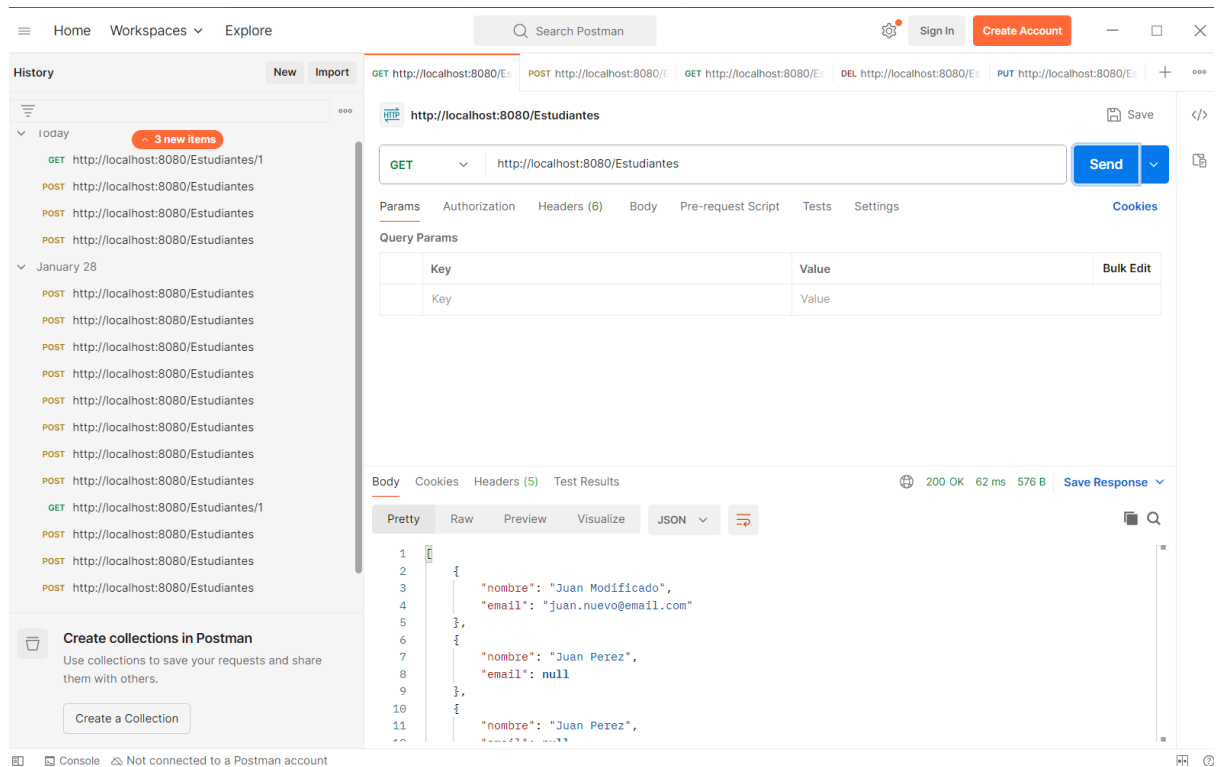


## 1. Listar todos los estudiantes



The screenshot shows the Postman interface with a GET request to `http://localhost:8080/Estudiantes`. The response is a JSON array of student objects. The status is 200 OK, 62 ms, 576 B.

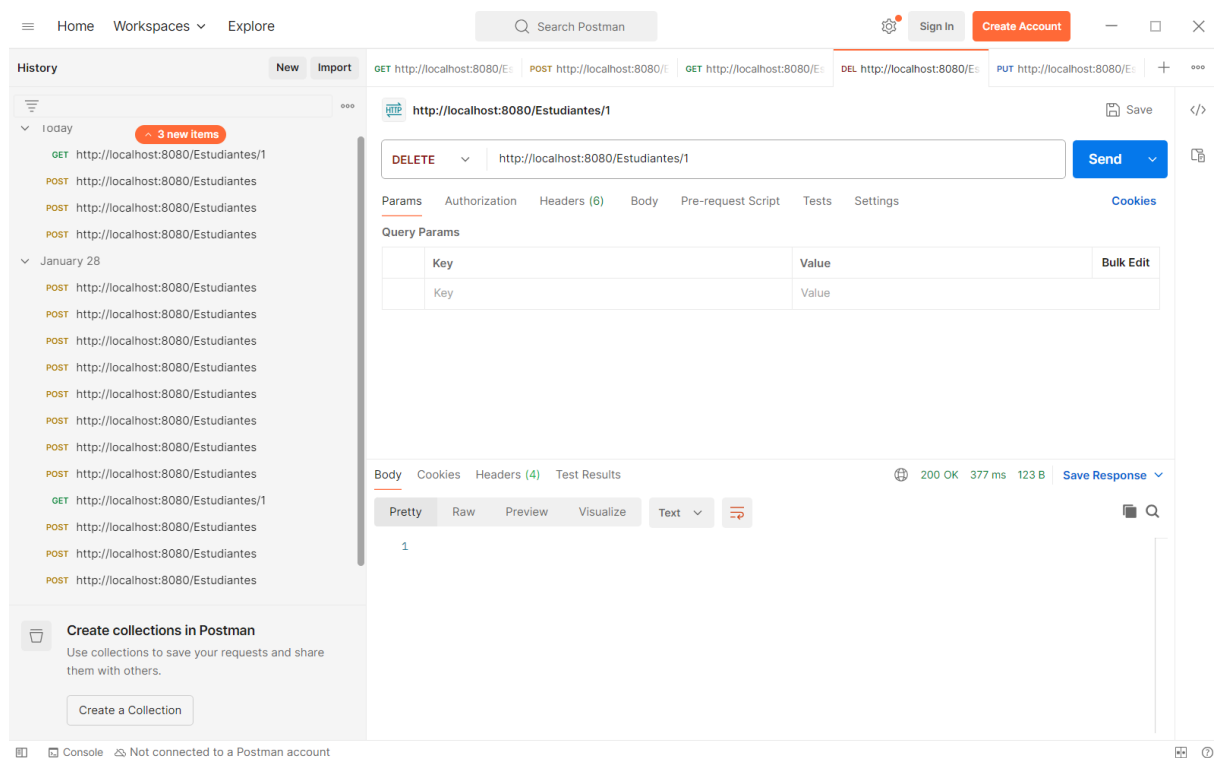
**Query Params**

Key	Value	Bulk Edit
Key	Value	

**Body** (JSON)

```
1 {
2   {
3     "nombre": "Juan Modificado",
4     "email": "juan.nuevo@email.com"
5   },
6   {
7     "nombre": "Juan Perez",
8     "email": null
9   },
10  {
11    "nombre": "Juan Perez",
12    "email": null
13  }
14 }
```

## 2. Eliminar estudiantes por su id



The screenshot shows the Postman interface with a DELETE request to `http://localhost:8080/Estudiantes/1`. The response is a 200 OK status, 377 ms, 123 B.

**Query Params**

Key	Value	Bulk Edit
Key	Value	

**Body** (Text)

```
1
```

### 3. Obtener estudiante por su id

The screenshot shows the Postman interface with a GET request to `http://localhost:8080/estudiantes/3`. The response is a JSON object with the following structure:

```
1 {
2   "nombre": "Juan Perez",
3   "email": null,
4   "links": {
5     "self": {
6       "href": "http://localhost:8080/estudiantes/3"
7     },
8     "estudiante": {
9       "href": "http://localhost:8080/estudiantes/3"
10    }
11  }
12 }
```

The status is 200 OK, 48 ms, 490 B. The response is saved.

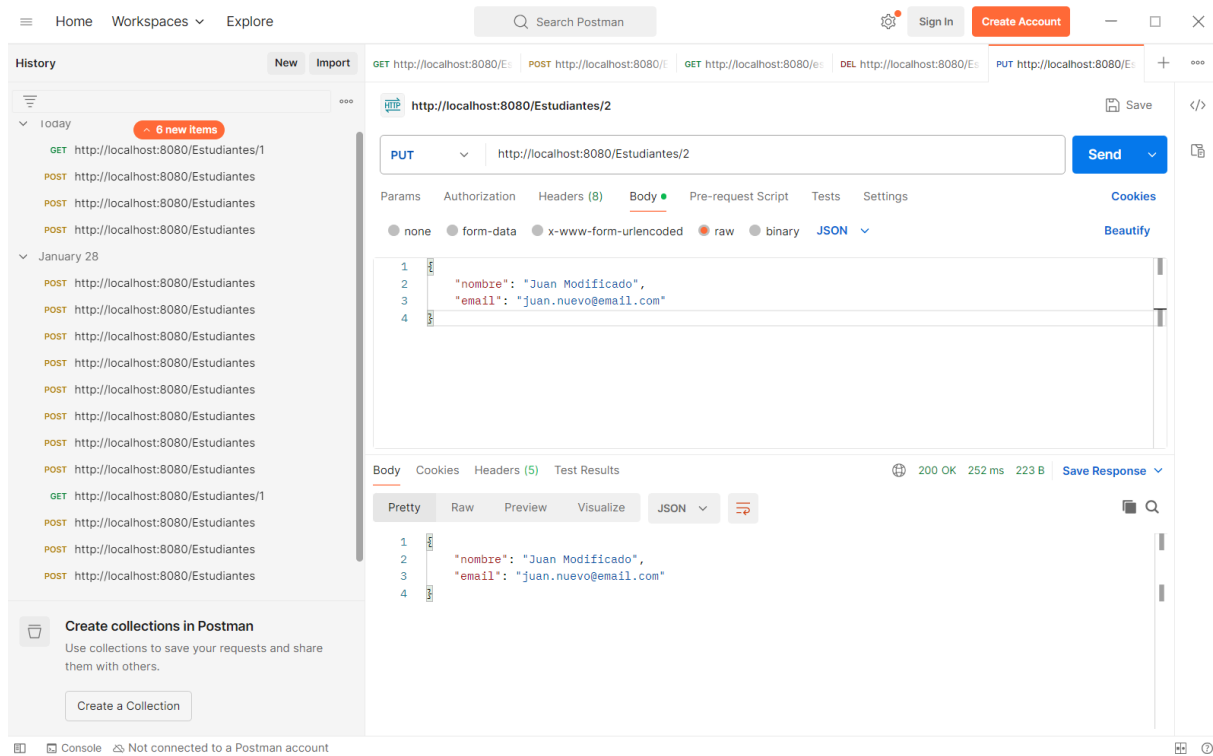
### 4. insertar estudiantes

The screenshot shows the Postman interface with a POST request to `http://localhost:8080/Estudiantes`. The request body is a JSON object with the following structure:

```
1 {
2   "nombre": "Luis Gomez",
3   "email": "luis.gomez@gmail.com"
4 }
```

The status is 200 OK, 1158 ms, 218 B. The response is saved.

5. Actualizar estudiante por su id



Mi Base de datos:

