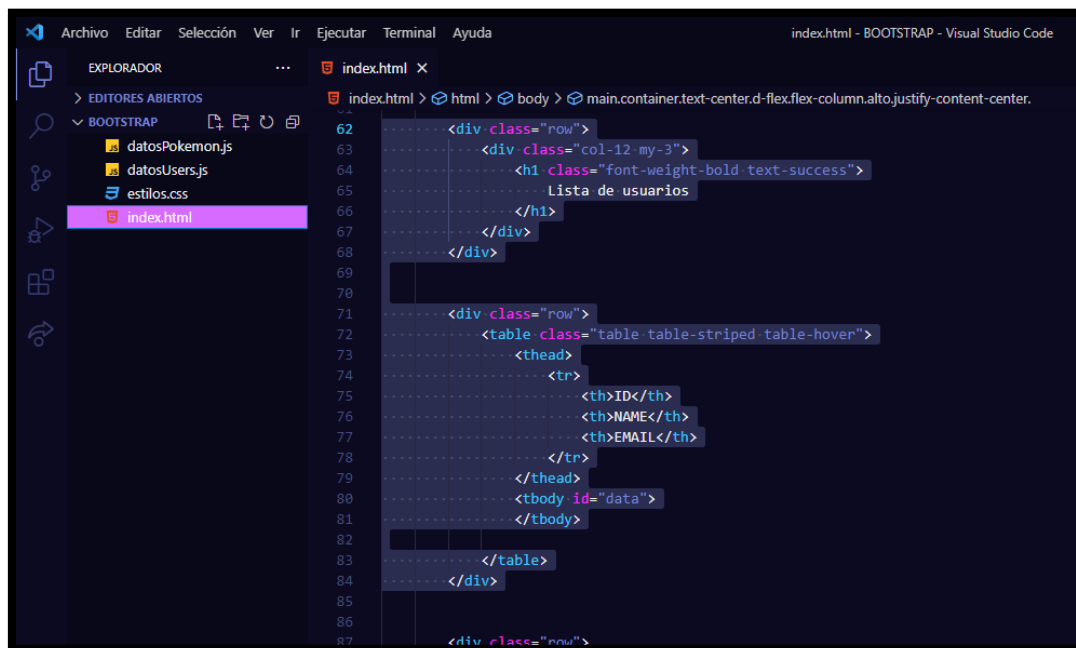
10. Verificar



11. Ahora se procede a consumir una API de JSONPlaceholder

API DE JSONPLACEHOLDER

12. Se procede a trabajar en la estructura del HTML



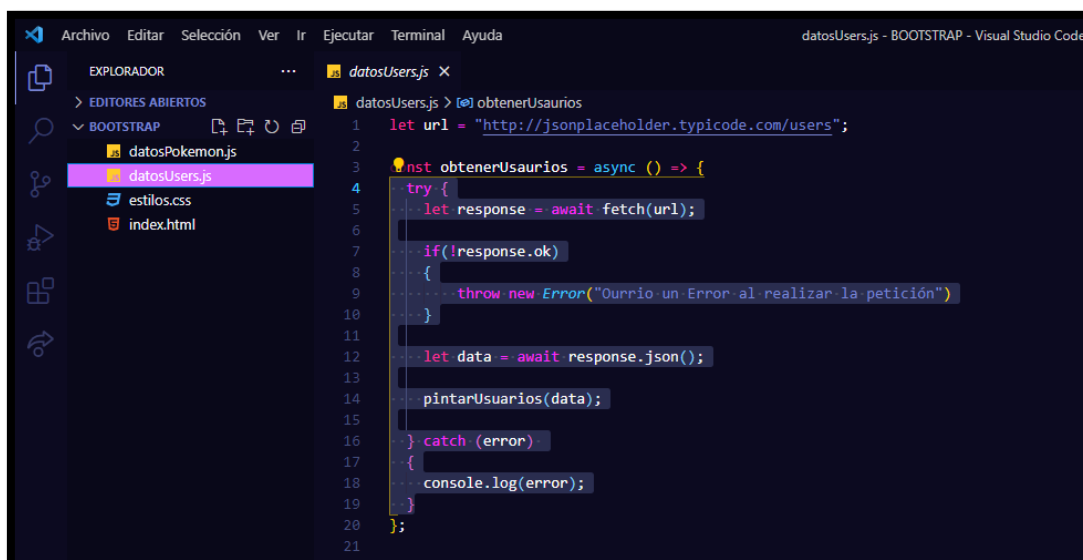
```
index.html > html > body > main.container.text-center.d-flex.flex-column.alto.justify-content-center.  
62 .....<div class="row">  
63 .....<div class="col-12 my-3">  
64 .....<h1 class="font-weight-bold text-success">  
65 .....Lista de usuarios  
66 .....</h1>  
67 .....</div>  
68 .....</div>  
69 .....  
70 .....  
71 .....<div class="row">  
72 .....<table class="table table-striped table-hover">  
73 .....<thead>  
74 .....<tr>  
75 .....<th>ID</th>  
76 .....<th>NAME</th>  
77 .....<th>EMAIL</th>  
78 .....</tr>  
79 .....</thead>  
80 .....<tbody id="data">  
81 .....</tbody>  
82 .....  
83 .....</table>  
84 .....</div>  
85 .....  
86 .....  
87 .....<div class="row">
```

13. Ahora se programa las dos funciones y en una de ellas utilizando async y await



```
datosUsers.js > ...  
1 let url = "http://jsonplaceholder.typicode.com/users";  
2  
3 > const obtenerUsuarios = async () => { ...  
20 };  
21  
22 obtenerUsuarios();  
23  
24  
25 > const pintarUsuarios = (data) => ...  
34 };  
35
```

14. Programar la primera función



```
datosUsers.js > obtenerUsuarios  
1 let url = "http://jsonplaceholder.typicode.com/users";  
2  
3 const obtenerUsuarios = async () => {  
4 try {  
5 let response = await fetch(url);  
6  
7 if(!response.ok)  
8 {  
9 throw new Error("Ocurrio un Error al realizar la petición")  
10 }  
11  
12 let data = await response.json();  
13  
14 pintarUsuarios(data);  
15  
16 } catch (error)  
17 {  
18 console.log(error);  
19 }  
20 };  
21
```

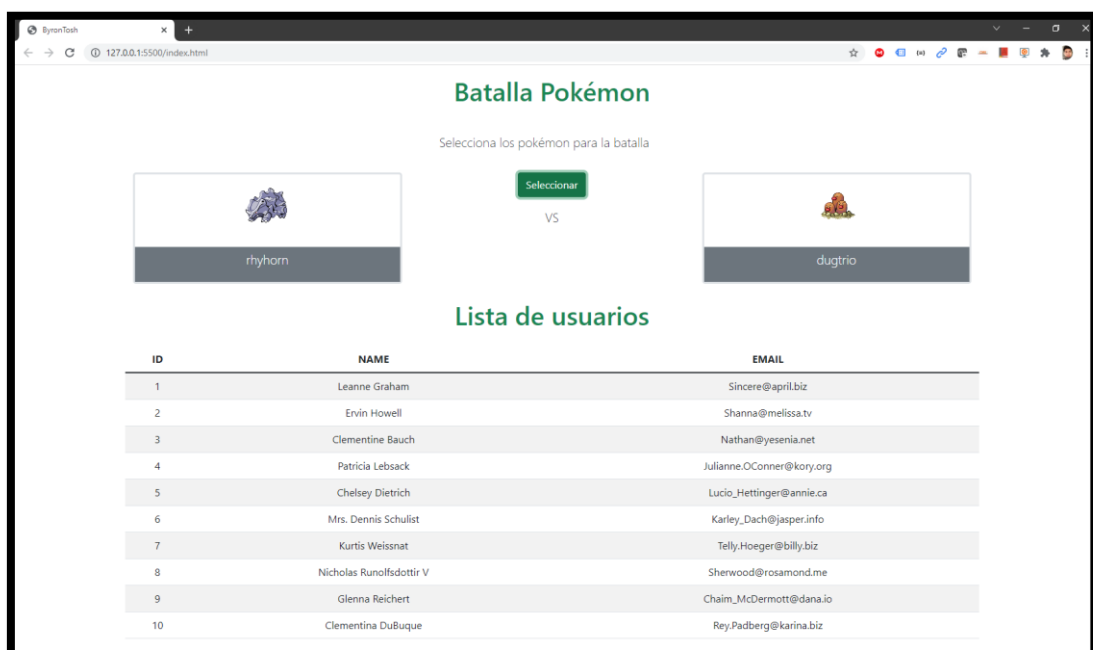
15. Programar la segunda función

```

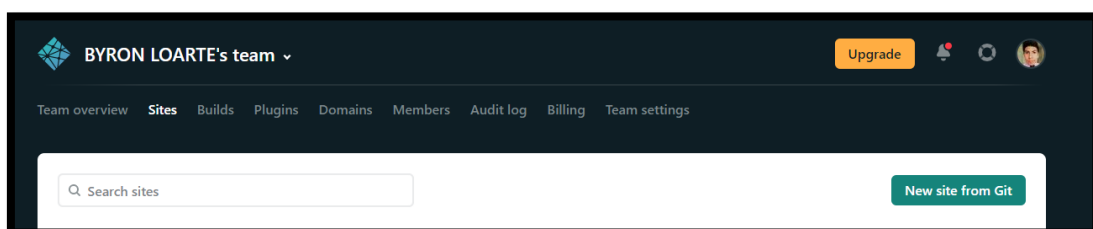
22
23 obtenerUsuarios();
24
25 const pintarUsuarios = (data) => {
26
27   let body = "";
28   for (let i = 0; i < data.length; i++) {
29     body += `<tr><td>${data[i].id}</td><td>${data[i].name}</td><td>${data[i].email}</td></tr>`;
30   }
31
32   document.getElementById("data").innerHTML = body;
33
34 }
35

```

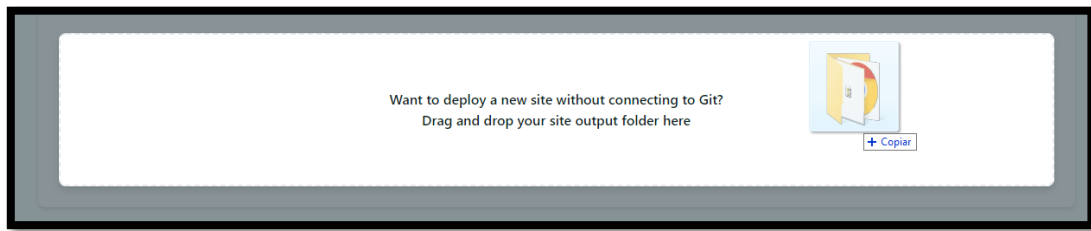
16. Visualizar los resultados



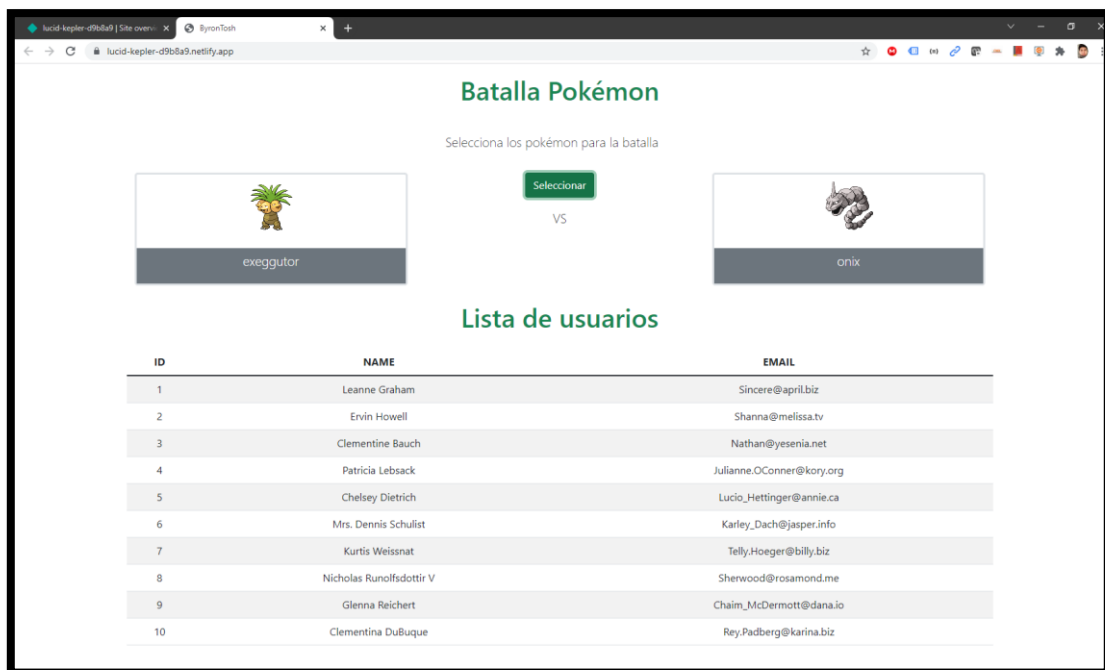
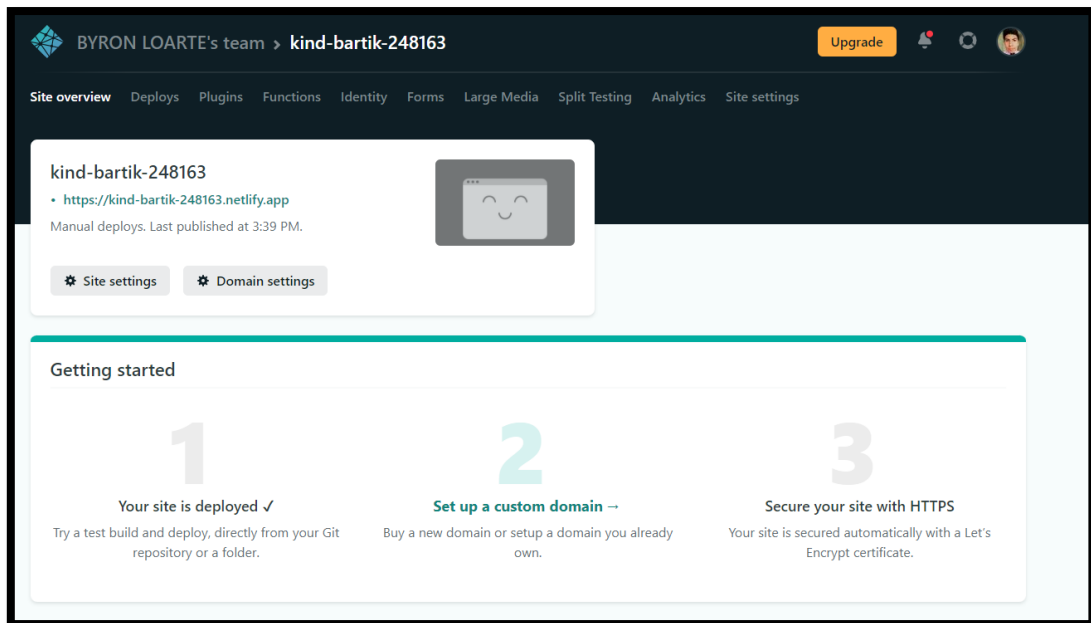
17. Ahora se procede a subir el proyecto en NETLIFY, para lo cual se procede a crear una cuenta y luego dar clic en la pestaña sitios



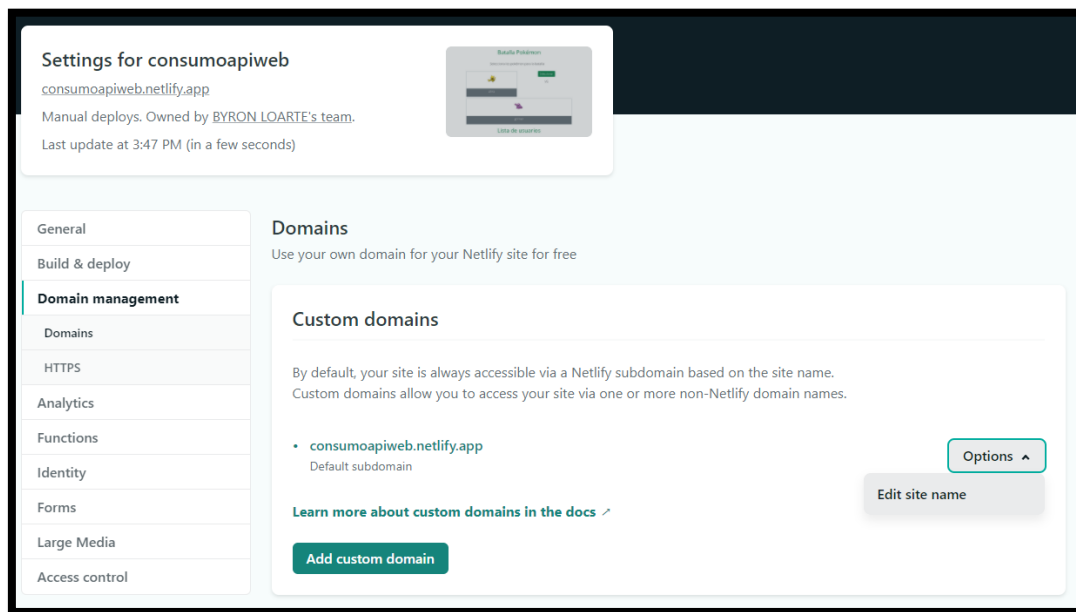
18. Luego arrastrar y soltar la carpeta del proyecto



19. A continuación, se procede a dar clic en la **URL** proporcionado

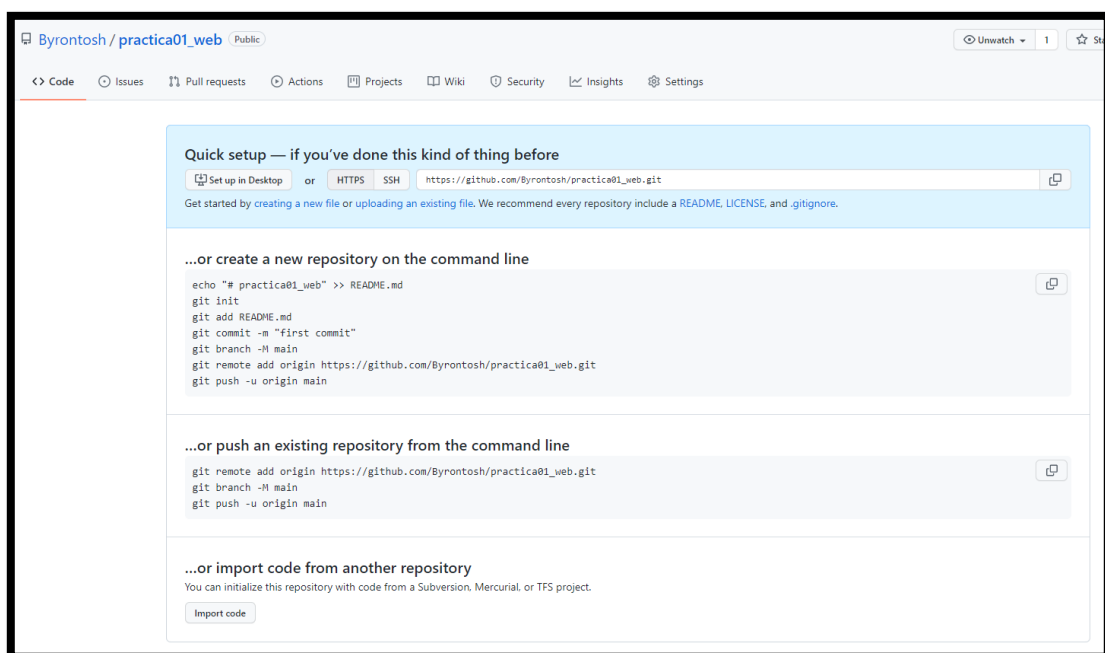


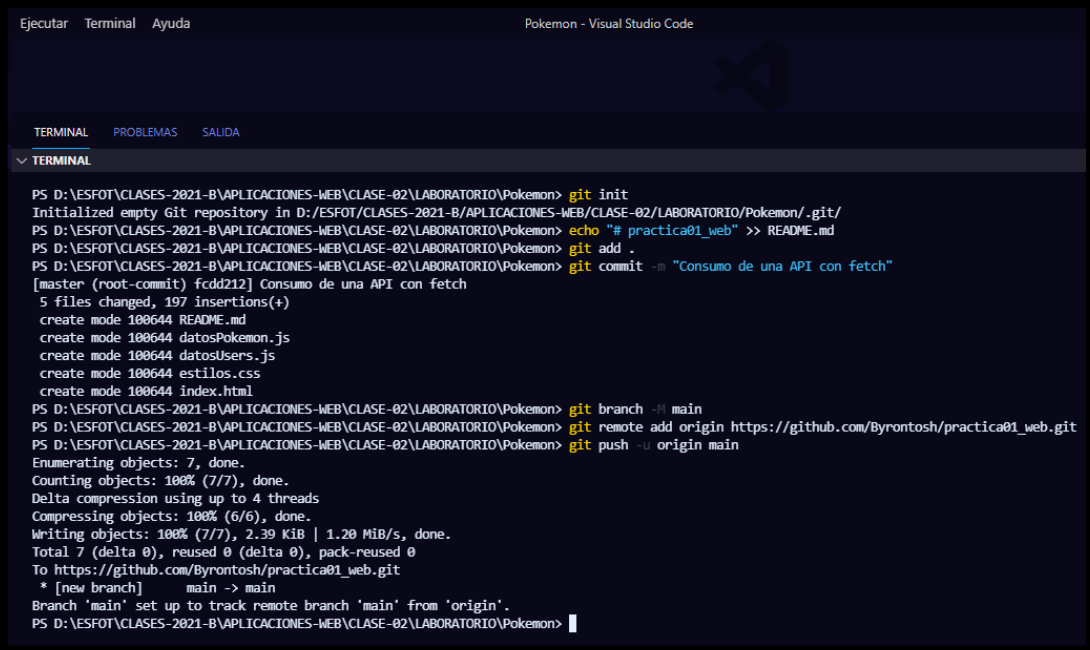
20. Si se desea se puede modificar el subdominio



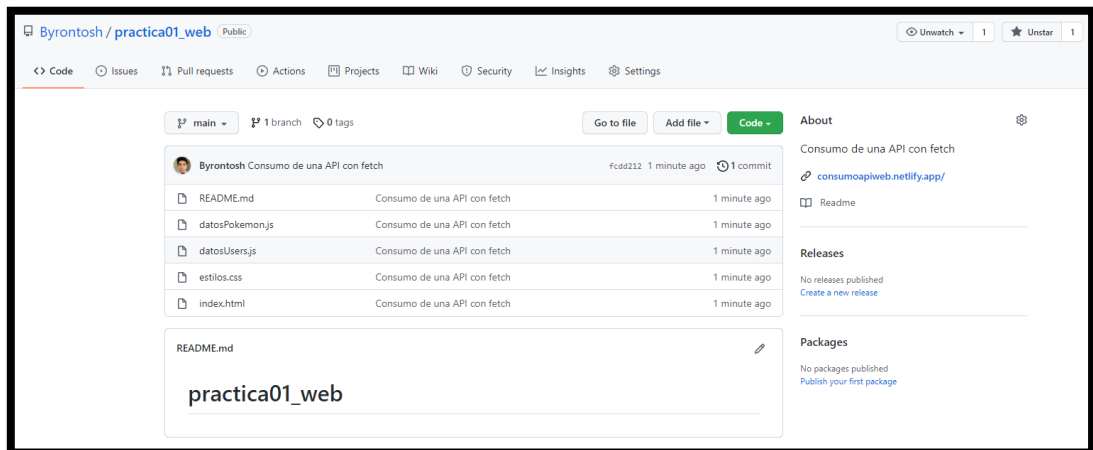
<https://consumoapiweb.netlify.app/>

21. Ahora se procede a subir en un repositorio en GitHub





```
PS D:\ESFOT\CLASES-2021-B\APLICACIONES-WEB\CLASE-02\LABORATORIO\Pokemon> git init
Initialized empty Git repository in D:/ESFOT/CLASES-2021-B/APLICACIONES-WEB/CLASE-02/LABORATORIO/Pokemon/.git/
PS D:\ESFOT\CLASES-2021-B\APLICACIONES-WEB\CLASE-02\LABORATORIO\Pokemon> echo "# practica01_web" >> README.md
PS D:\ESFOT\CLASES-2021-B\APLICACIONES-WEB\CLASE-02\LABORATORIO\Pokemon> git add .
PS D:\ESFOT\CLASES-2021-B\APLICACIONES-WEB\CLASE-02\LABORATORIO\Pokemon> git commit -m "Consumo de una API con fetch"
[master (root-commit) fcd212] Consumo de una API con fetch
5 files changed, 197 insertions(+)
create mode 100644 README.md
create mode 100644 datosPokemon.js
create mode 100644 datosUsers.js
create mode 100644 estilos.css
create mode 100644 index.html
PS D:\ESFOT\CLASES-2021-B\APLICACIONES-WEB\CLASE-02\LABORATORIO\Pokemon> git branch -M main
PS D:\ESFOT\CLASES-2021-B\APLICACIONES-WEB\CLASE-02\LABORATORIO\Pokemon> git remote add origin https://github.com/Byrontosh/practica01_web.git
PS D:\ESFOT\CLASES-2021-B\APLICACIONES-WEB\CLASE-02\LABORATORIO\Pokemon> git push -u origin main
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 4 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (7/7), 2.39 KiB | 1.20 MiB/s, done.
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Byrontosh/practica01_web.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
PS D:\ESFOT\CLASES-2021-B\APLICACIONES-WEB\CLASE-02\LABORATORIO\Pokemon>
```



https://github.com/Byrontosh/practica01_web