

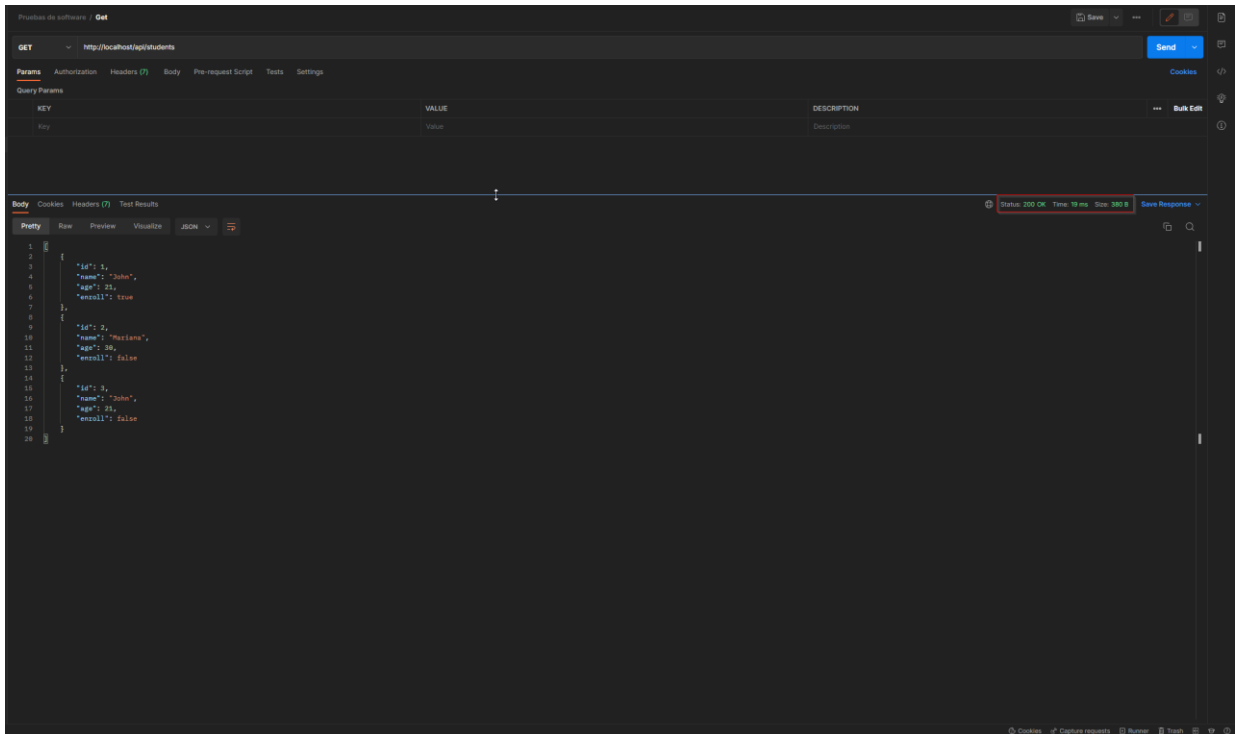
	Primer Entregable	Código: FO-DOC-12
		Versión: 01
		Fecha de Aprobación: Mayo 02 de 2018
		Página 1 de 7

COPIA CONTROLADA

FACULTAD/DEPENDENCIA: Ingeniería		FECHA DE ENTREGA		
PROGRAMA ACADÉMICO/ ÁREA: Pruebas de software		DD 14	MM 11	AAAA 2022
ASIGNATURA: Pruebas de software		PROFESOR: Daniel Felipe Agudelo		
TIPO DE EVALUACIÓN		PORCENTAJE:	CALIFICACIÓN:	
ALUMNOS:(Nombres Apellido - Identificación): Juan Fernando Muñoz 1036604143				

CP1-Connsulta exitosa	Realizar consulta exitosa
Pasos para seguir	*Realizar petición a endpoint
Resultado esperado	*Código de respuesta 200 *Tiempo de respuesta >100ms *Respuesta con los atributos esperados

Consulta exitosa



GET http://localhost:3000/api/students

Status: 200 OK Time: 10 ms Size: 382 B

```

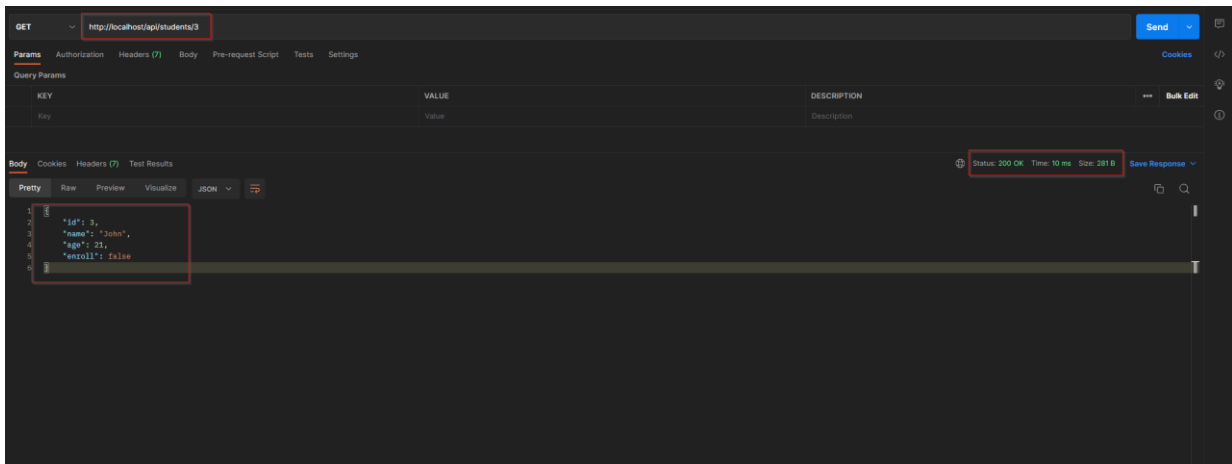
1 {
2   {
3     "id": 1,
4     "name": "John",
5     "age": 21,
6     "email": true
7   },
8   {
9     "id": 2,
10    "name": "Marlene",
11    "age": 38,
12    "email": false
13  },
14  {
15    "id": 3,
16    "name": "John",
17    "age": 21,
18    "email": false
19  }
20 }

```

	Primer Entregable	Código: FO-DOC-12
		Versión: 01
		Fecha de Aprobación: Mayo 02 de 2018
		Página 2 de 7

COPIA CONTROLADA

CP2-Consultar por ID	Realizar proceso de búsqueda por ID
Pasos para seguir	*Realizar petición a endpoint
Resultado esperado	*Código de respuesta 200 *Tiempo de respuesta >100ms *Respuesta con solo el atributo solicitado



GET http://localhost/api/students/3

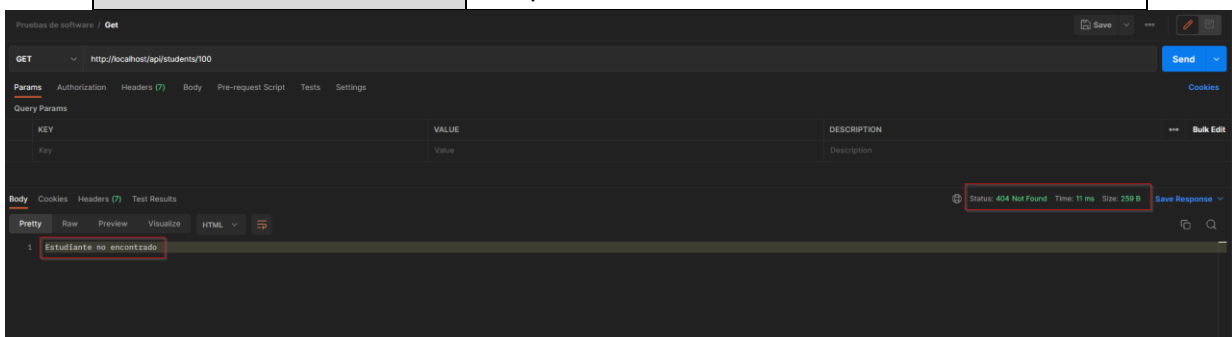
Status: 200 OK Time: 10 ms Size: 281 B

```

{
  "id": 3,
  "name": "John",
  "age": 21,
  "enroll": false
}

```

CP3-Consultar por ID no registrado	Realizar proceso de búsqueda por ID no registrado
Pasos para seguir	*Realizar petición a endpoint
Resultado esperado	*Código de respuesta 404 *Tiempo de respuesta >100ms *Respuesta Estudiante no encontrado



GET http://localhost/api/students/100

Status: 404 Not Found Time: 11 ms Size: 259 B

```

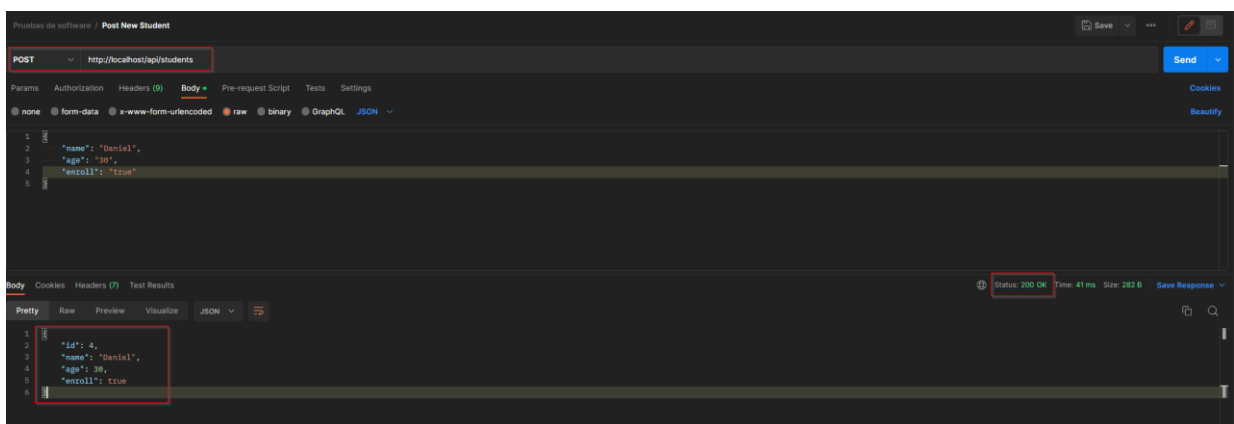
1 Estudiante no encontrado

```

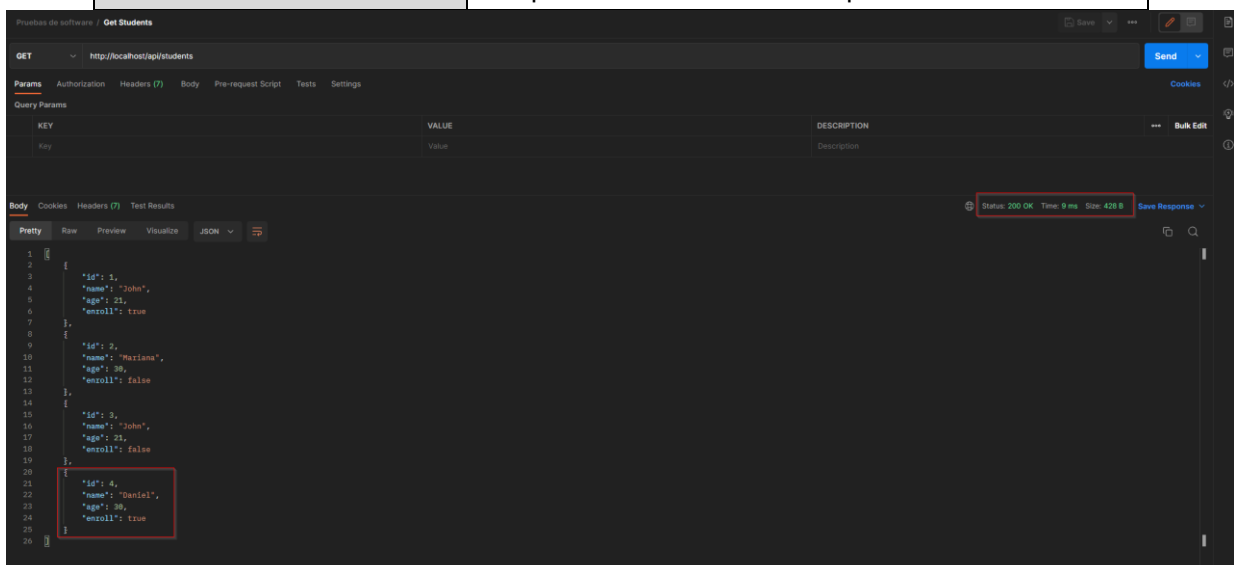
	Primer Entregable	Código: FO-DOC-12
		Versión: 01
		Fecha de Aprobación: Mayo 02 de 2018
		Página 3 de 7

COPIA CONTROLADA

CP4-Registrar un nuevo estudiante	Realizar proceso de registro de un nuevo estudiante
Pasos para seguir	*Realizar petición a endpoint
Resultado esperado	*Código de respuesta 200 *Tiempo de respuesta >100ms *Respuesta con atributos esperados



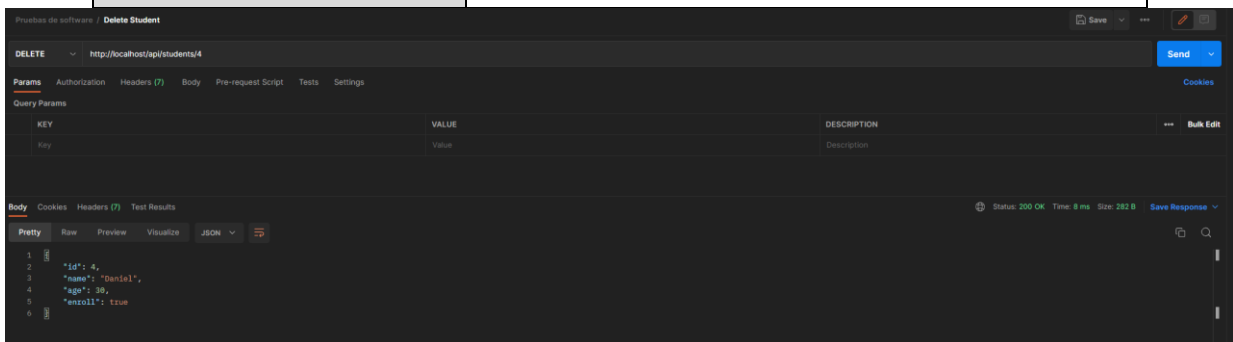
CP5-Connsulta nuevo registro	Realizar consulta del nuevo registro ingresado
Pasos para seguir	*Realizar petición a endpoint
Resultado esperado	*Código de respuesta 200 *Tiempo de respuesta >100ms *Respuesta con los atributos esperados



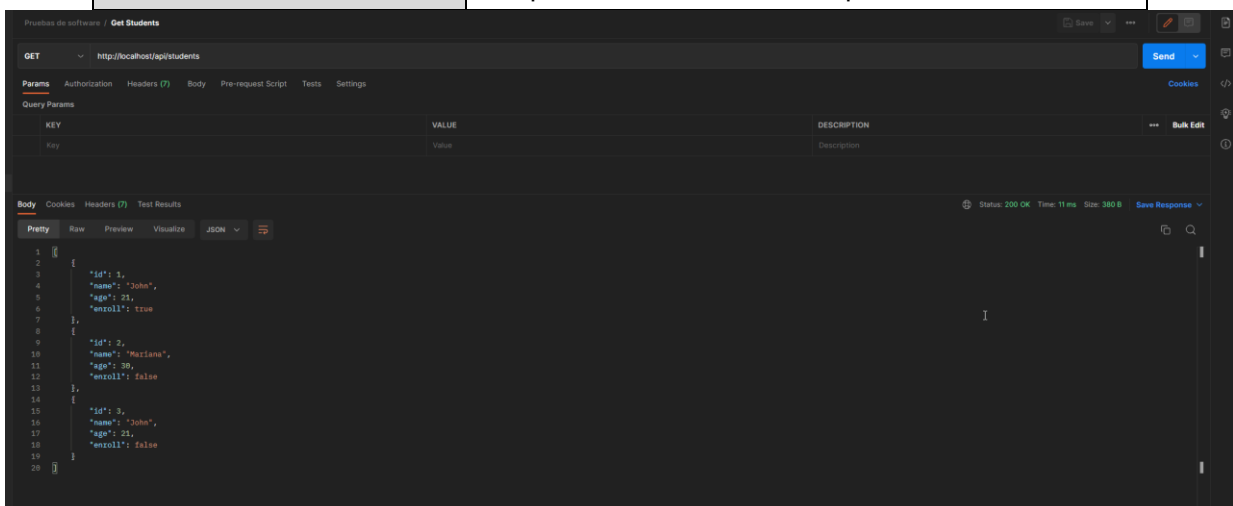
	Primer Entregable	Código: FO-DOC-12
		Versión: 01
		Fecha de Aprobación: Mayo 02 de 2018
		Página 4 de 7

COPIA CONTROLADA

CP6-Eliminar registro	Realizar eliminación de un registro
Pasos para seguir	*Realizar petición a endpoint
Resultado esperado	*Código de respuesta 200 *Tiempo de respuesta >100ms *Respuesta con los atributos esperados



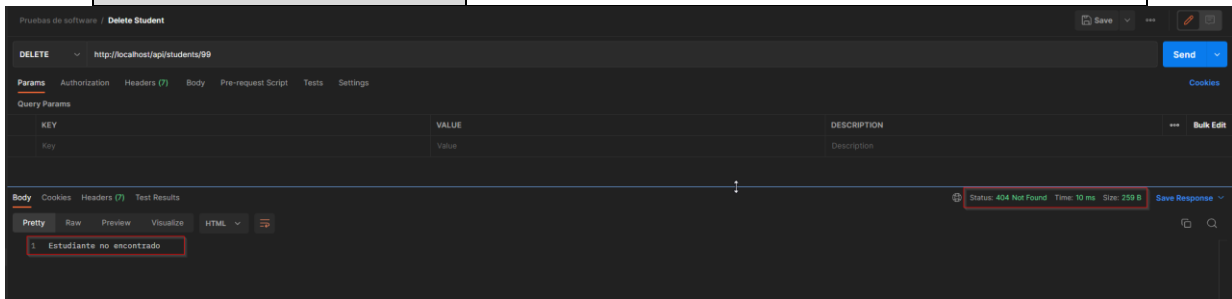
CP7-Consultar registro eliminado	Realizar consulta para validar q el registro fue eliminado
Pasos para seguir	*Realizar petición a endpoint
Resultado esperado	*Código de respuesta 200 *Tiempo de respuesta >100ms *Respuesta con los atributos esperados



	Primer Entregable	Código: FO-DOC-12
		Versión: 01
		Fecha de Aprobación: Mayo 02 de 2018
		Página 5 de 7

COPIA CONTROLADA

CP8-Eliminar registro que no esté en BD	Realizar eliminación de un registro que no se encuentre en la base de datos
Pasos para seguir	*Realizar petición a endpoint
Resultado esperado	*Código de respuesta 200 *Tiempo de respuesta >100ms *Respuesta con los atributos esperados



The screenshot shows a REST client interface with the following details:

- Method:** DELETE
- URL:** http://localhost/api/students/99
- Response Status:** 404 Not Found
- Response Time:** 10 ms
- Response Size:** 259 B
- Response Body:** Estudiante no encontrado

	Primer Entregable	Código: FO-DOC-12
		Versión: 01
		Fecha de Aprobación: Mayo 02 de 2018
		Página 6 de 7

COPIA CONTROLADA

Código creado

```
const express = require('express');
const app = express();

app.use(express.json());

const students = [
  {id: 1, name: 'John', age: 21, enroll: true},
  {id: 2, name: 'Mariana', age: 30, enroll: false},
  {id: 3, name: 'John', age: 21, enroll: false},
];

app.get('/', (req, res) =>{
  res.send('Node JS api');
});

app.get('/api/students', (req, res) =>{
  res.send(students);
});

app.get('/api/students/:id', (req, res) =>{
  const student = students.find(c => c.id === parseInt(req.params.id));
  if(!student) return res.status(404).send('Estudiante no encontrado');
  else res.send(student);
})

app.post('/api/students', (req, res) =>{
  const student ={
    id: students.length + 1,
    name: req.body.name,
    age: parseInt(req.body.age),
    enroll: (req.body.enroll === 'true')
  };

  students.push(student);
  res.send(student);
});

app.delete('/api/students/:id', function(req, res) {
```

	Primer Entregable	Código: FO-DOC-12
		Versión: 01
		Fecha de Aprobación: Mayo 02 de 2018
		Página 7 de 7

COPIA CONTROLADA

```

const student = students.find(c => c.id === parseInt(req.params.id));
if(!student) return res.status(404).send('Estudiante no encontrado');

const index = students.indexOf(student);
students.splice(index, 1);
res.send(student);
});

const port = process.env.port || 80;
app.listen(port, () => console.log('Listening on port ${port}...'));

```