

Alumno: William Manuel Postillos Aquino

Evidencias:

Postman:

GET <http://localhost:8080/api/v1/usuarios>

The screenshot displays the Postman interface. On the left, the 'History' panel shows a list of requests. The main area shows a GET request to `http://localhost:8080/api/v1/usuarios`. The 'Body' tab is selected, showing a JSON response with three user objects. The status bar at the bottom indicates a 200 OK response with a time of 687 ms and a size of 331 B.

```
@GetMapping
public List<Usuario> getAllUsuarios(){
    return usuarioRepository.findAll();
}
```

GET `http://localhost:8080/api/v1/usuarios`

Params Authorization Headers (6) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary JSON

1

Body Cookies Headers (5) Test Results Status: 200 OK Time: 687 ms Size: 331 B Save Response

Pretty Raw Preview Visualize JSON

```
{
  "nombre": "william",
  "email": "william@gmail.com"
},
{
  "id": 2,
  "nombre": "William",
  "email": "william@gmail.com"
},
{
  "id": 3,
  "nombre": "Manuel",
  "email": "manuel@gmail.com"
}
```

POST <http://localhost:8080/api/v1/usuarios>

The screenshot shows a code editor with a `@PostMapping` method. The method is named `createUsuario` and takes a `Usuario` object as a parameter. It returns the result of `usuarioRepository.save(usuario)`.

```
}
@PostMapping
public Usuario createUsuario(@RequestBody Usuario usuario){
    return usuarioRepository.save(usuario);
}
```

Postman interface showing a POST request to `http://localhost:8080/api/v1/usuarios`. The request body is a JSON object:

```
{  "nombre": "Ronald",  "email": "ronald@gmail.com"}
```

The response status is 200 OK, Time: 4.98 s, Size: 217 B. The response body is a JSON object:

```
{  "id": 6,  "nombre": "Ronald",  "email": "ronald@gmail.com"}
```

GET <http://localhost:8080/api/v1/usuarios/1>

```
@GetMapping("/{id}")
public ResponseEntity<Usuario> getUsuarioById(@PathVariable long id){
    Usuario usuario = usuarioRepository.findById(id)
        .orElseThrow(() -> new ResourceNotFoundException("Usuario no existente con id:" + id));
    return ResponseEntity.ok(usuario);
}
```

Postman interface showing a GET request to `http://localhost:8080/api/v1/usuarios/1`. The response status is 200 OK, Time: 6.83 s, Size: 219 B. The response body is a JSON object:

```
{  "id": 1,  "nombre": "william",  "email": "william@gmail.com"}
```

PUT <http://localhost:8080/api/v1/usuarios/1>

```

37
38 @PutMapping("/{id}")
39 public ResponseEntity<Usuario> updateUsuario(@PathVariable long id, @RequestBody Usuario usuarioDetails){
40     Usuario updateUsuario = usuarioRepository.findById(id).orElseThrow(() -> new ResourceNotFoundException("Usuario no existente con id: " + id));
41
42     updateUsuario.setNombre(usuarioDetails.getNombre());
43     updateUsuario.setEmail(usuarioDetails.getEmail());
44
45     usuarioRepository.save(updateUsuario);
46
47     return ResponseEntity.ok(updateUsuario);
48 }
49

```

GET http://localhost:8080/api/ POST http://localhost:8080/api/ GET http://localhost:8080/api/ PUT http://localhost:8080/api/ + ...

HTTP http://localhost:8080/api/v1/usuarios/1 Save </>

PUT http://localhost:8080/api/v1/usuarios/1 Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary JSON

```

1 {
2   "nombre": "Pedro",
3   "email": "pedro@gmail.com"
4 }

```

Body Cookies Headers (5) Test Results Status: 200 OK Time: 2.47 s Size: 215 B Save Response

Pretty Raw Preview Visualize JSON

```

1 {
2   "id": 1,
3   "nombre": "Pedro",
4   "email": "pedro@gmail.com"
5 }

```

GET http://localhost:8080/api/ POST http://localhost:8080/api/ GET http://localhost:8080/api/ PUT http://localhost:8080/api/ X + ...

HTTP http://localhost:8080/api/v1/usuarios/1 Save </>

GET http://localhost:8080/api/v1/usuarios/1 Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Bulk Edit
Key	Value	

Body Cookies Headers (5) Test Results Status: 200 OK Time: 672 ms Size: 215 B Save Response



Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 1,
3   "nombre": "Pedro",
4   "email": "pedro@gmail.com"
5 }
```

DELETE <http://localhost:8080/api/v1/usuarios/2>

```
50 @DeleteMapping("/{id}")
51 public ResponseEntity<HttpStatus> deleteUsuario(@PathVariable long id){
52     Usuario usuario = usuarioRepository.findById(id).orElseThrow(()-> new ResourceNotFoundException("Usuario no existente con el id:" + id));
53
54     usuarioRepository.delete(usuario);
55
56     return new ResponseEntity<>(HttpStatus.NO_CONTENT);
57 }
58
59
```

GET http://localhost:8080/api/ POST http://localhost:8080/ap/ GET http://localhost:8080/api/ PUT http://localhost:8080/api/ DEL http://localhost:8080/api/ + ...


 http://localhost:8080/api/v1/usuarios/2  Save </>



DELETE Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params



	Key	Value	Bulk Edit
	Key	Value	

Body Cookies Headers (3) Test Results  Status: 204 No Content Time: 425 ms Size: 112 B Save Response

Pretty Raw Preview Visualize Text  

1

GET http://localhost:8080/api/ POST http://localhost:8080/ap/ GET http://localhost:8080/api/ PUT http://localhost:8080/api/ DEL http://localhost:8080/api/ + ...


 http://localhost:8080/api/v1/usuarios/2  Save </>



GET Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

	Key	Value	Bulk Edit
	Key	Value	

Body Cookies Headers (5) Test Results  Status: 404 Not Found Time: 108 ms Size: 277 B Save Response

Pretty Raw Preview Visualize JSON  

```
1 {
2   "timestamp": "2023-11-23T08:55:15.189+00:00",
3   "status": 404,
4   "error": "Not Found",
5   "path": "/api/v1/usuarios/2"
6 }
```

Base de datos:

