**APLICACIÓN OH MY TOWN**

**BACKEND USUARIOS**

**Recuperar usuario por ID**

url= “/api/usuario/”+id\_usuario

Llamada fetch: const resp = await fetch(process.env.BACKEND\_URL + *url*)

https://3001-4geeksacade-reactflaskh-kbhpyhxtwr1.ws-eu78.gitpod.io/api/usuario/1

Resultado:

{

"apellidos": "apellidos",

"ciudad": "ciudad",

"descripcion": "descripcion",

"email": "email",

"foto": "src/imgs/usuario.jpg",

"id": 1,

"nombre": "nombre",

"tipo": 0,

"activo": 1

}

Tipo 0 usuario normal, tipo 1 guía

El campo “activo” se usa por si el usuario se quiere dar de baja

**Desactiva usuario por id**

Usar si el usuario se quiere dar de baja

https://3001-4geeksacade-reactflaskh-kbhpyhxtwr1.ws-eu78.gitpod.io/api/desactiva\_user/7

Cambia el campo “activo” a 0

Devuelve “Usuario desactivado con éxito”

**Modifica usuario por id**

Se usa para que el usuario rellene los datos de su perfil o los modifique

https://3001-4geeksacade-reactflaskh-kbhpyhxtwr1.ws-eu78.gitpod.io/api/modifica\_user/7

url =”/api/modifica\_user/”+ usuario\_id

head = {"Content-Type": "application/json"}

body = {

"apellidos": "alegre",

"ciudad": "leon",

"descripcion": "descripcion",

"email": "pablo5@gmail.com",

"nombre": "Pablo",

"tipo": 0,

"password": **false**

}

Estos datos hay que pasarlos obligatoriamente, por lo tanto hay que recuperarlos y variar los que el usuario quiera modificar.

Devuelve "Usuario modificado con éxito"

Llamada fetch:

await fetch(process.env.BACKEND\_URL + url, {method: “POST”, headers: head, body: body,})

**Cambia o sube foto usuario por id**

Hay que hacer un componente específico para esta ruta

https://3001-4geeksacade-reactflaskh-kbhpyhxtwr1.ws-eu78.gitpod.io/api/foto\_user/7

Ruta “/api/foto\_user/”+id\_usuario

La imagen viene de un input de tipo file que se llama “archivo” por el método POST

Sube a “src/imgs/users/” la imagen

Devuelve "Foto cambiada con exito"

**Nuevo usuario**

<https://3001-4geeksacade-reactflaskh-kbhpyhxtwr1.ws-eu78.gitpod.io/api/new_user>

url =”/api/new\_user”

head = {"Content-Type": "application/json"}

body = { "email": "loquesea@gmail.com", "password": "123456"}

Llamada fetch:

await fetch(process.env.BACKEND\_URL + url, {method: “POST”, headers: head, body: body,})

Si el email ya existe devuelve 1, redirigir a password olvidado

El usuario es tipo 0 por defecto

El usuario es activo 1 por defecto

**Login usuario**

https://3001-4geeksacade-reactflaskh-kbhpyhxtwr1.ws-eu78.gitpod.io/api/login

url =”/api/login”

head = {"Content-Type": "application/json"}

body = { "email": "loquesea@gmail.com", "password": "123456"}

await fetch(process.env.BACKEND\_URL + url, {method: “POST”, headers: head, body: body,})

Si no hay datos en el json devuelve:

{“error”: “no\_data”}

Si no existe el email devuelve:

{“error”: “no\_user”}

Si el password no es correcto devuelve:

{“error”: “no\_pass”}

Si está todo correcto devuelve:

{“token”: el\_token\_creado}

**Recuperar contraseña**

**Crea una nueva contraseña y la envía por email**

https://3001-4geeksacade-reactflaskh-kbhpyhxtwr1.ws-eu79.gitpod.io/api/new\_pass

url =”/api/new\_pass”

head = {"Content-Type": "application/json"}

body = { "email": "loquesea@gmail.com"}

await fetch(process.env.BACKEND\_URL + url, {method: “POST”, headers: head, body: body,})

Si no existe el email en la BBDD devuelve "El email no es válido"

Si por cualquier razón no puede enviar el email devuelve "Password no enviado"

Si funcionó todo devuelve "Enviado password", advertir que mire la carpeta de Spam y redirigir a la página de Login