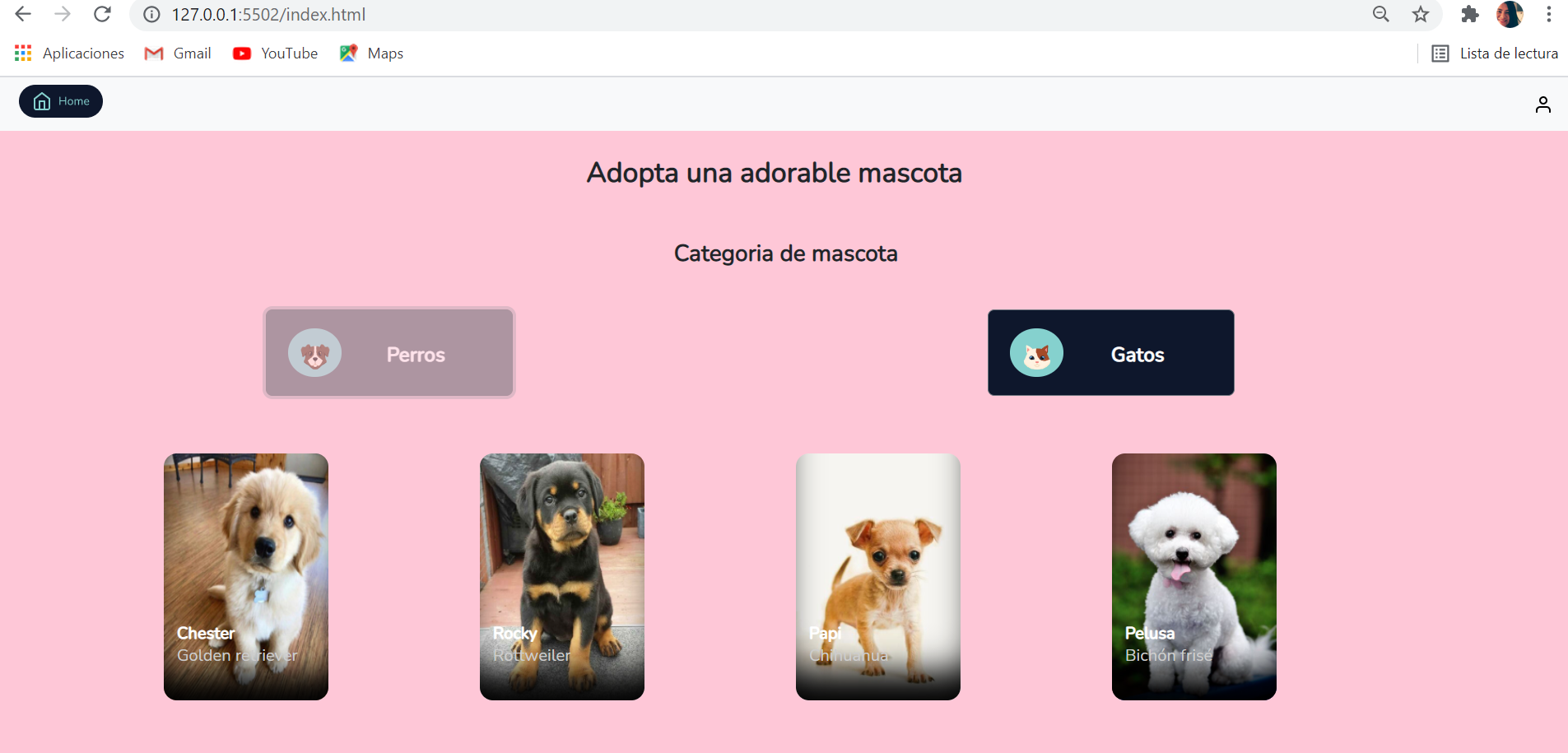
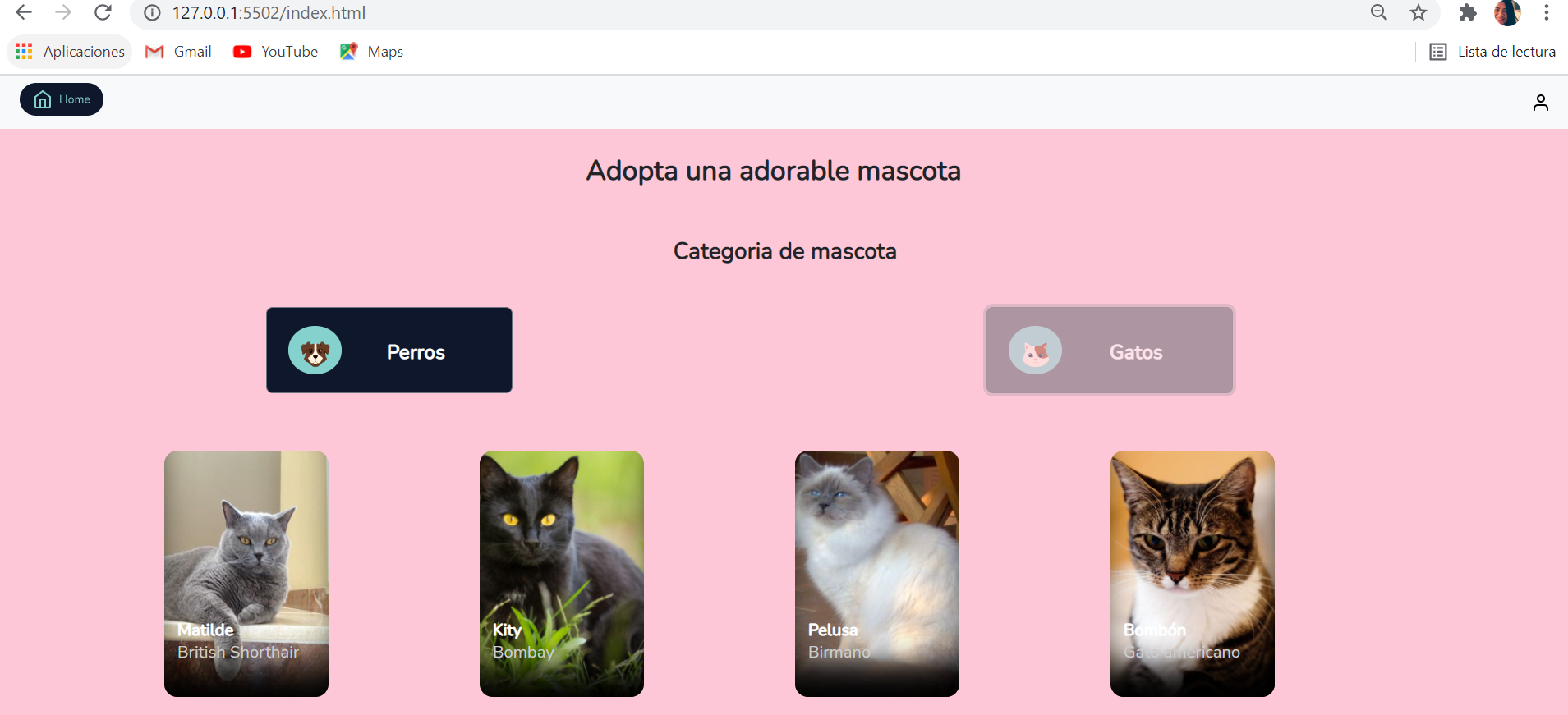
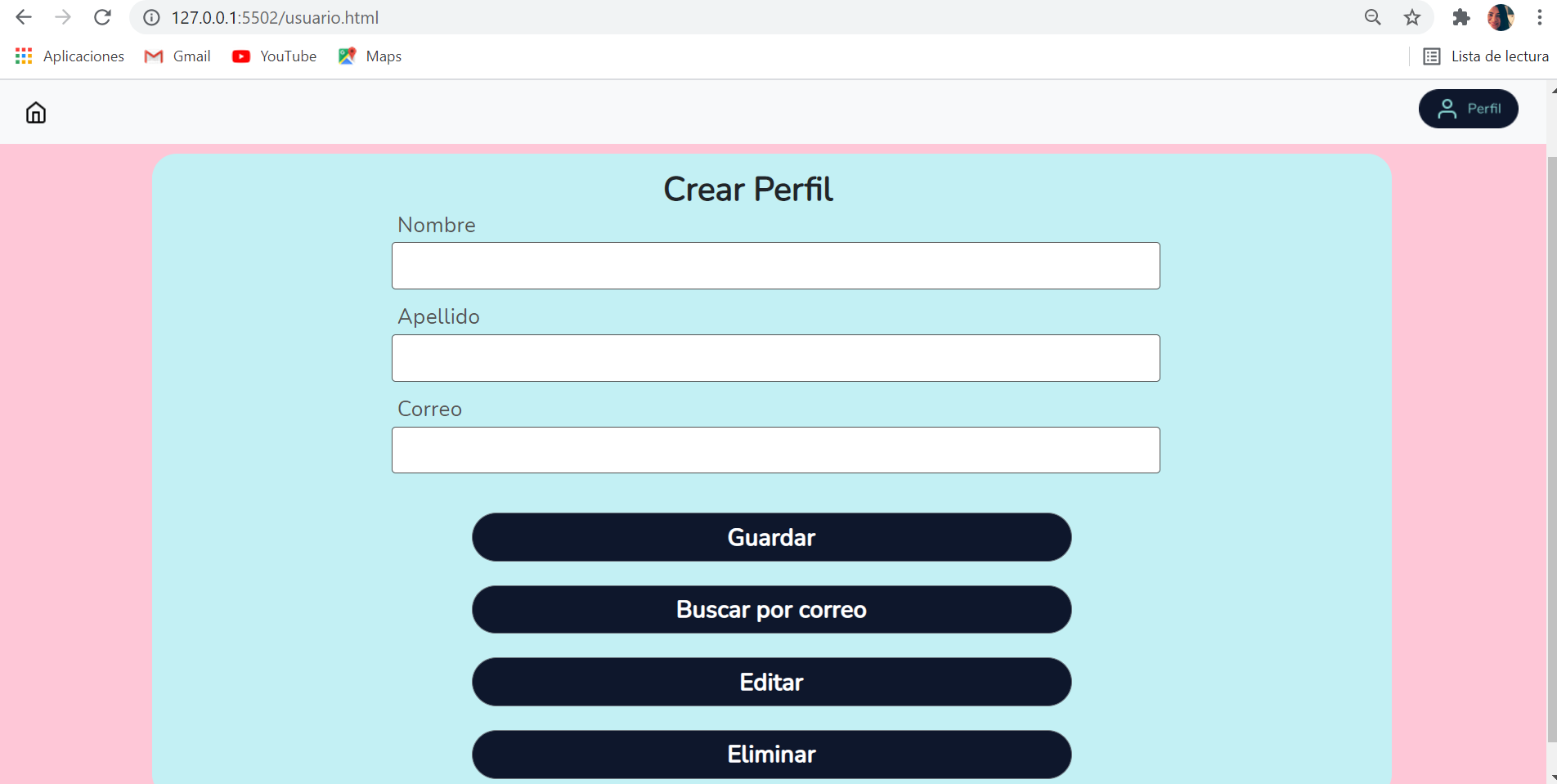
App Mascotas

index.html categoría Perros



Index.html categoría Gatos

usuario.html



Creamos nuestra estructura de trabajo

Creamos un archivo index.html y otro llamado usuario.html

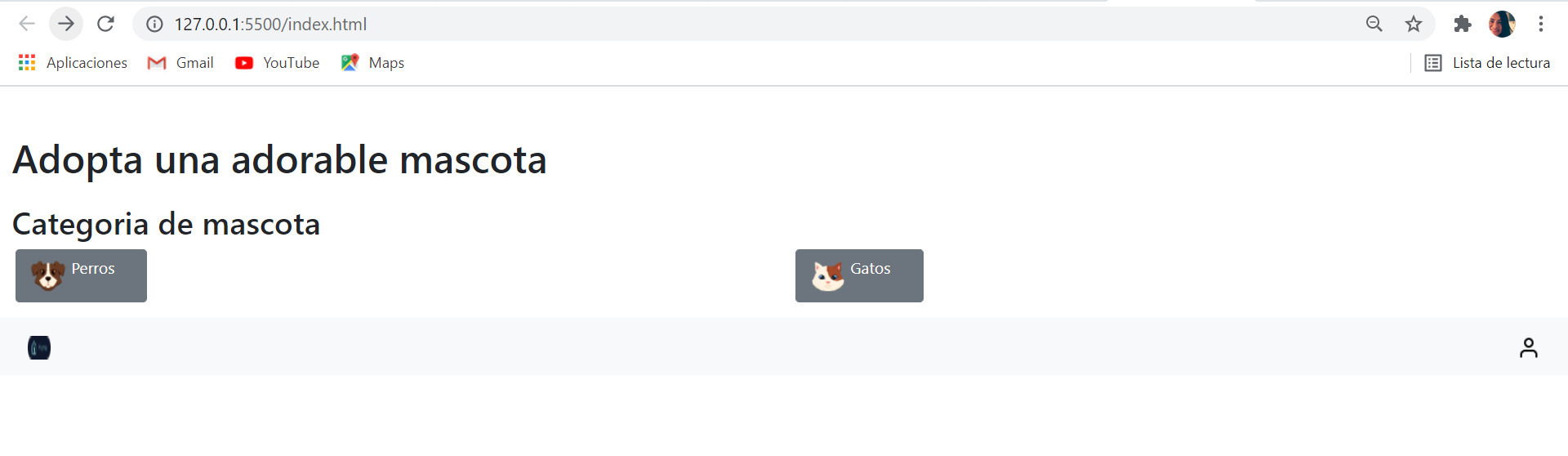
Dentro de index.html incorporamos el siguiente código

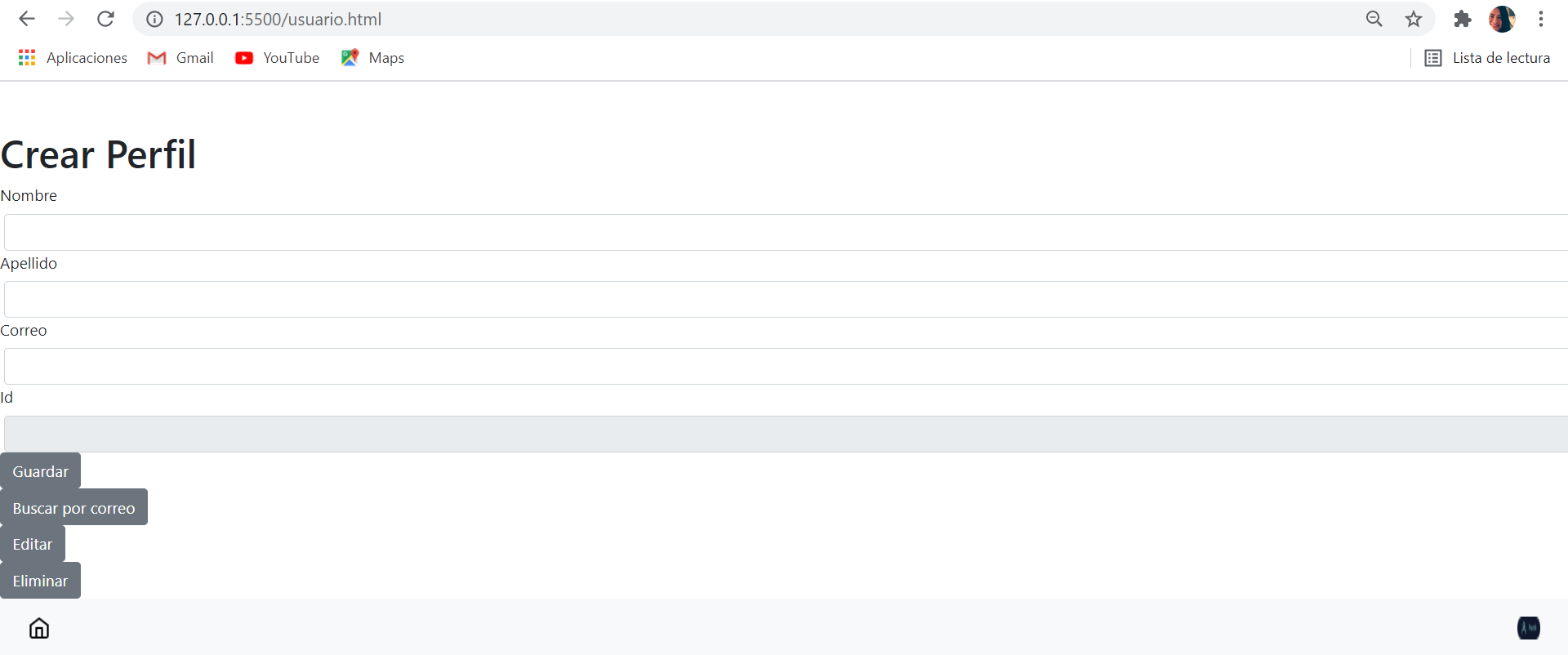
<https://raw.githubusercontent.com/jennymontoya1001/ArchivosmascotasAppVanilla/main/index.html>

Dentro de usuario.html incorporamos el siguiente código

<https://raw.githubusercontent.com/jennymontoya1001/ArchivosmascotasAppVanilla/main/usuario.html>

Se debe visualizar de la siguiente manera





Incorporamos nuestros estilos

Creamos en la raíz una carpeta que se llame styles y dentro de ella creamos tres archivos

index1.css

<https://raw.githubusercontent.com/jennymontoya1001/ArchivosmascotasAppVanilla/main/index1.css>

index2.css

<https://raw.githubusercontent.com/jennymontoya1001/ArchivosmascotasAppVanilla/main/index2.css>

usuario.css

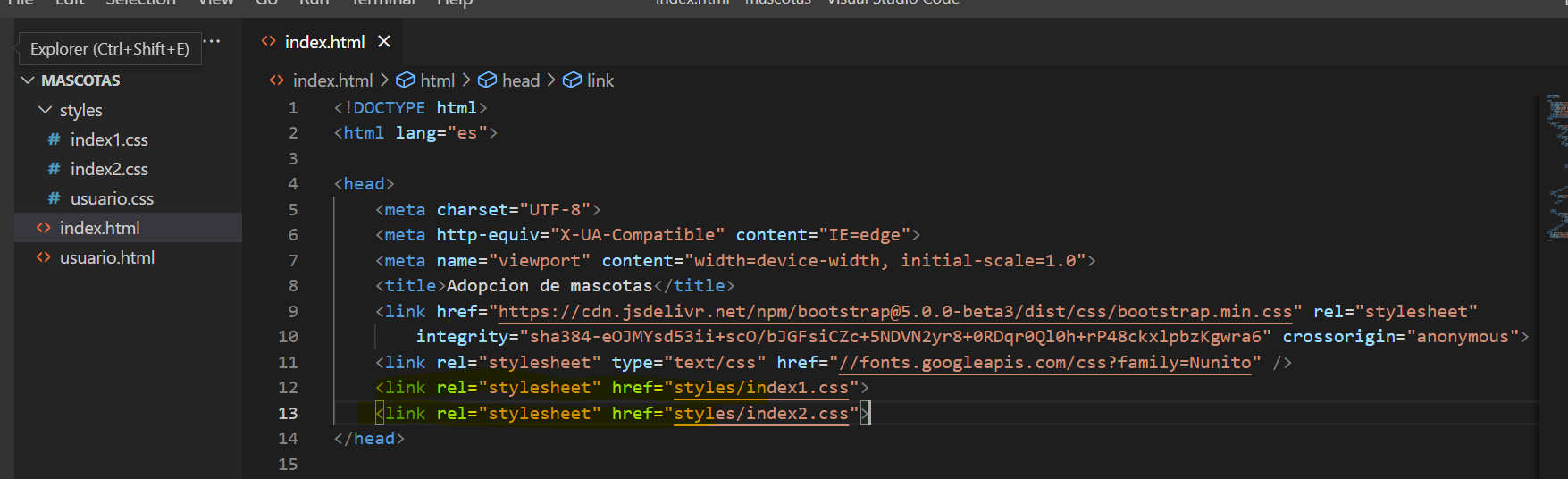
<https://raw.githubusercontent.com/jennymontoya1001/ArchivosmascotasAppVanilla/main/usuario.css>

vinculamos los archivos de estilos con nuestros html

En el archivo index.html

  <link rel="stylesheet" href="styles/index1.css">

    <link rel="stylesheet" href="styles/index2.css">

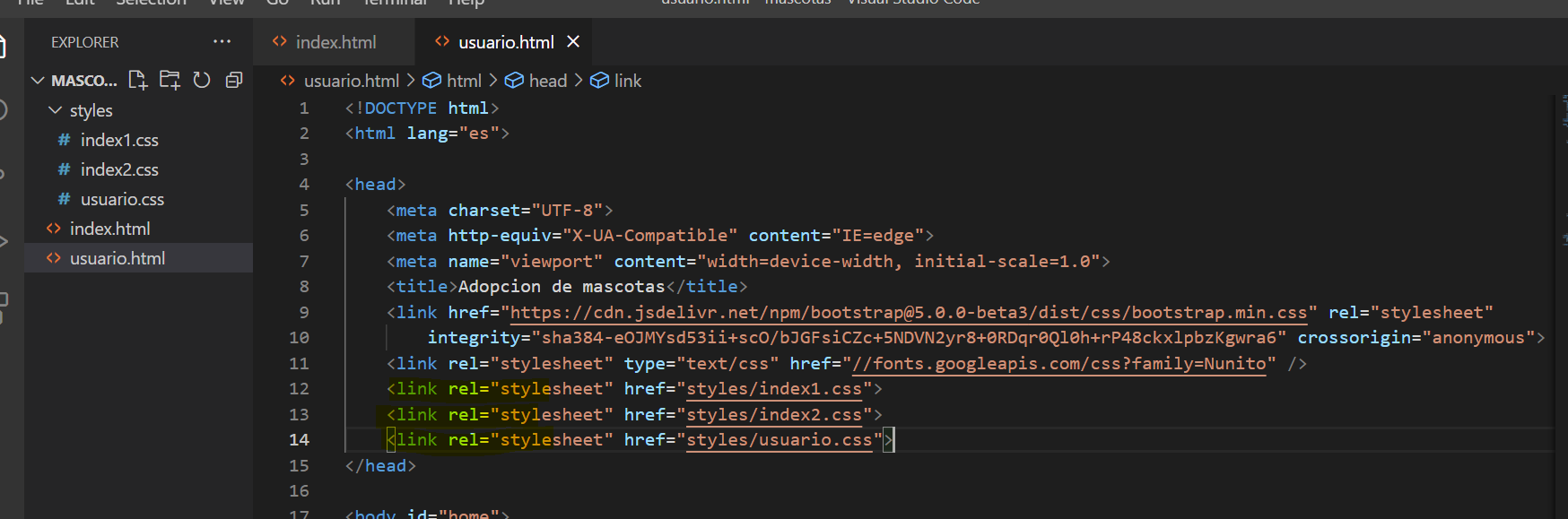


En el archive usuario.html

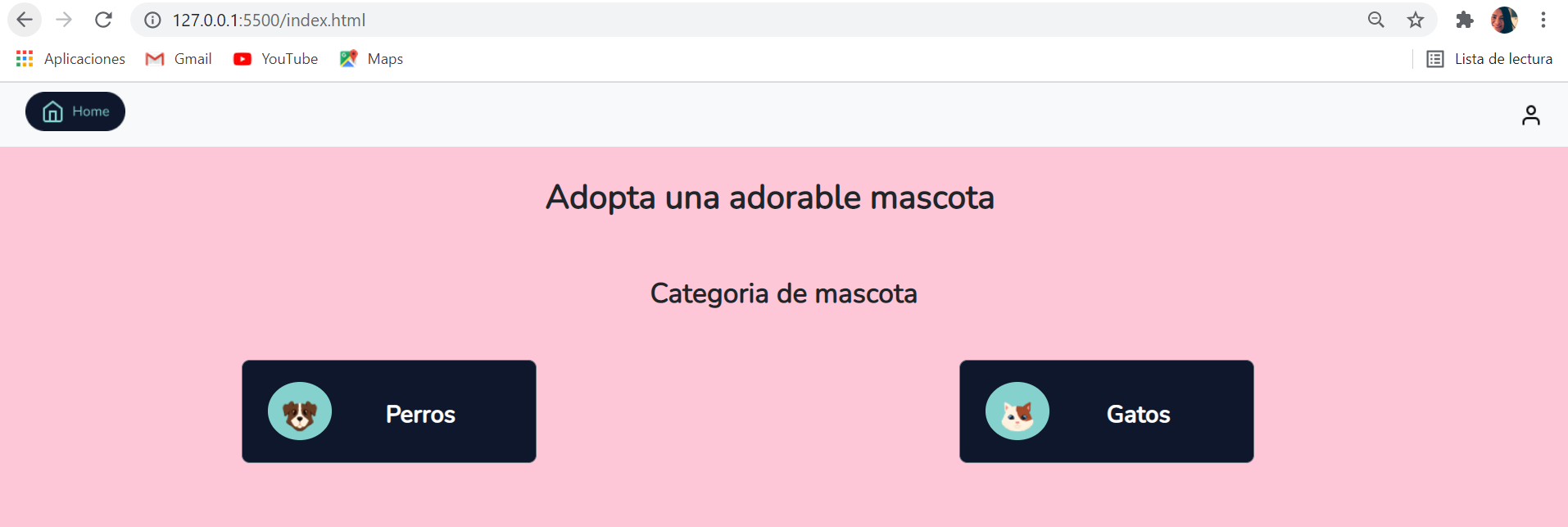
  <link rel="stylesheet" href="styles/index1.css">

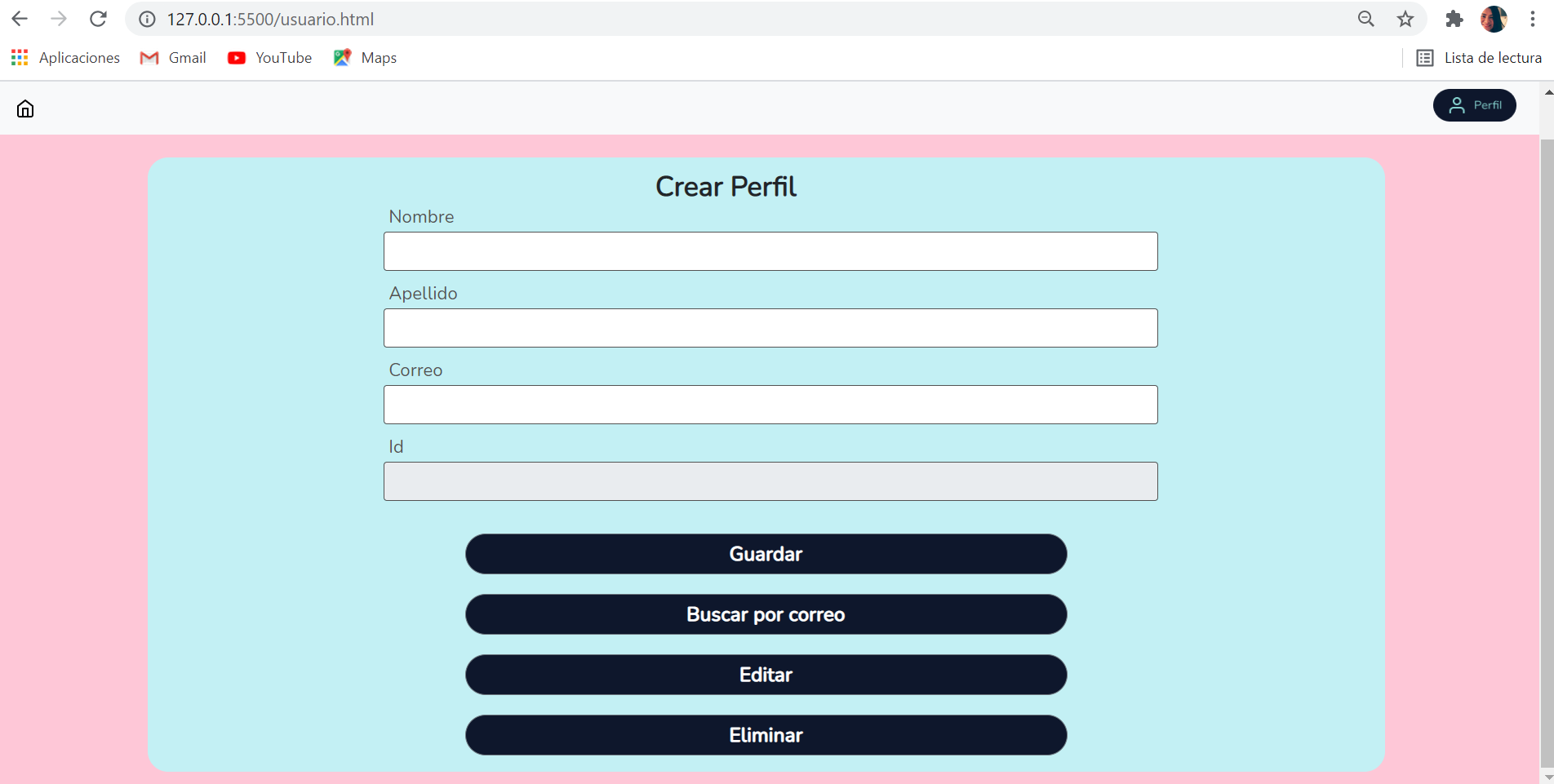
    <link rel="stylesheet" href="styles/index2.css">

    <link rel="stylesheet" href="styles/usuario.css">



Probamos en nuestro navegador y se debe visualizar de la siguiente manera





Si yo hago clic en la categoría Gatos se debe visualizar la data de los gatos, de igual manera sucede con los perros

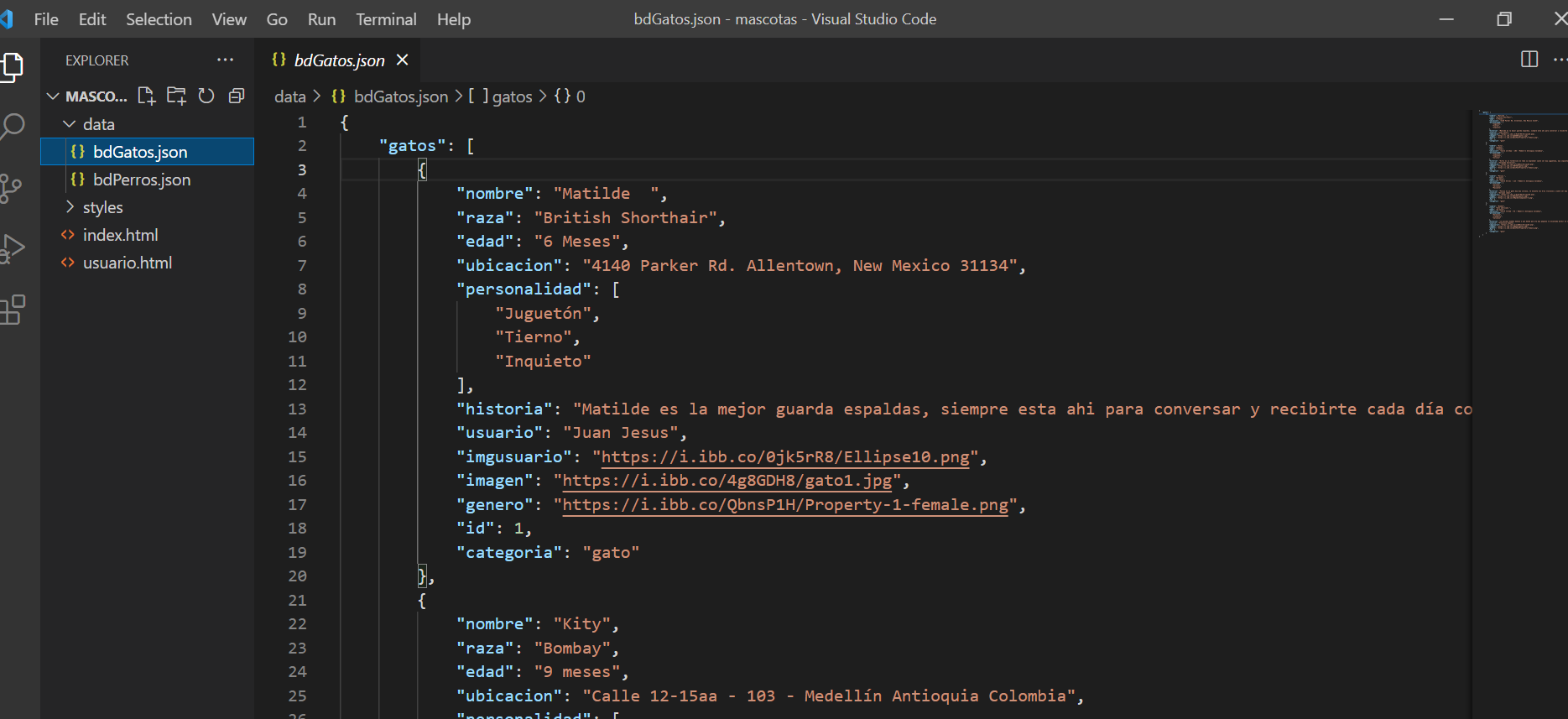
Vamos a construir nuestra data en unos archivos json y vamos a crear un servidor local con json server

Empecemos

En la raíz creamos una carpeta llamada data y dentro de ella los siguientes archivos y relacionamos la siguiente información

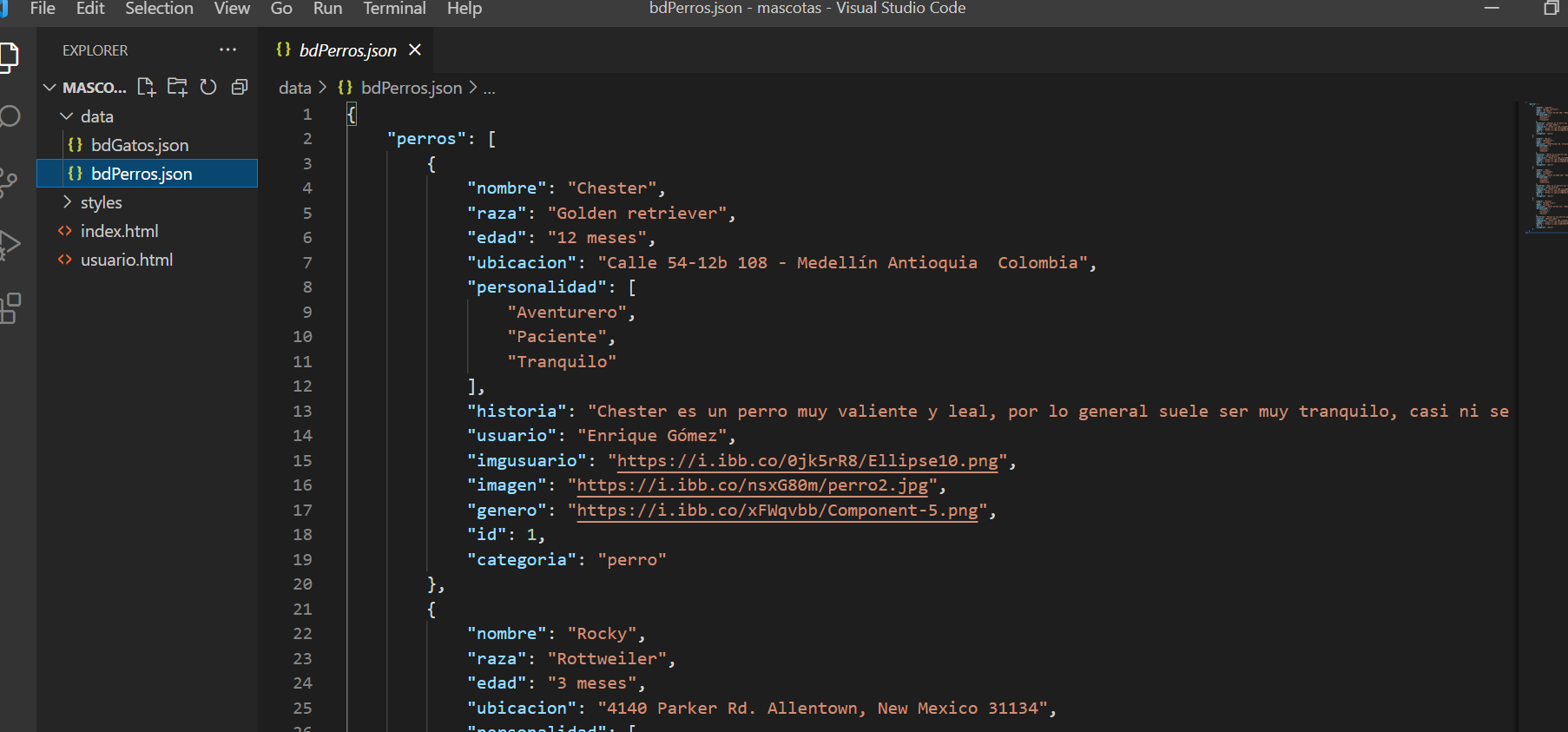
Archivo bdGatos.json

<https://raw.githubusercontent.com/jennymontoya1001/ArchivosmascotasAppVanilla/main/bdGatos.json>



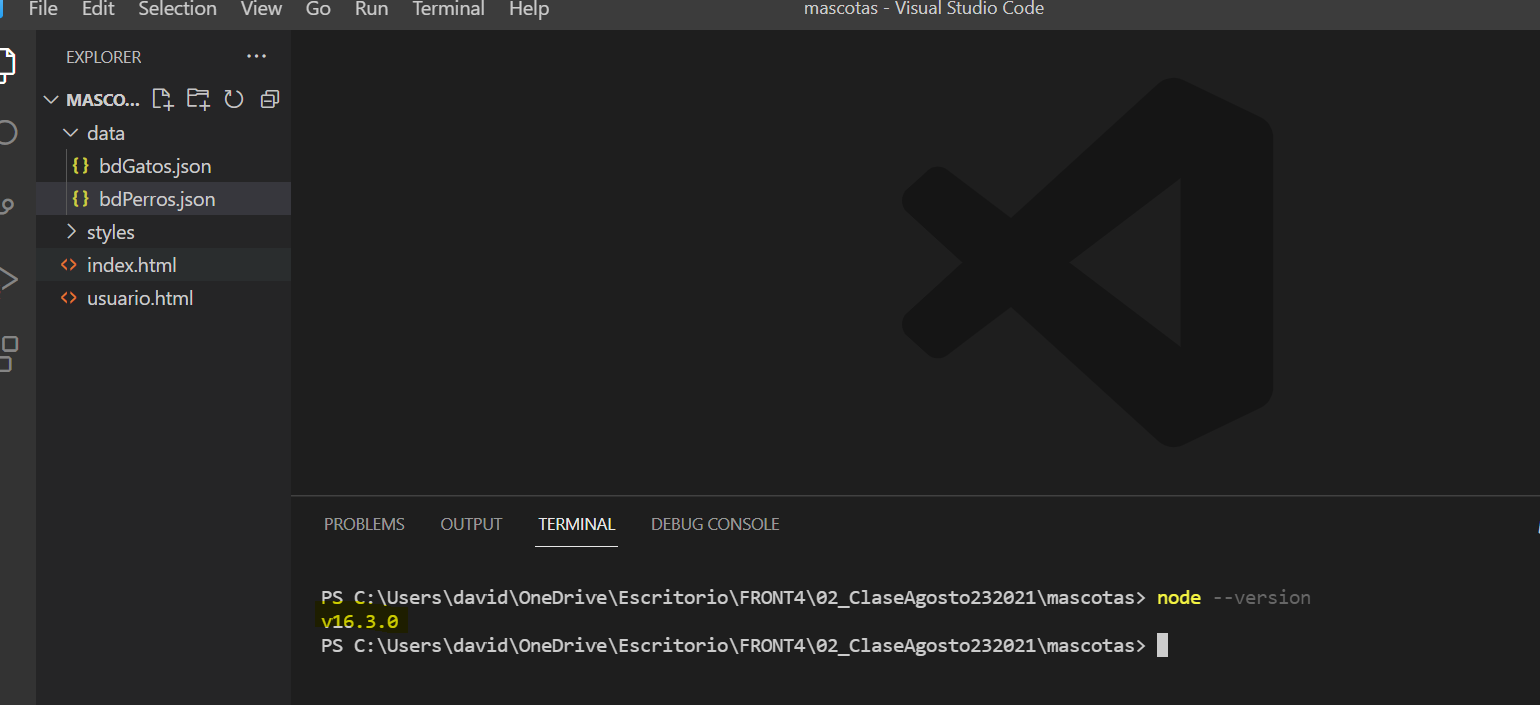
Archivo bdPerros.json

<https://raw.githubusercontent.com/jennymontoya1001/ArchivosmascotasAppVanilla/main/bdPerros.json>



Ahora sí, levantemos nuestro servidor API REST FAKE con json server

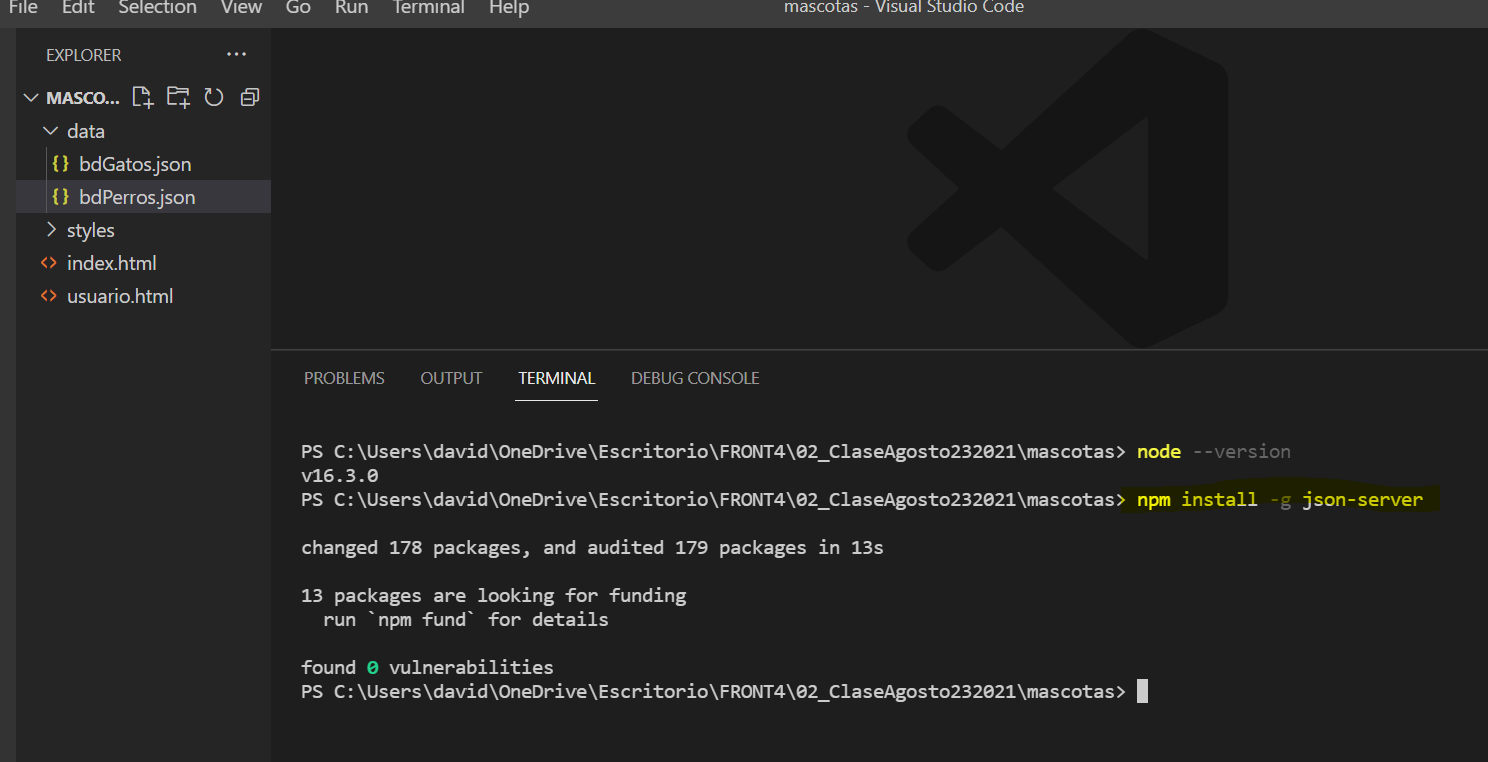
Verificamos si tenemos instalado node.js



npm json server instalación <https://www.npmjs.com/package/json-server>

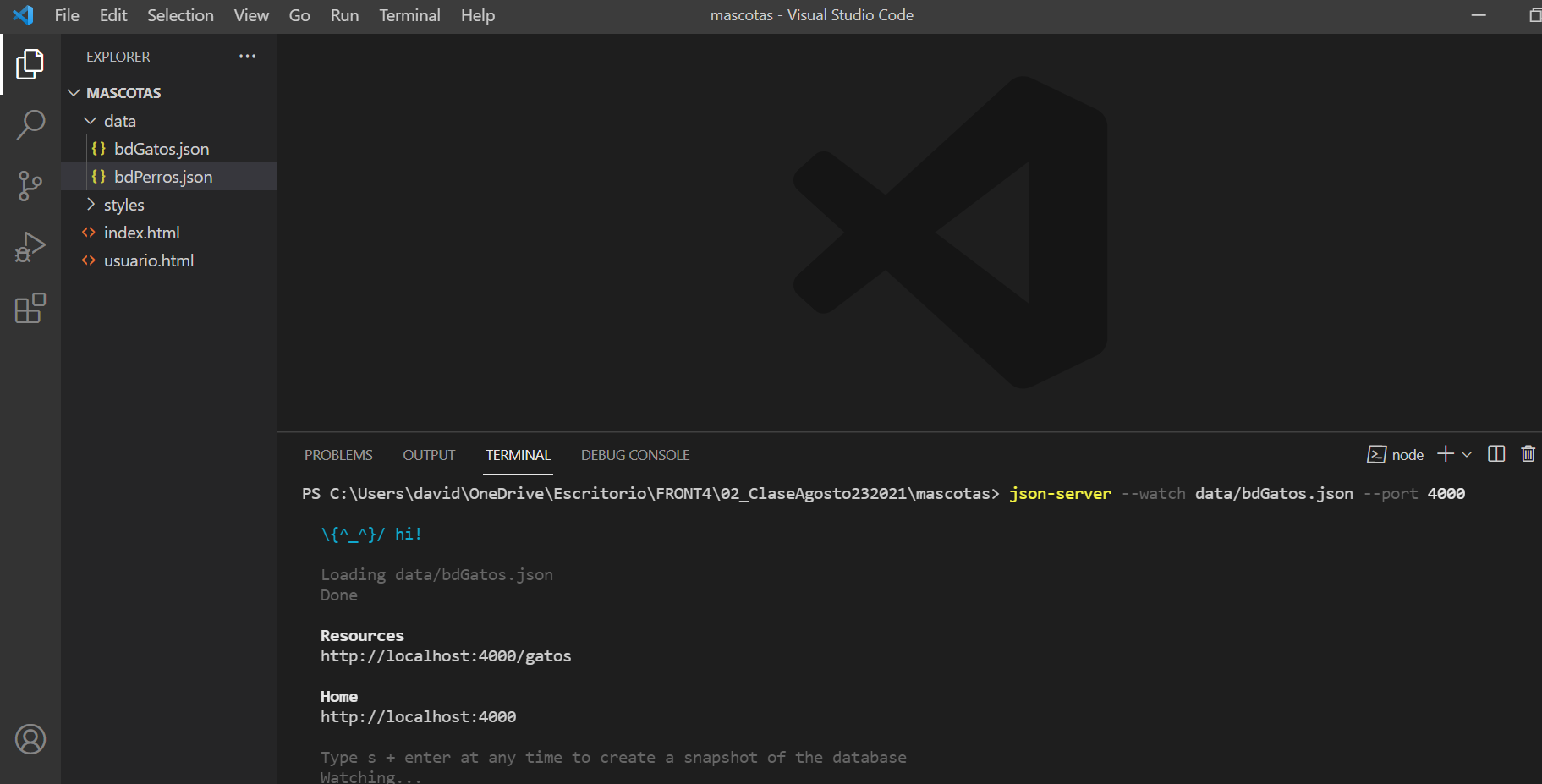
Instalamos la dependencia json server de forma global

npm install -g json-server



luego lo iniciamos relacionando el nombre del archivo que contiene la data y el puerto. Al generar esto nos proporciona una endpoint para consumir. Relacionamos el archivo bdGatos.json

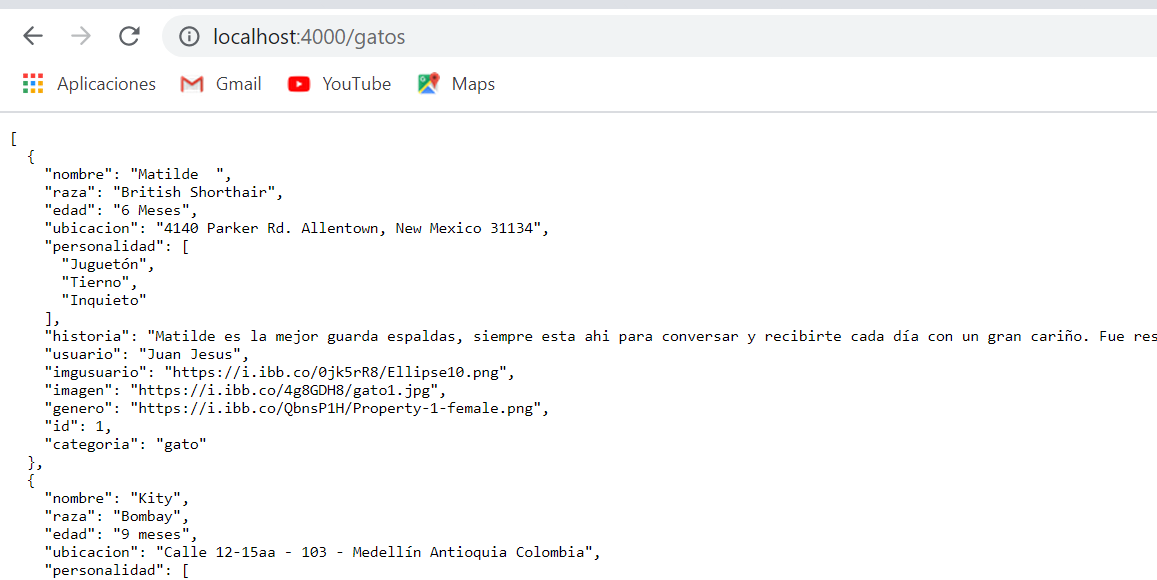
json-server --watch data/bdGatos.json --port 4000



Nos genera la endpoint de gatos que podemos consumir. Relacionada en Resources

<http://localhost:4000/gatos>

Nuestro navegador realiza una petición tipo GET – lectura de datos

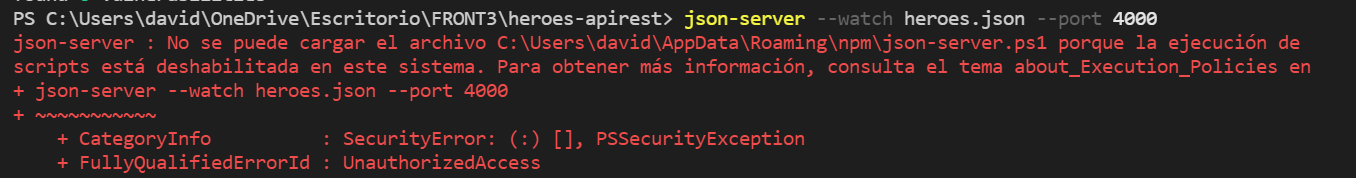


-----------------------------------------------------------------------------------------------------------------------------------

Si falla es porque está deshabilitada la ejecución de scripts. Ingresamos a la siguiente página Y buscamos el error e implementamos la solución que nos brinda

<https://www.blai.blog/2019/05/habilitar-ejecucion-de-archivos-ps1-en.html>

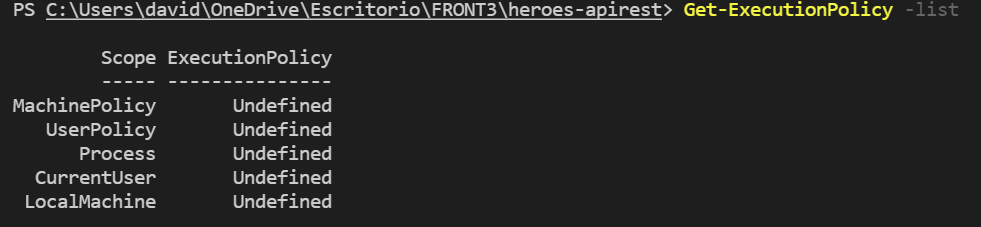
Error:



Verificación:

Corremos la siguiente instrucción

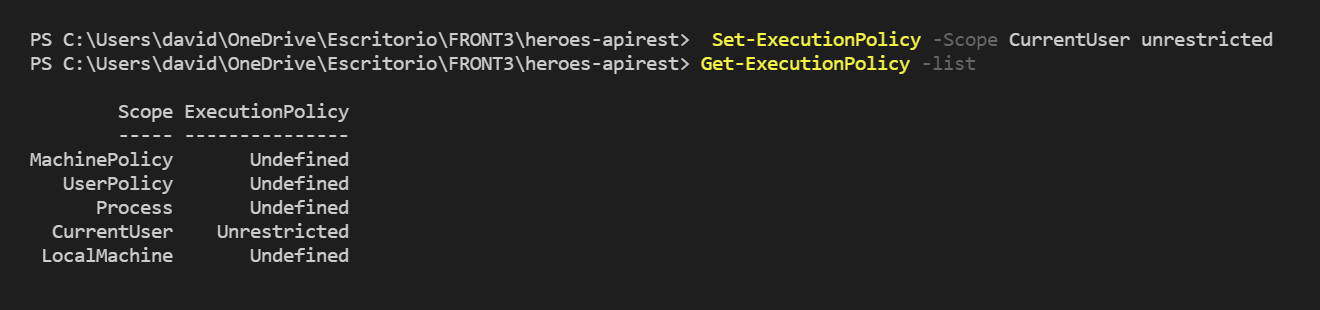
Get-ExecutionPolicy –list y vemos como está la configuración de scripts de power Shell.



Solución:

Debemos configurar como no restringido la opción de CurrentUser y corremos la siguiente instrucción Set-ExecutionPolicy -Scope CurrentUser unrestricted y luego esta Get-ExecutionPolicy -list

Y verificamos que el CurrentUser se encuentra sin restricciones

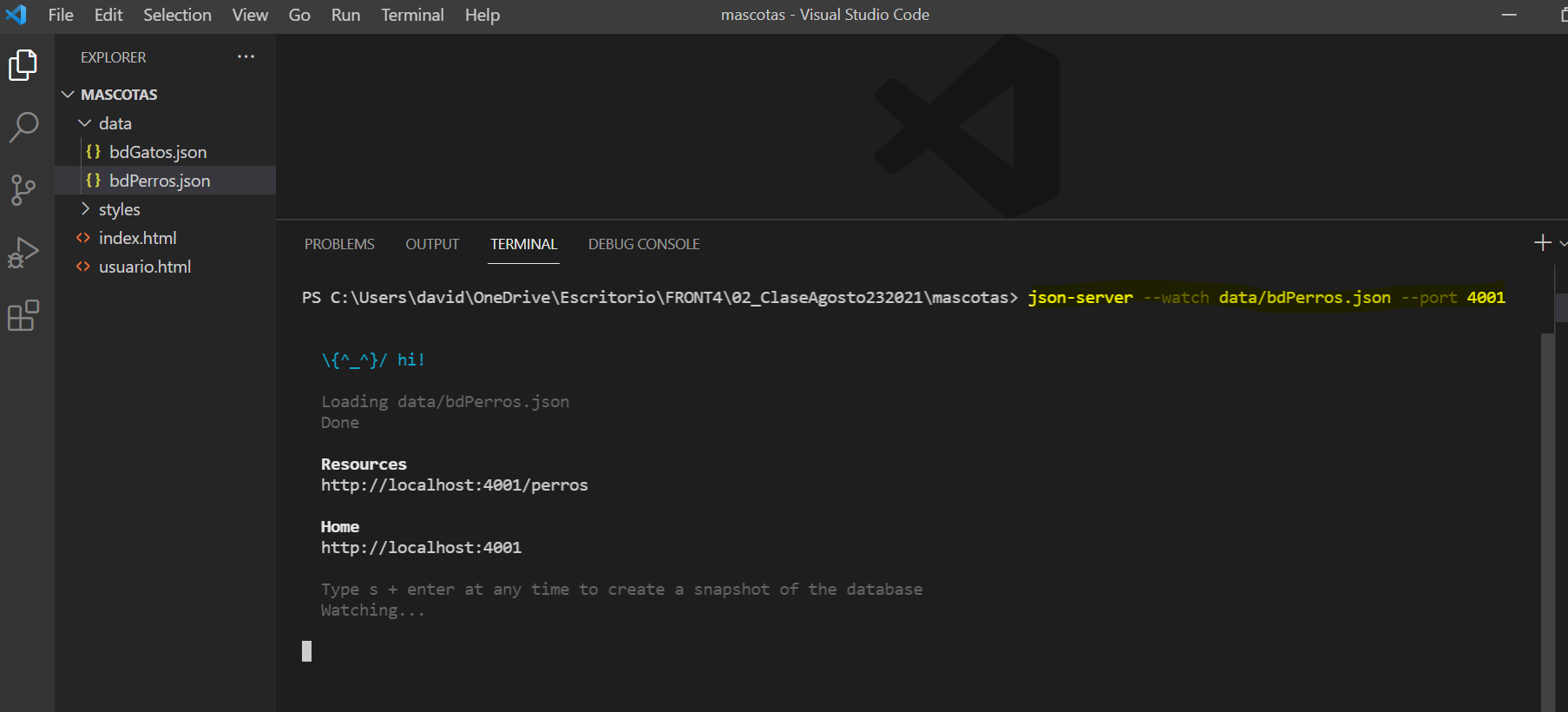


Ahora nuevamente iniciamos relacionando el nombre del archivo que contiene la data y el puerto. Al generar esto nos proporciona una endpoint para consumir

-----------------------------------------------------------------------------------------------------------------------------

Realizamos el mismo procedimiento con el archivo de perros

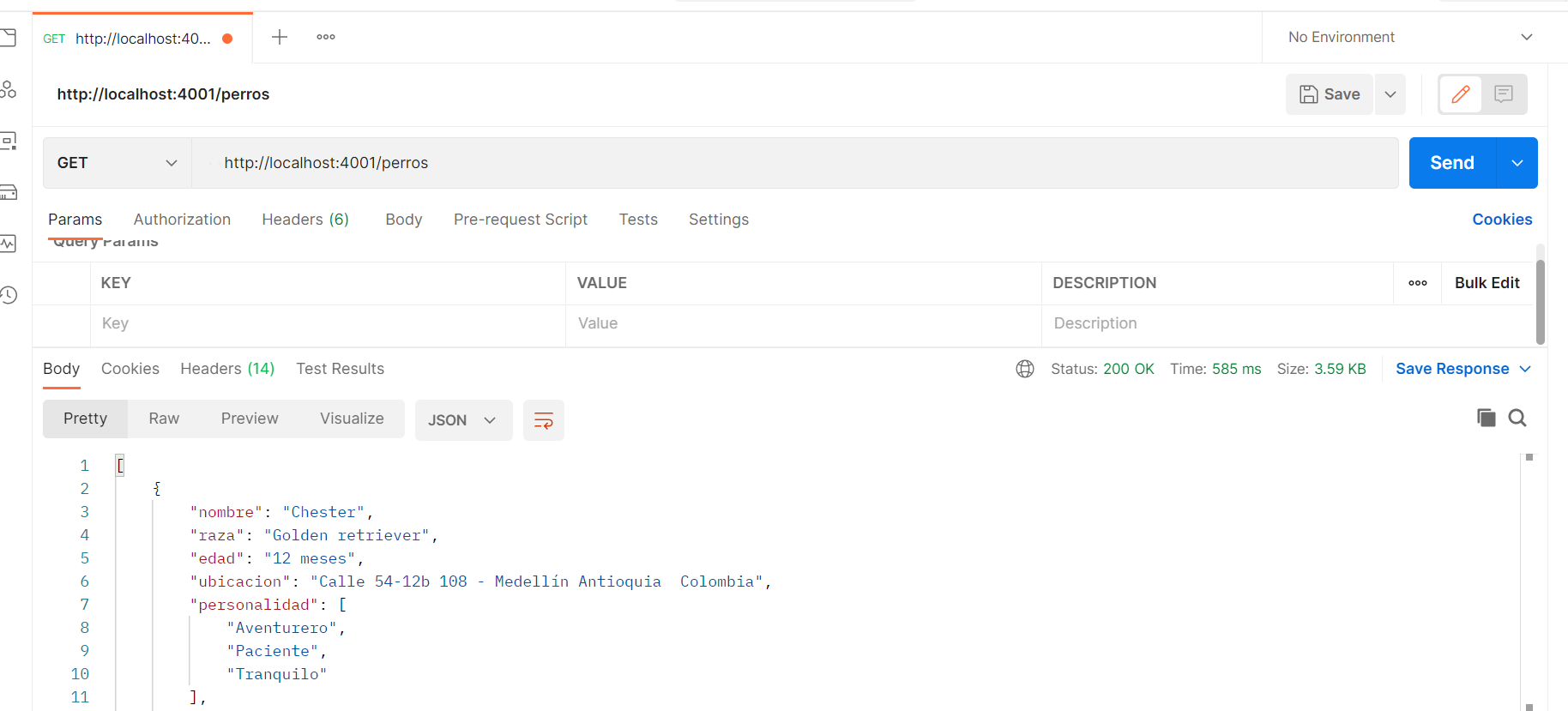
Abrimos una nueva terminal



Esta endpoint la probaremos en postman

Descargar postman

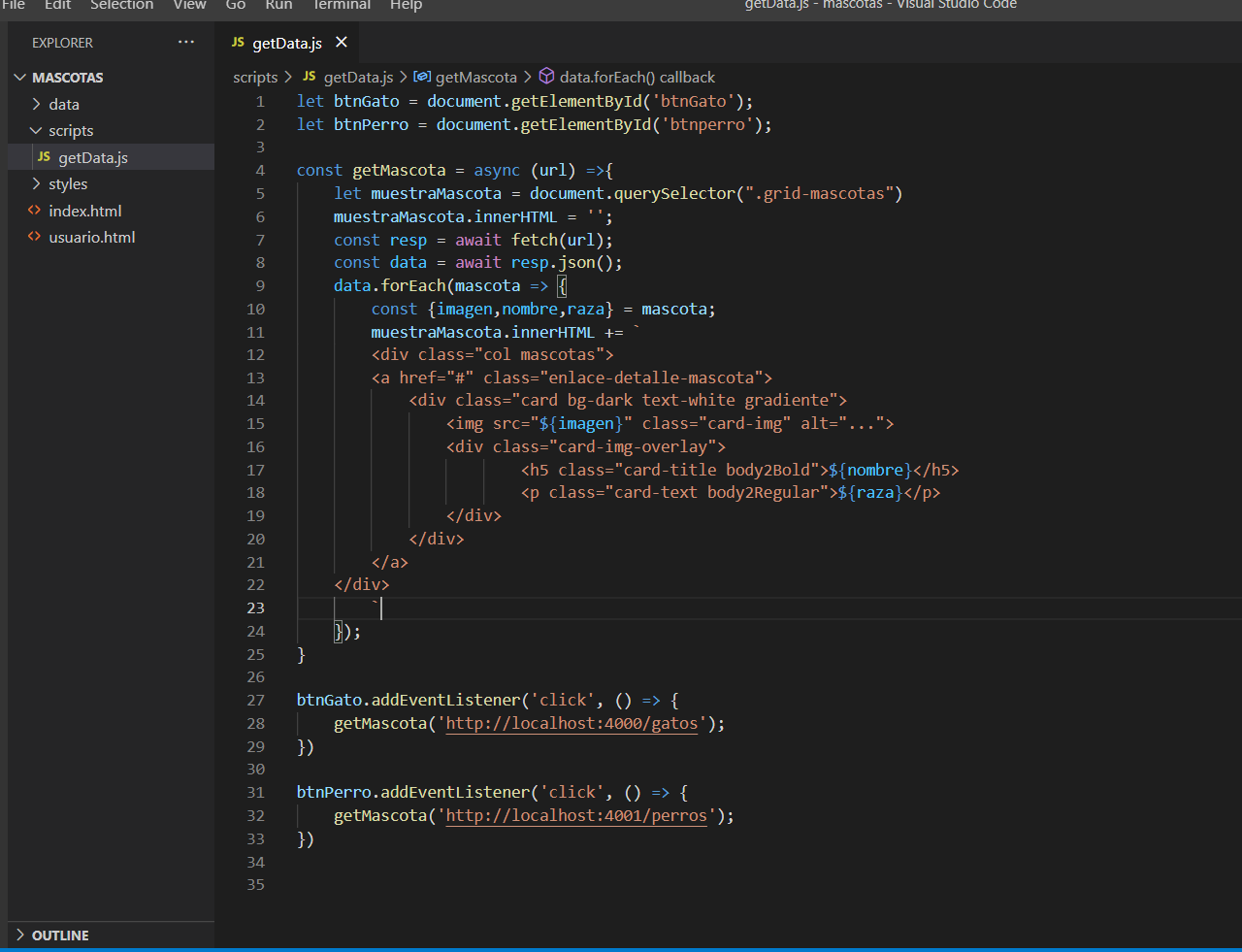
<https://www.postman.com/downloads/>



Nuestra endpoint está respondiendo bien, vamos a consumirla desde nuestra aplicación

Creamos una carpeta que se llame scripts y dentro de ella creamos un archivo llamado getData.js

Si hacemos clic sobre el botón gato debemos mostrar los gatos. Si hacemos clic en perros debemos cargar los perros



    <div class="col mascotas">

        <a href="#" class="enlace-detalle-mascota">

            <div class="card bg-dark text-white gradiente">

                <img src="${imagen}" class="card-img" alt="...">

                <div class="card-img-overlay">

                        <h5 class="card-title body2Bold">${nombre}</h5>

                        <p class="card-text body2Regular">${raza}</p>

                </div>

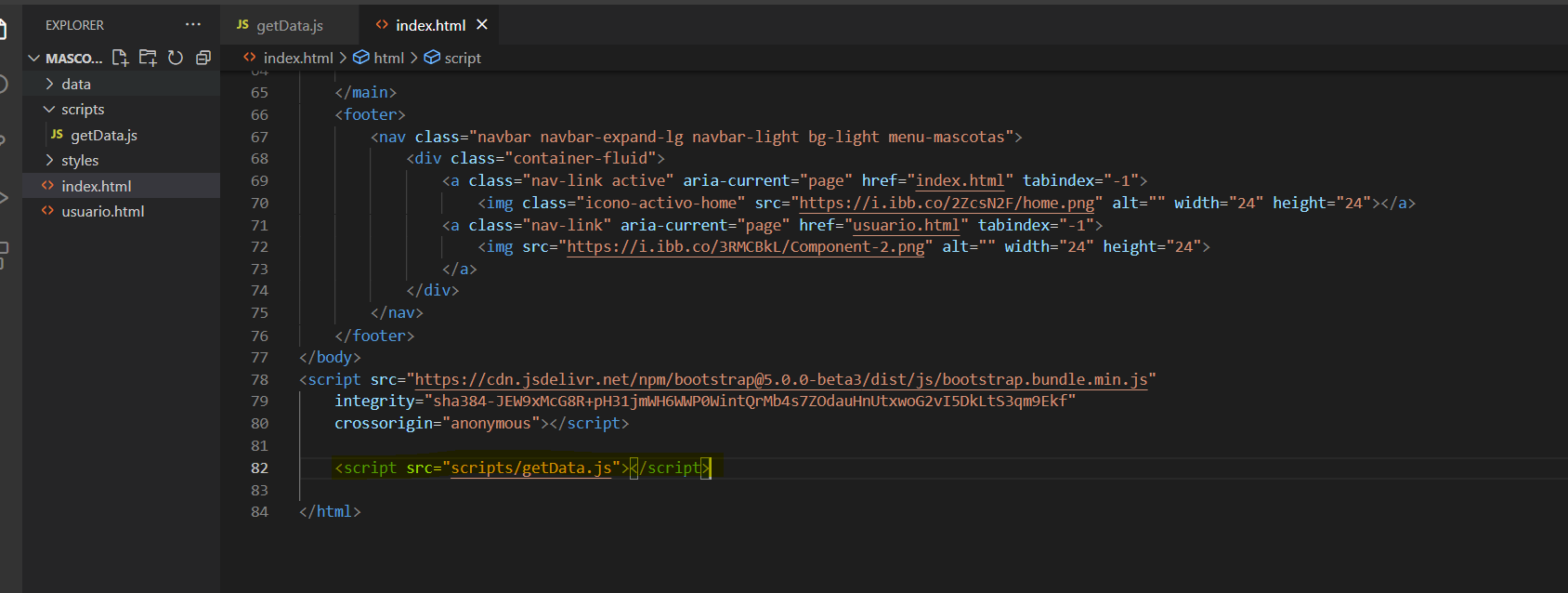
            </div>

        </a>

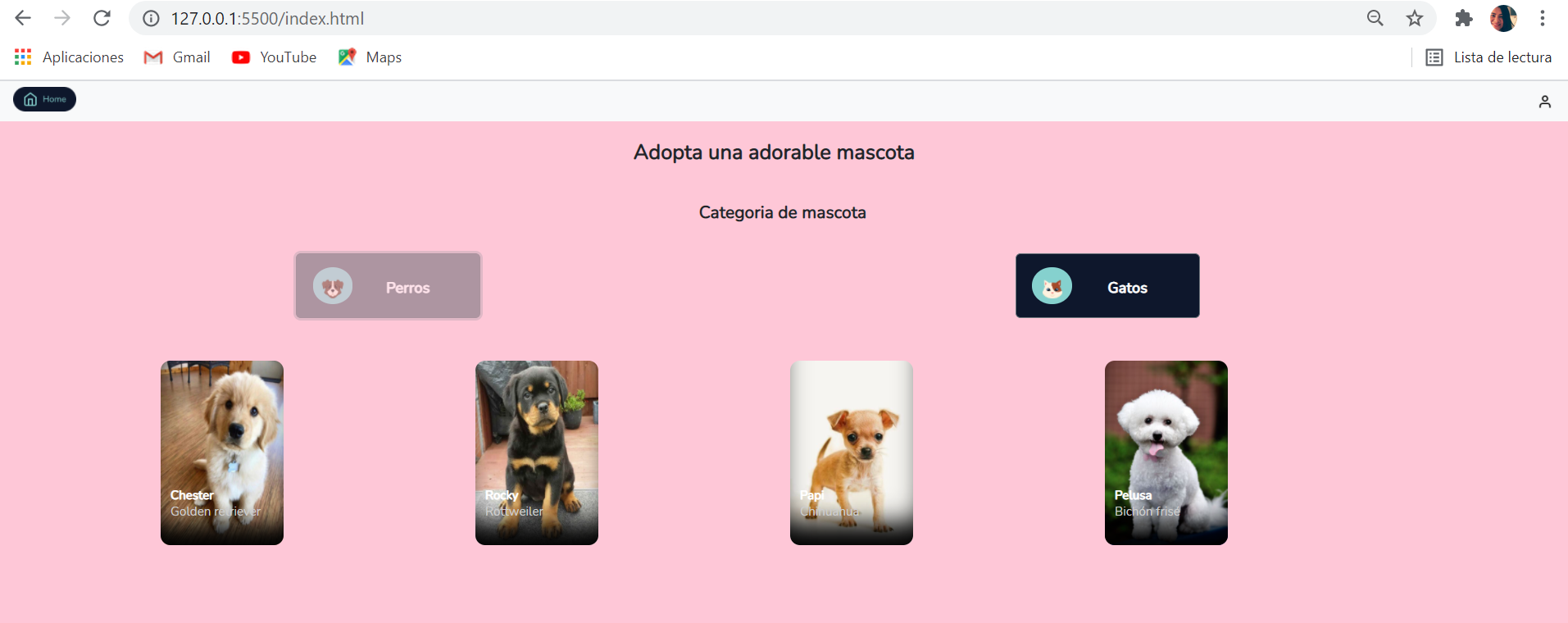
    </div>

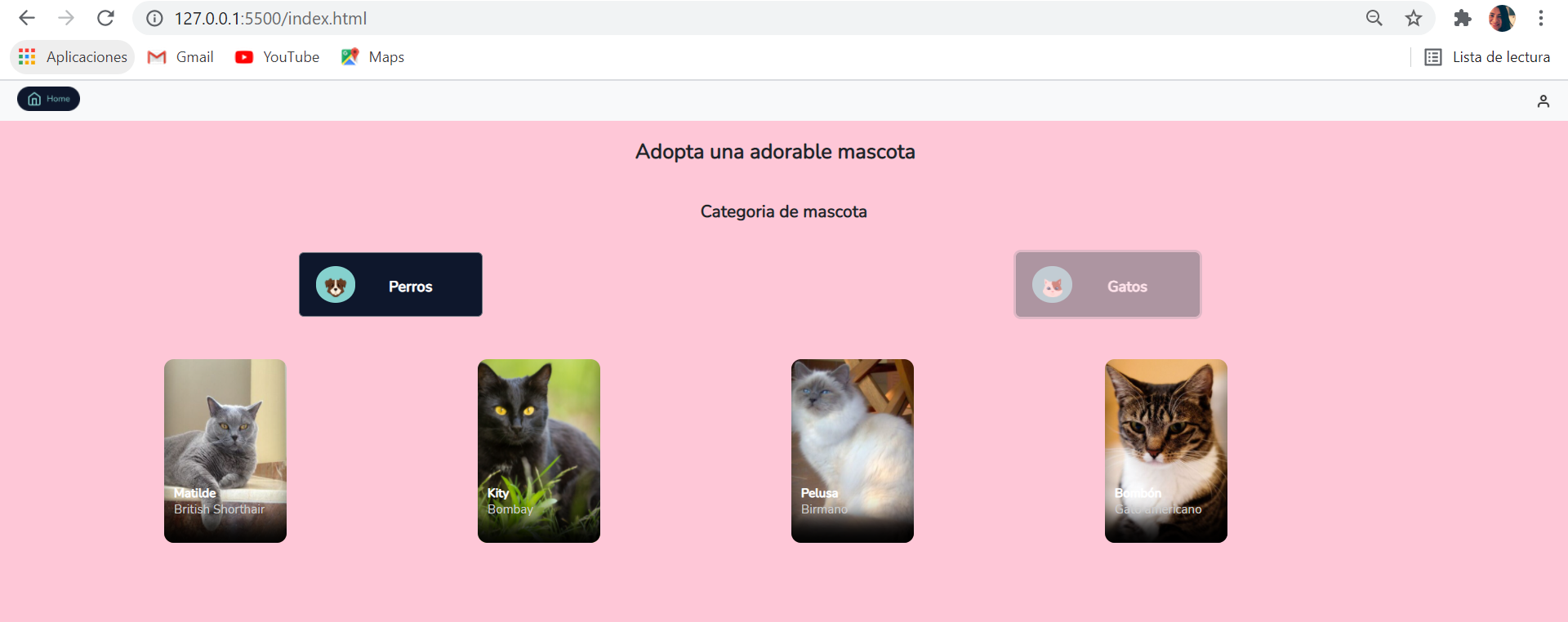
        `

Vinculamos nuestro getData.js con nuestro archivo index.html



Probamos

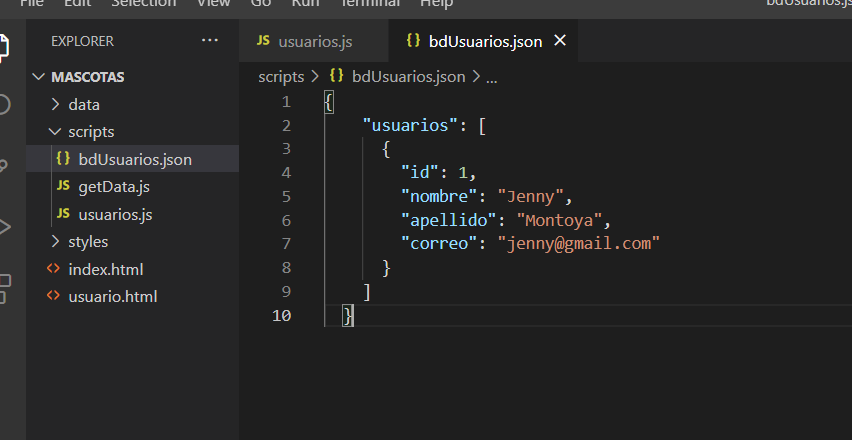




Ahora programaremos el perfil

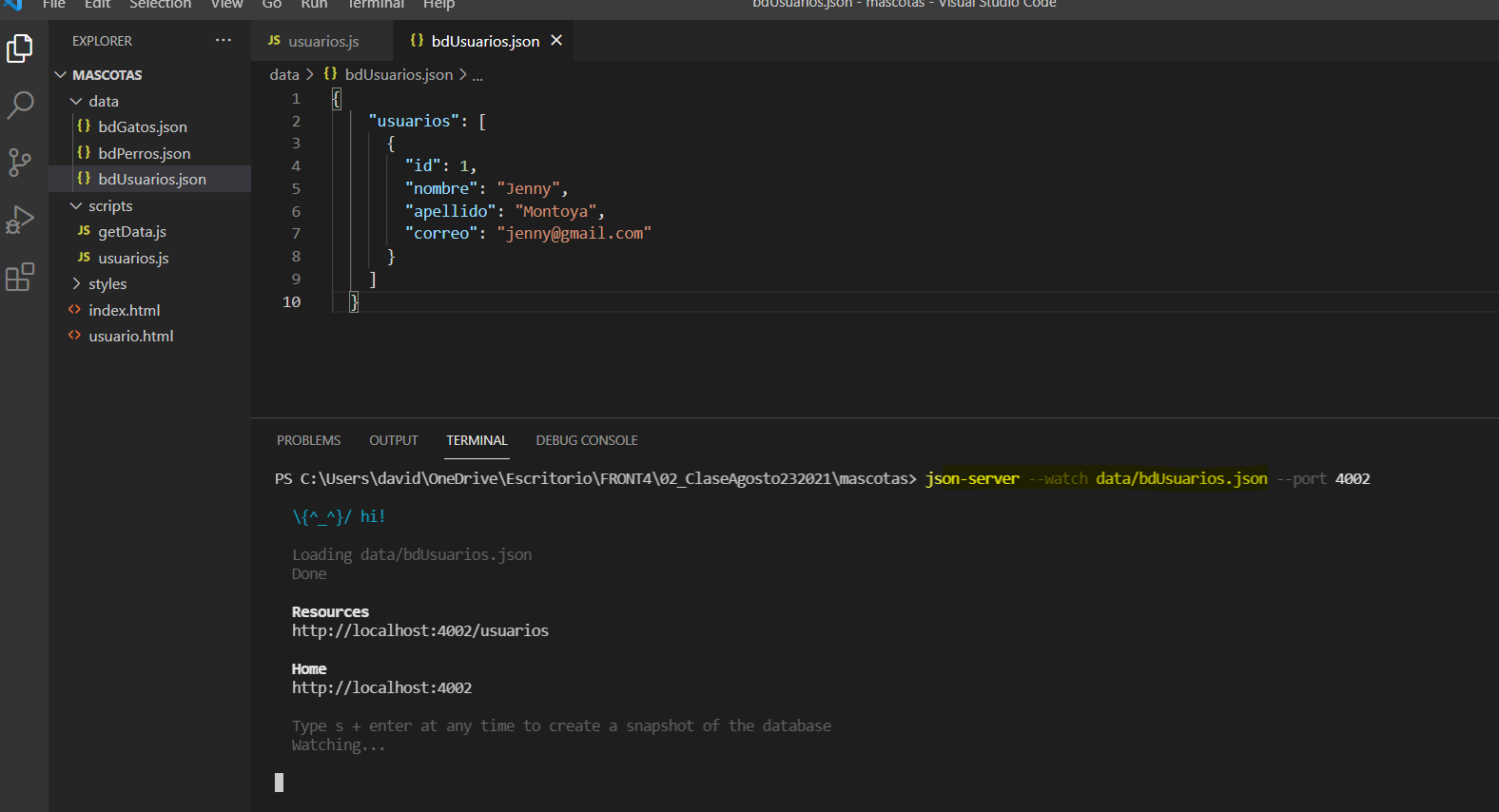


Crear dentro de la carpeta data un archivo llamado bdUsuarios.json



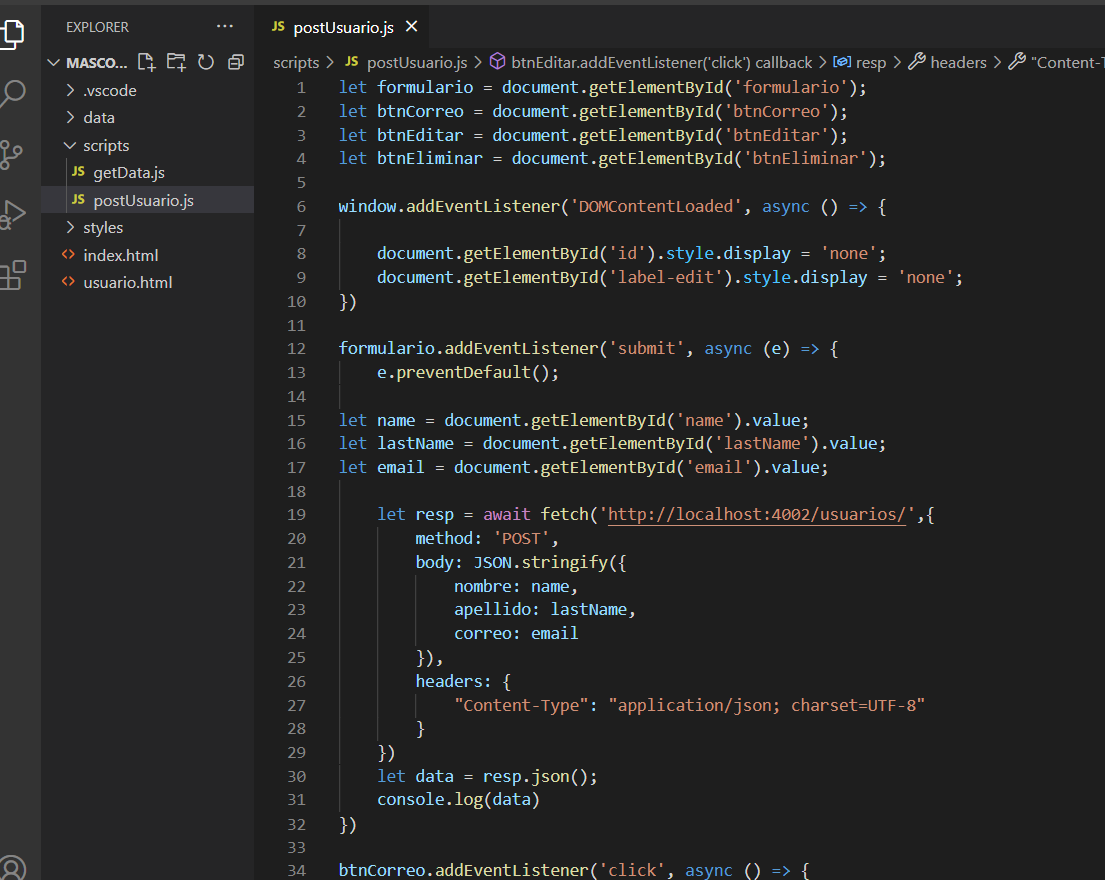
Y levantaremos un json server

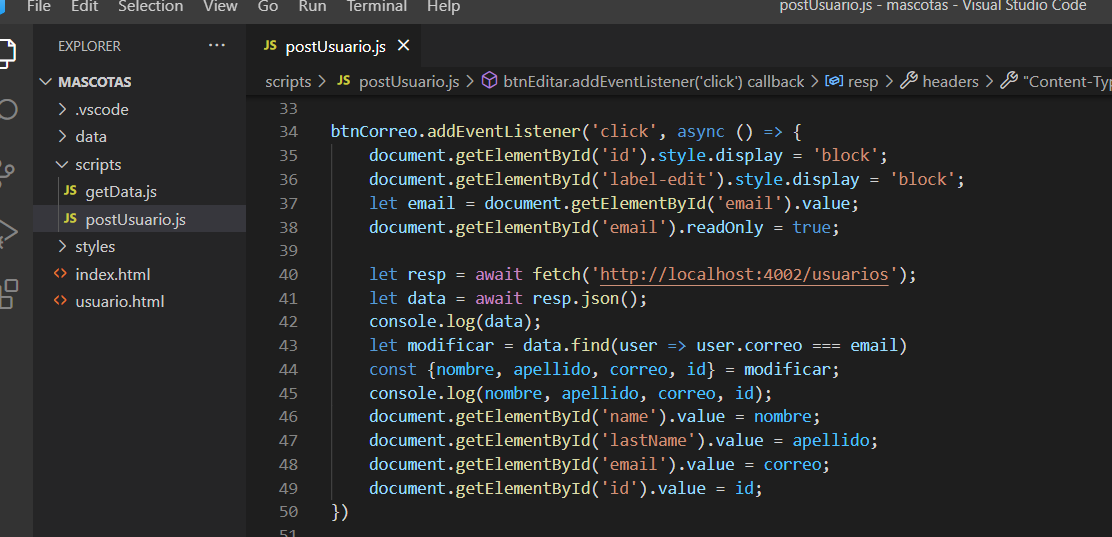
json-server --watch data/bdUsuarios.json --port 4002

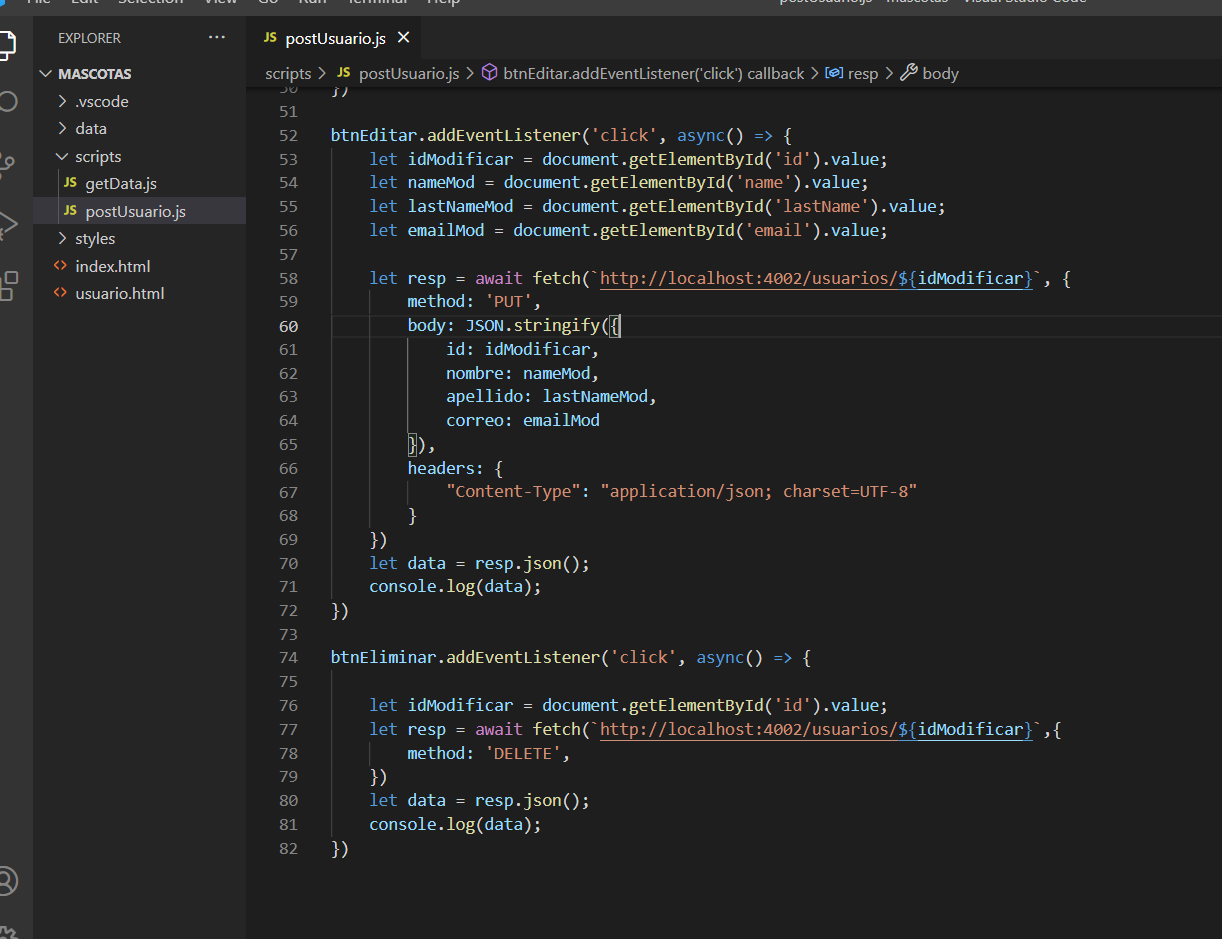


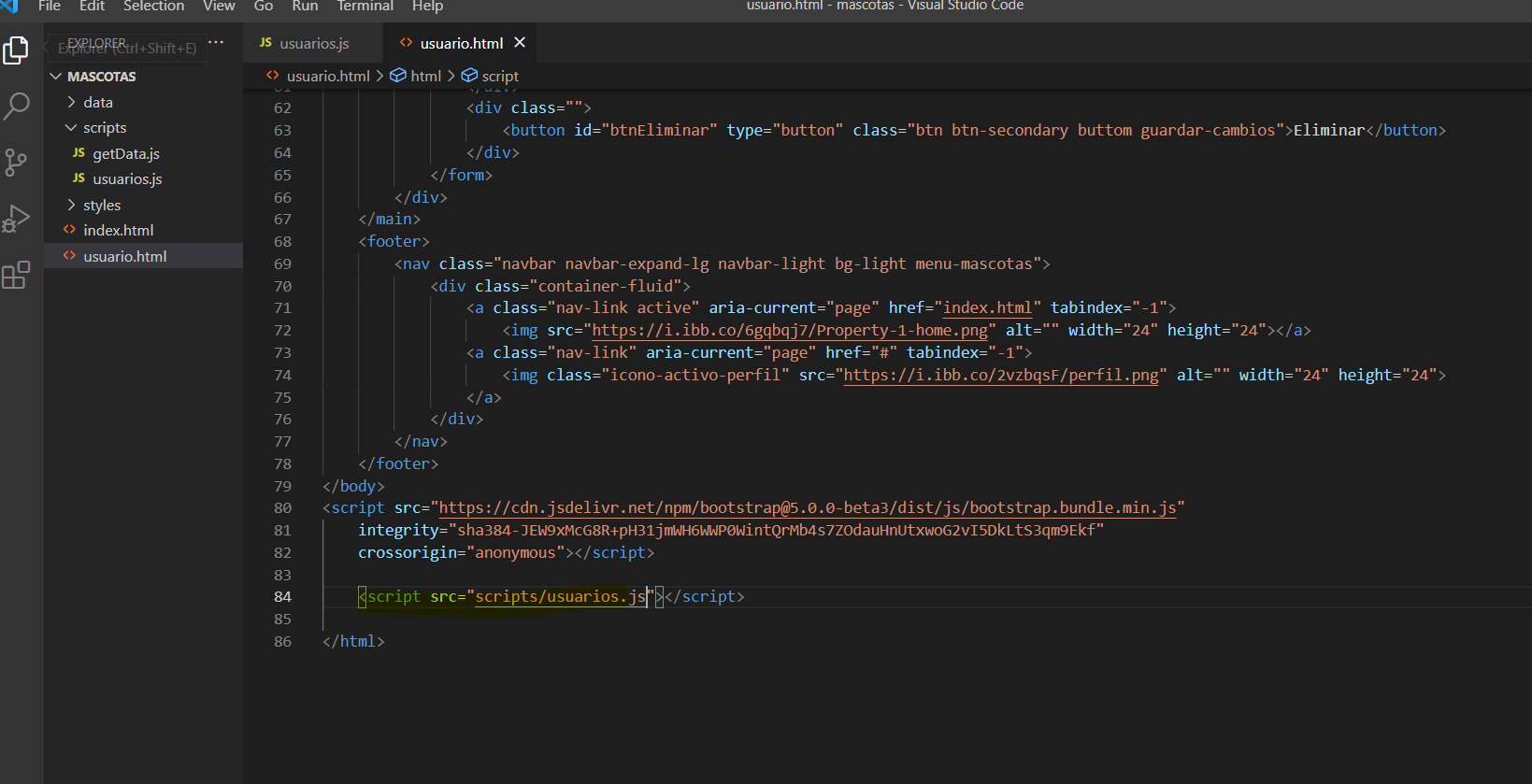
http://localhost:4002/usuarios

Crear dentro de la carpeta scripts un archivo llamado usuarios.js y adicionaremos un objeto de prueba



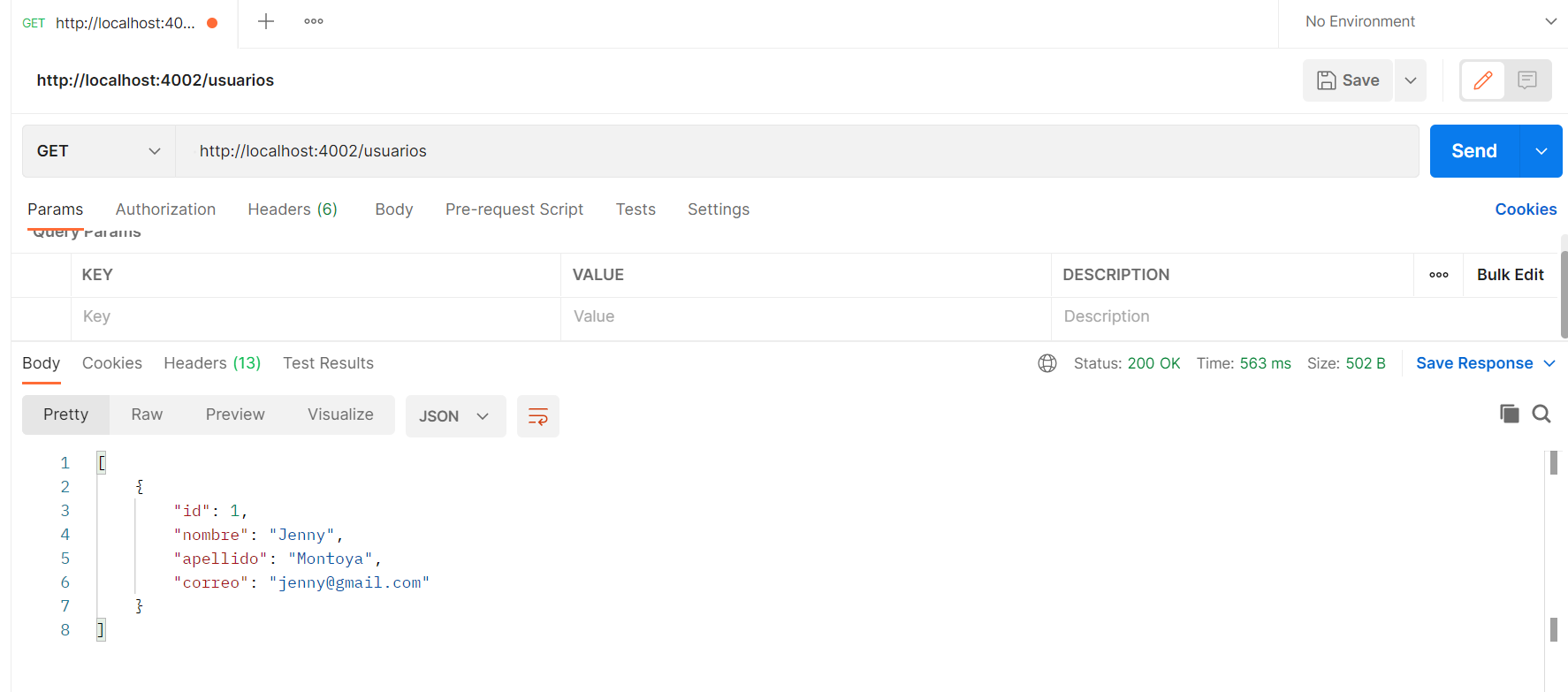




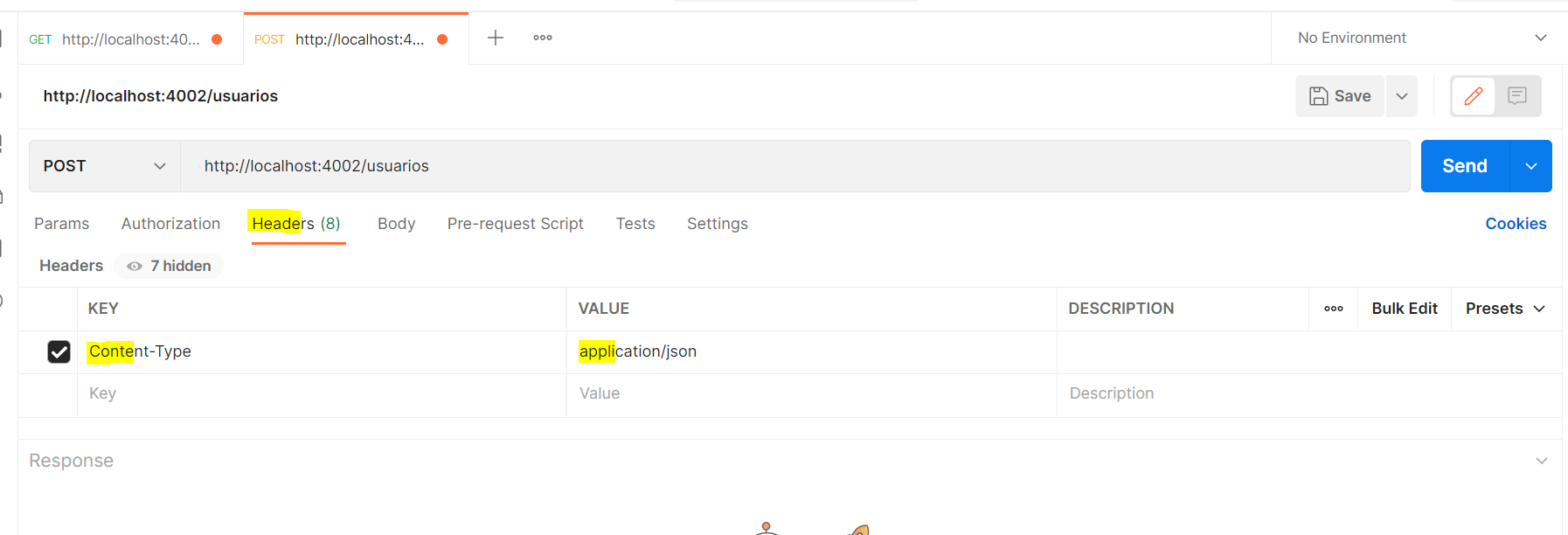


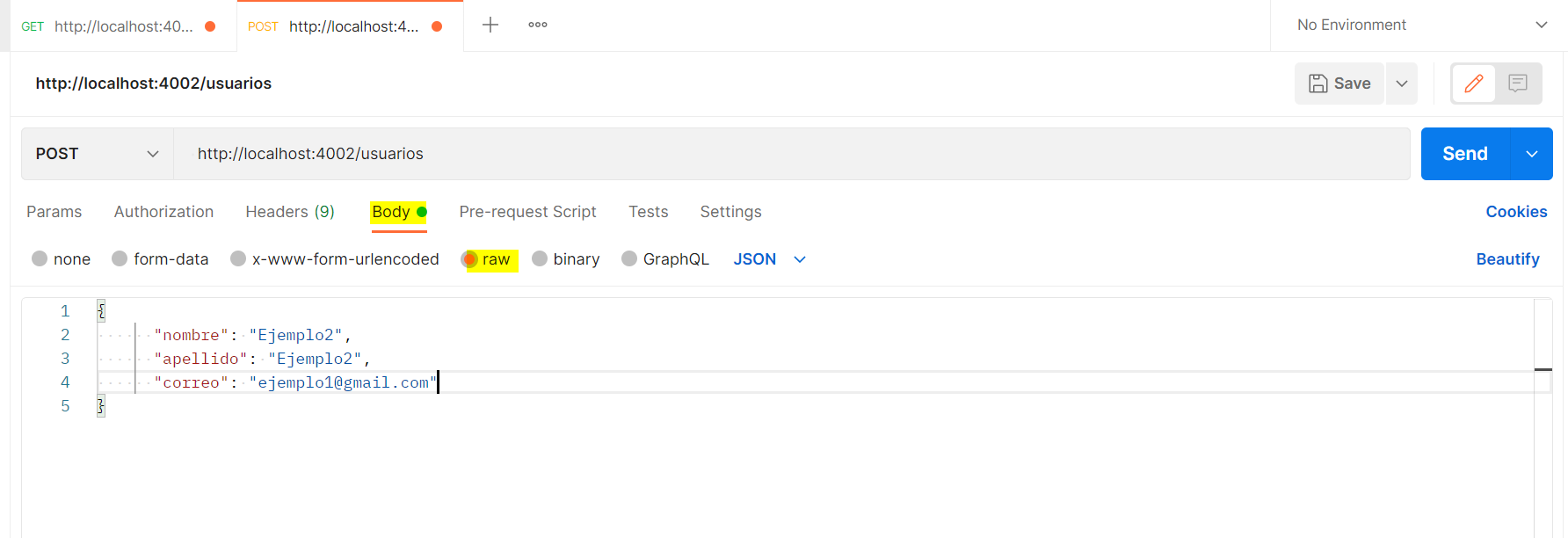
Probemos con postman y con nuestra aplicación

Petición GET de usuarios postman

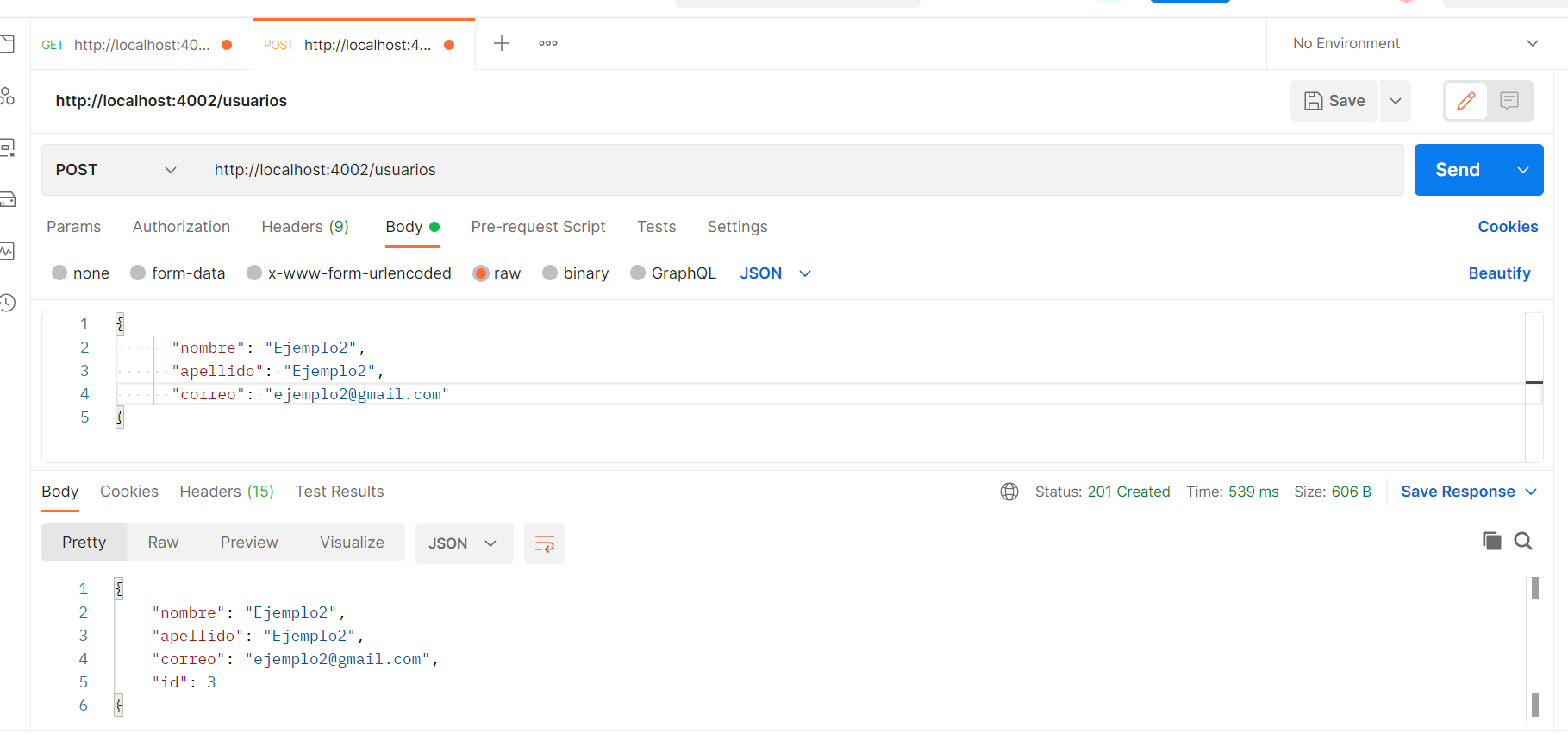


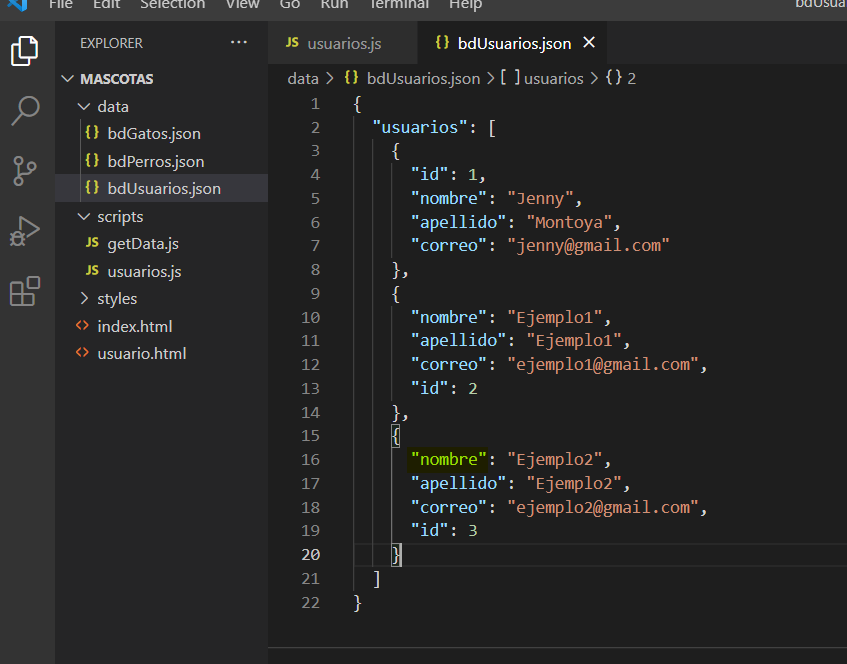
Petición POST --- Creación de usuarios Postman



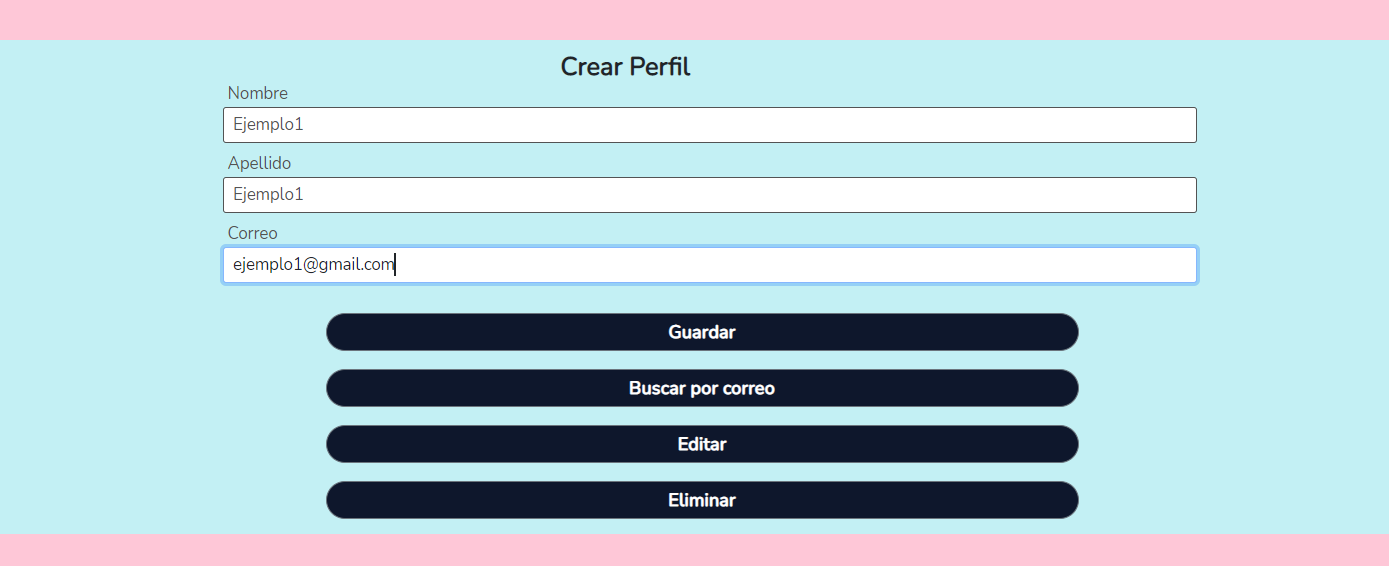


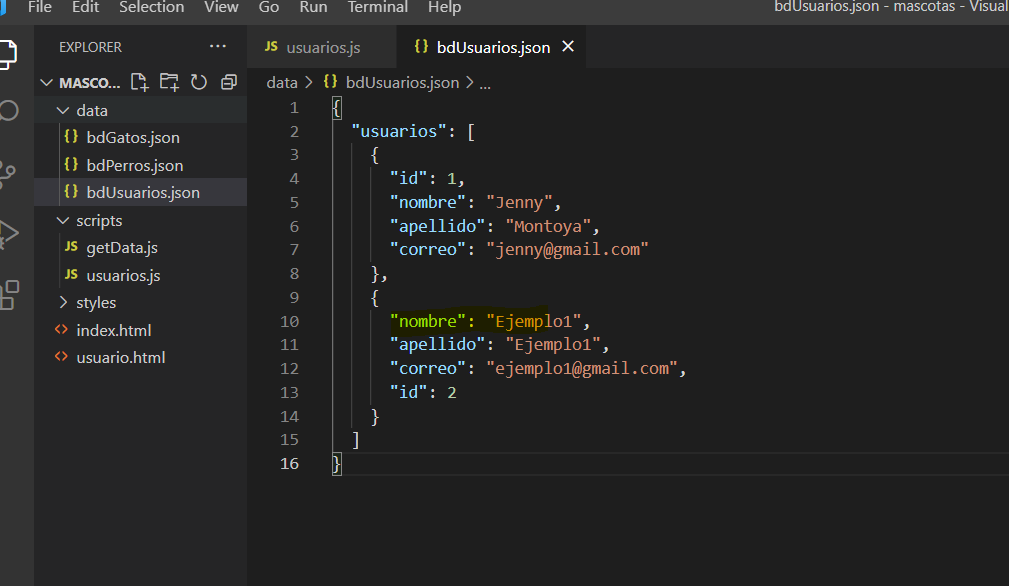
Clic en send



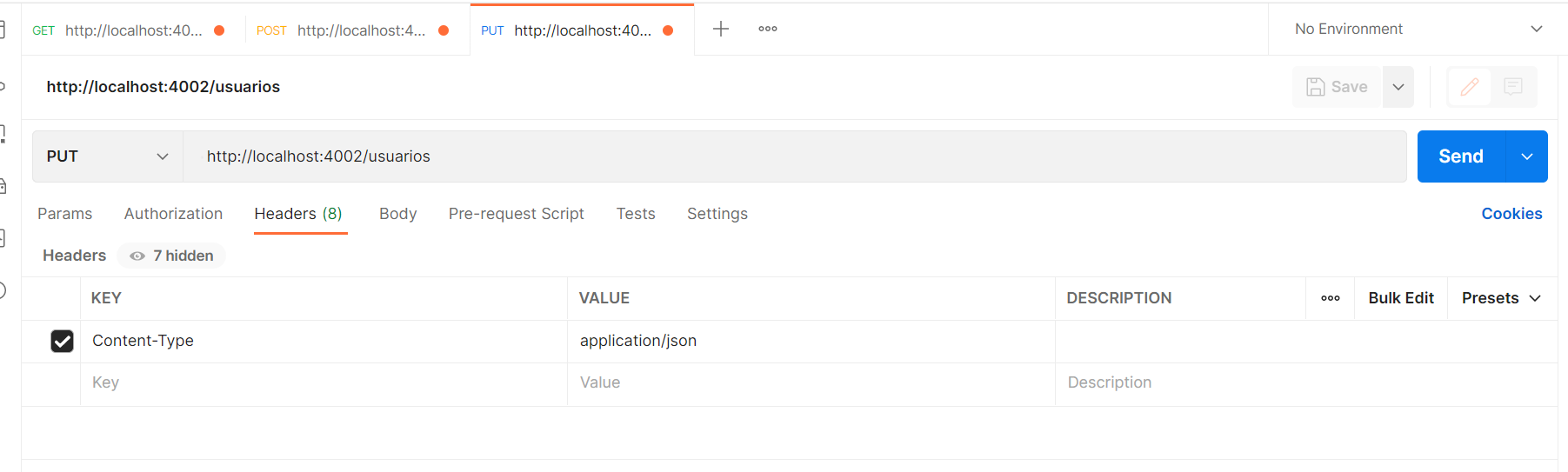


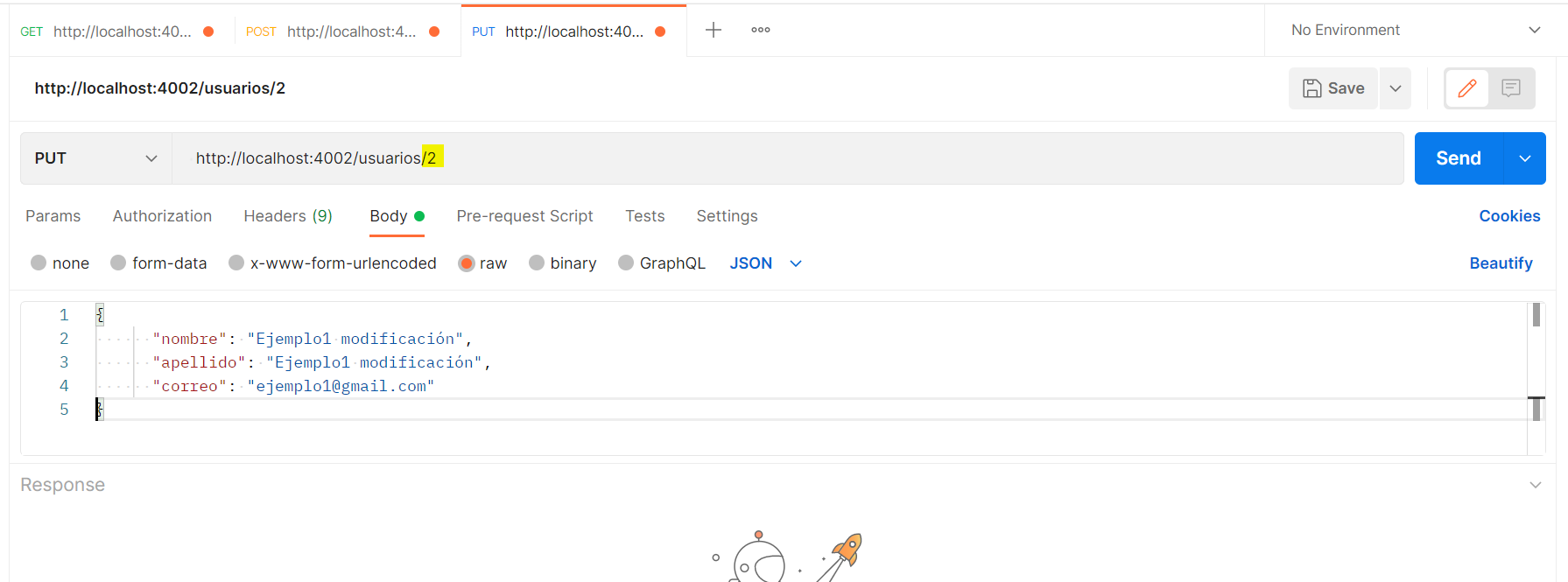
Petición POST con nuestra aplicación



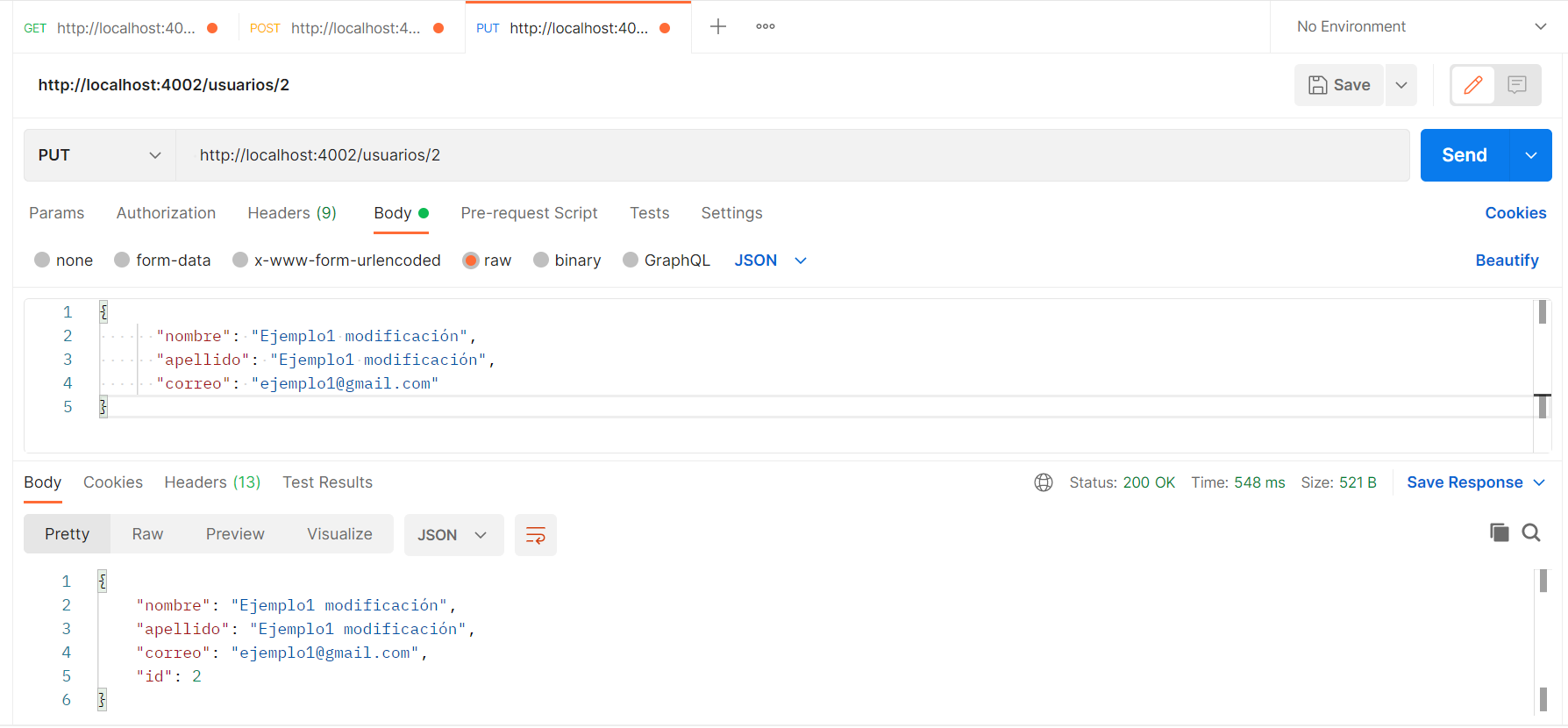


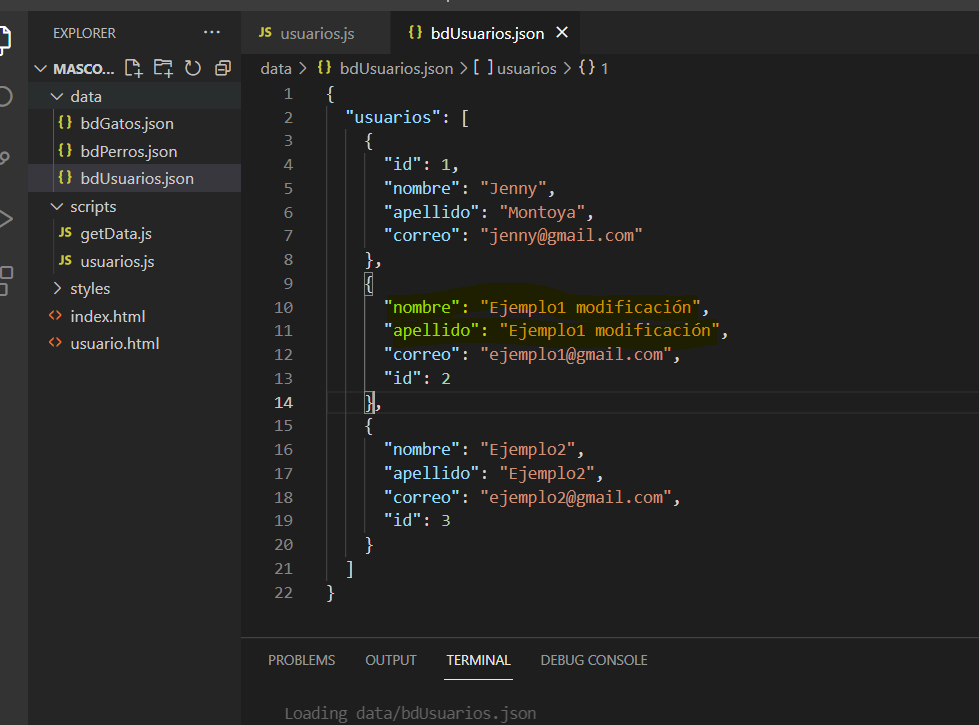
Petición PUT --- Modifición Postman



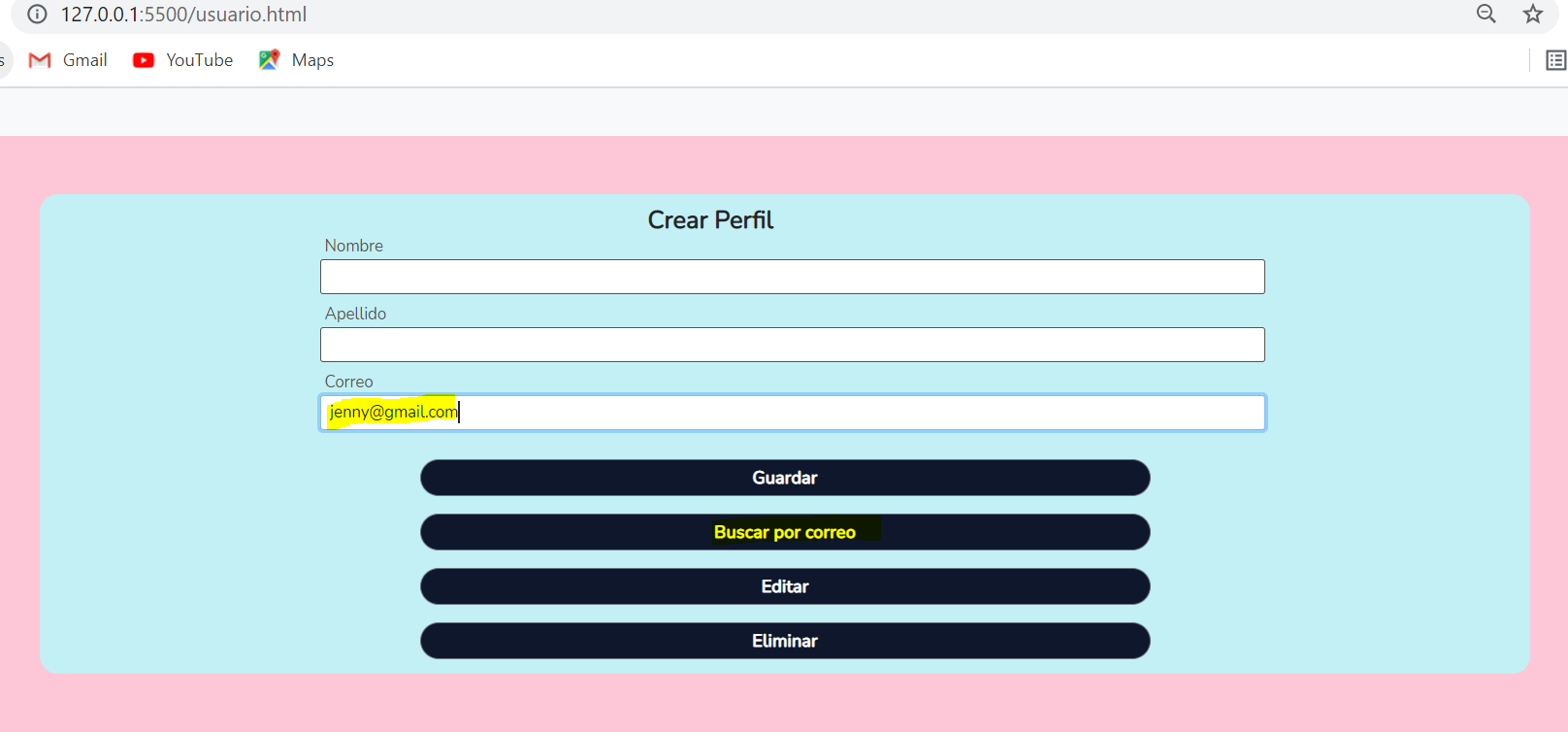


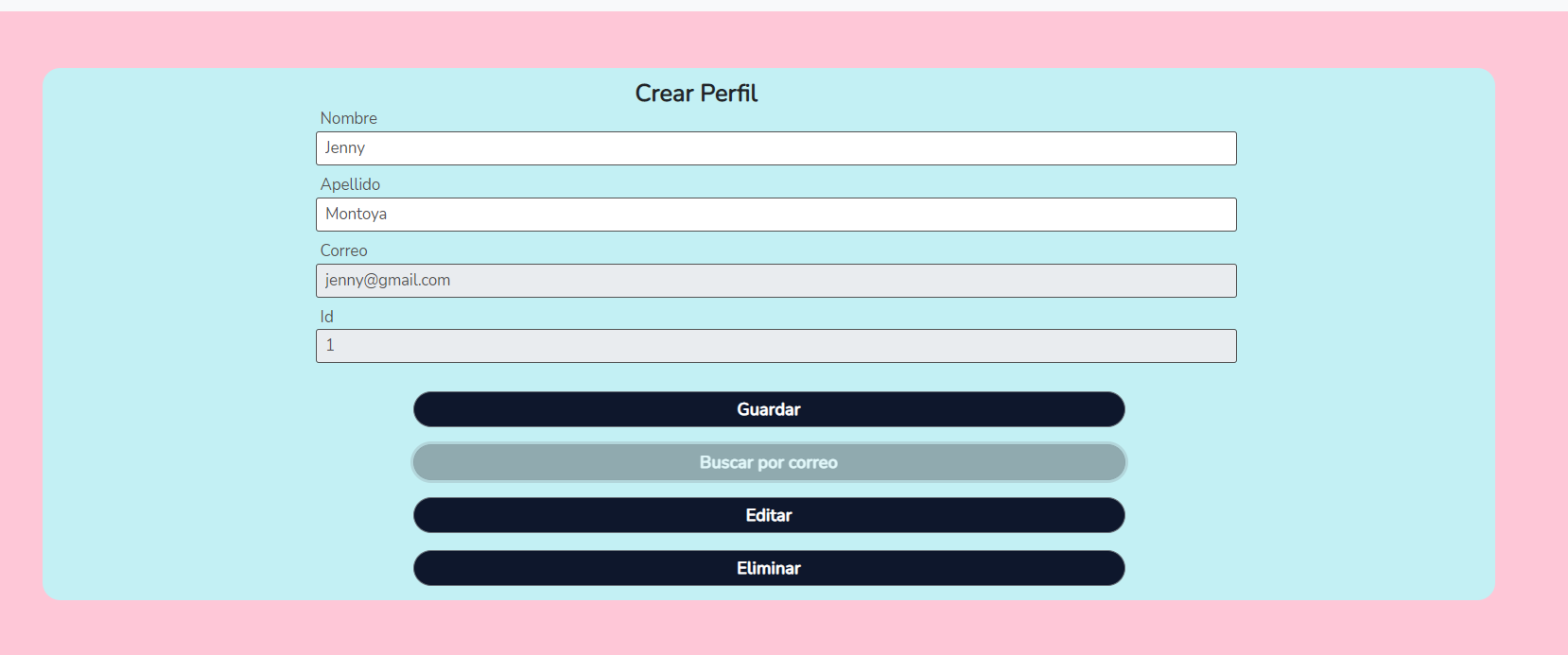
Clic en send





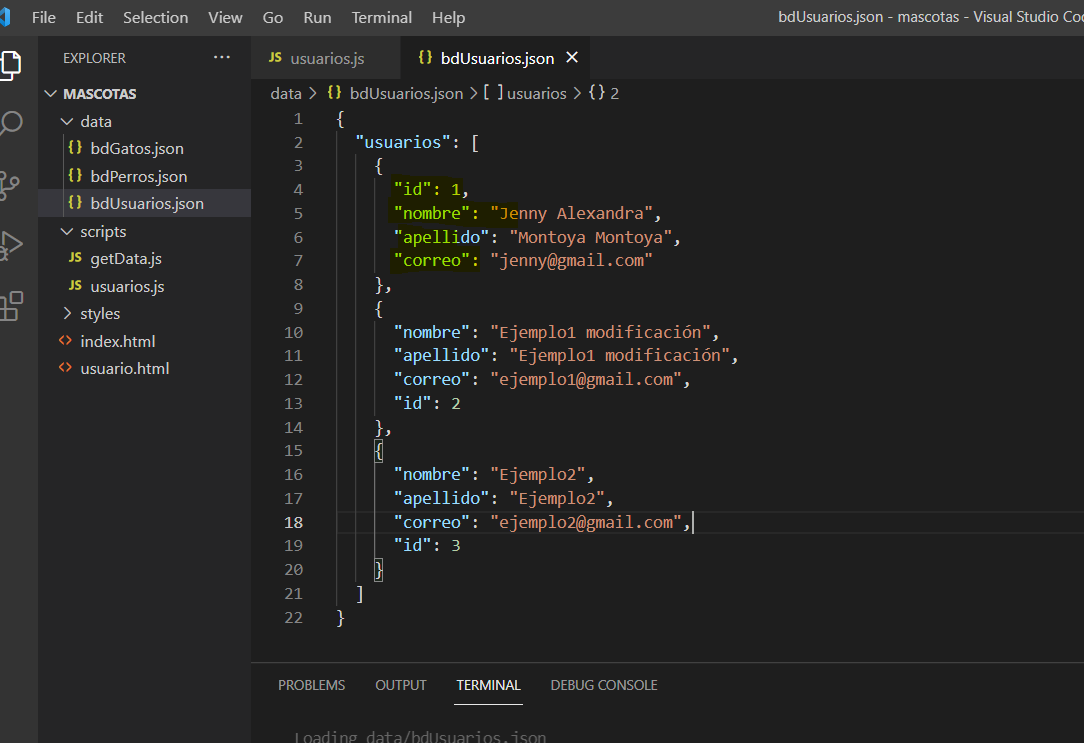
Petición PUT --- Modificación Aplicación





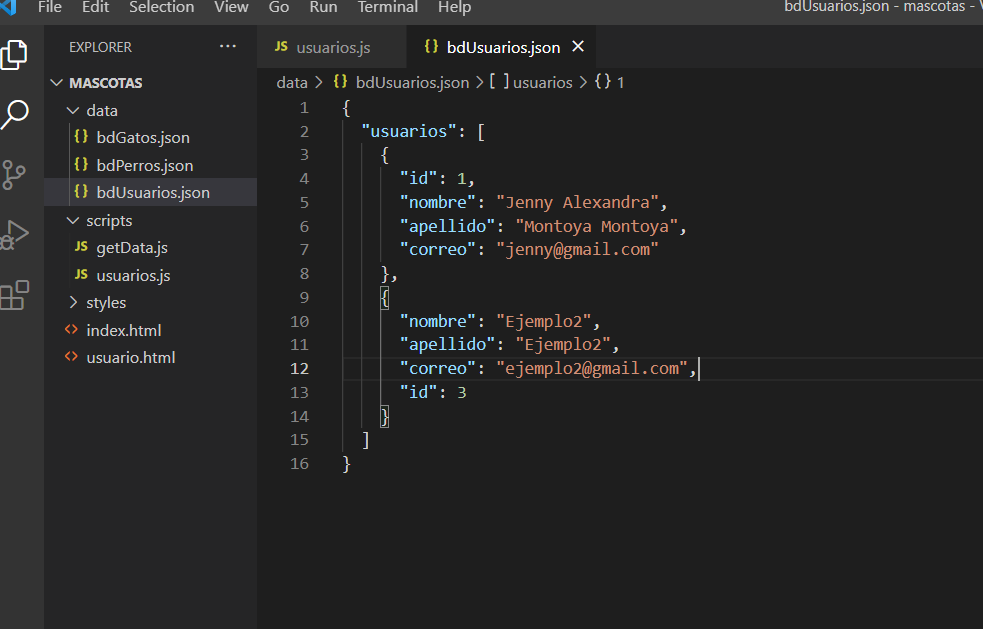
Realizar los cambios y hacer clic en Editar





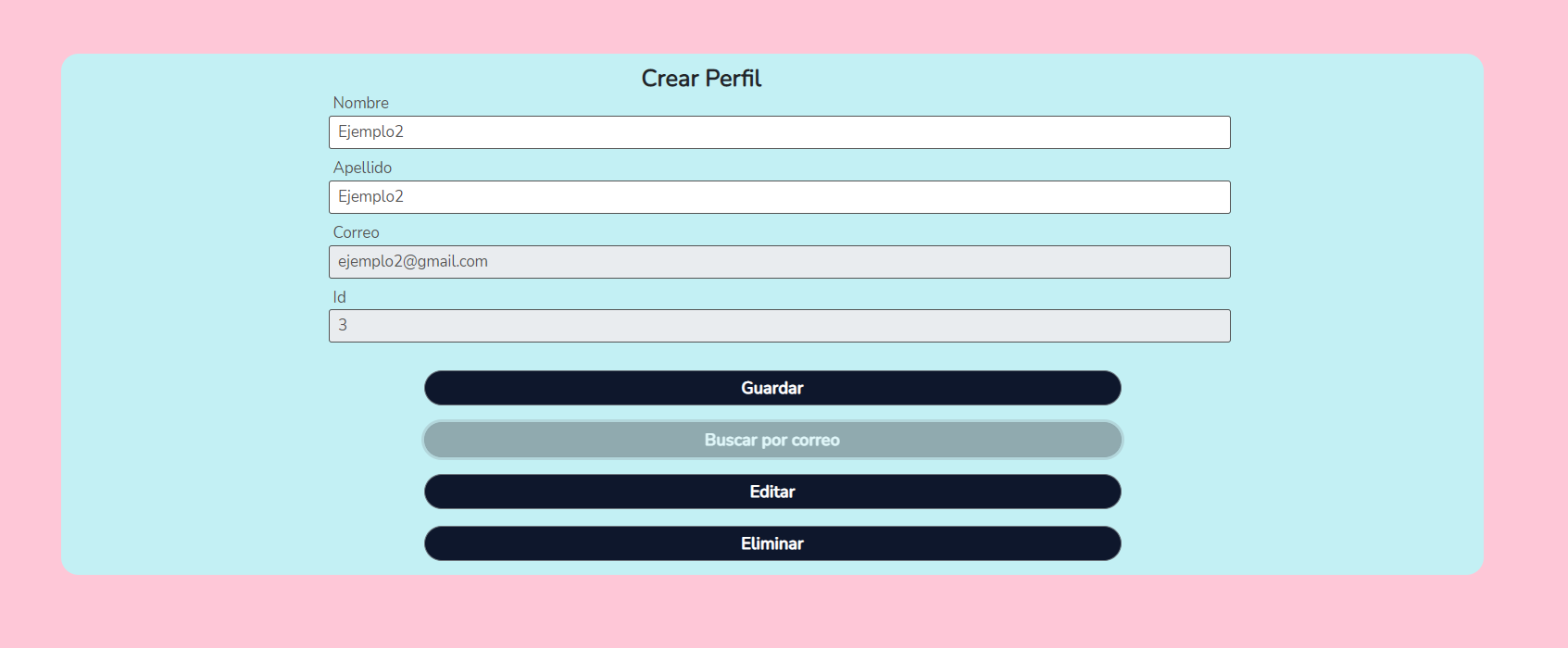
Petición DELETE ---- Eliminar postman

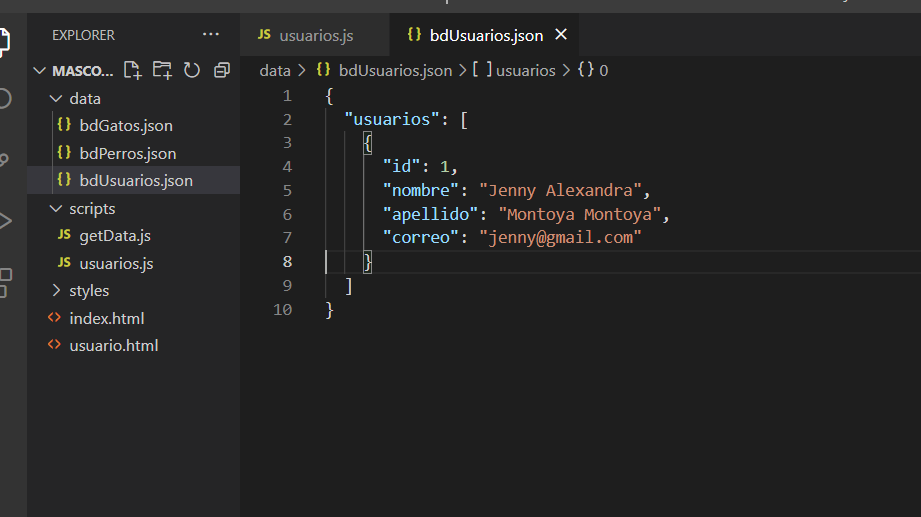




Petición PUT desde la aplicación

Buscar por correo antes de eliminar, luego clic en el botón Eliminar





Repositorio

<https://github.com/jennymontoya1001/mascotasAppVanilla.git>