

# Pruebas Postman

## Contenido

Estudiante .....	2
Crear Estudiante.....	2
Crear Estudiante Invalido .....	2
Eliminar Estudiante.....	3
Eliminar estudiante inválido .....	3
Evaluación.....	4
Crear Evaluación .....	4
Crear Evaluación Inválida .....	4
Profesor.....	5
Crear Profesor .....	5
Crear Profesor Invalido .....	5
Asignar Evaluación .....	6
Asignar Evaluación Inválida .....	6
Proyecto .....	7
Crear Proyecto.....	7
Crear Proyecto Inválido .....	7
Avanzar Proyecto .....	8
Avanzar Proyecto Inválido.....	8
Find estudiante.....	9
Find Estudiante Inválido .....	9
Justificaciones .....	9

# Estudiante

## Crear Estudiante

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

- Method:** POST
- URL:** `{{url_base}}/estudiantes`
- Body (raw):**

```
1 {
2   "nombre": "Alejandro Charry",
3   "cedula": 1090366478,
4   "semestre": 7,
5   "programa": "Ingenieria de Sistemas",
6   "promedio": 375
7 }
```
- Response:** 201 Created, 47 ms, 362 B. The response body (JSON) is:

```
1 {
2   "id": 1,
3   "nombre": "Alejandro Charry",
4   "cedula": 1090366478,
5   "semestre": 7,
6   "programa": "Ingenieria de Sistemas",
7   "promedio": 375
8 }
```

## Crear Estudiante Invalido

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

- Method:** POST
- URL:** `{{url_base}}/estudiantes`
- Body (raw):**

```
1 {
2   "nombre": "Luis Pérez",
3   "cedula": 123456789,
4   "semestre": 2,
5   "programa": "Ingeniería Química",
6   "promedio": 2.9
7 }
```
- Response:** 400 Bad Request, 48 ms, 362 B. The response body (JSON) is:

```
1 {
2   "message": "El promedio debe ser mayor a 3.2 y el semestre igual o mayor a 4",
3   "error": "Bad Request",
4   "statusCode": 400
5 }
```

## Eliminar Estudiante

The screenshot shows a REST client interface with a tab labeled "DEL EliminarEstudiante". The URL bar contains the endpoint `{{url_base}} /estudiantes/1` with a "Send" button. The "Body" tab is selected, showing the message "This request does not have a body". The status bar at the bottom indicates a successful "204 No Content" response with a 27 ms duration and 134 B of data.

Params Authorization Headers (6) **Body** Scripts Settings Cookies

☒ none ☐ form-data ☐ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

This request does not have a body

Body Cookies Headers (4) Test Results 204 No Content • 27 ms • 134 B Save Response

Raw Preview Visualize

1

## Eliminar estudiante inválido

The screenshot shows a REST client interface with a tab labeled "DEL EliminarEstudianteInvalid". The URL bar contains the endpoint `{{url_base}} /estudiantes/00000000-0000-0000-0000-000000000000` with a "Send" button. The "Body" tab is selected, showing a JSON response indicating an invalid student. The status bar at the bottom indicates a "400 Bad Request" response with a 12 ms duration and 321 B of data.

Params Authorization Headers (6) **Body** Scripts Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

Body Cookies Headers (7) Test Results 400 Bad Request • 12 ms • 321 B Save Response

JSON Preview Visualize

```
1 {
2   "message": "Estudiante no encontrado",
3   "error": "Bad Request",
4   "statusCode": 400
5 }
```

# Evaluación

## Crear Evaluación

The screenshot shows a REST client interface with a tab titled "POST CrearEvaluacion". The URL is set to `{{url_base}} /evaluaciones`. The request method is "POST". The "Body" tab is selected, showing a JSON payload:

```
1 {
2   "proyecto": {
3     "id": 1,
4     "titulo": "Titulo del proyecto con más de 15 caracteres",
5     "area": "Área del proyecto",
6     "presupuesto": 5000,
7     "notafinal": 3,
8     "estado": 4,
9     "fechainicio": "2023-01-01T00:00:00.000Z",
10    "fechafin": "2026-01-01T00:00:00.000Z",
11    "evaluaciones": [],
12    "mentor": {
13      "id": 1,
14      "nombre": "Luis Contreras",
15    }
16  }
17 }
```

The response status is "201 Created" with a response time of 8 ms and a size of 939 B. The "Body" tab shows the response JSON:

```
1 {
2   "id": 1,
3   "proyecto": {
4     "id": 1,
5     "titulo": "Titulo del proyecto con más de 15 caracteres",
6     "area": "Área del proyecto",
7     "presupuesto": 5000,
8     "notafinal": 3,
9     "estado": 4,
10    "fechainicio": "2023-01-01T00:00:00.000Z",
11    "fechafin": "2026-01-01T00:00:00.000Z",
12    "evaluaciones": [],
13    "mentor": {
14      "id": 1,
15      "nombre": "Luis Contreras",
16    }
17  }
18 }
```

## Crear Evaluación Inválida

The screenshot shows a REST client interface with a tab titled "POST CrearEvaluacionInvalida". The URL is set to `{{url_base}} /evaluaciones`. The request method is "POST". The "Body" tab is selected, showing a JSON payload:

```
1 {
2   "proyecto": {
3     "id": 1,
4     "titulo": "Titulo del proyecto con más de 15 caracteres",
5     "area": "Área del proyecto",
6     "presupuesto": 5000,
7     "notafinal": 3,
8     "estado": 4,
9     "fechainicio": "2023-01-01T00:00:00.000Z",
10  }
11 }
```

The response status is "400 Bad Request" with a response time of 13 ms and a size of 345 B. The "Body" tab shows the response JSON:

```
1 {
2   "message": "El proyecto no puede ser evaluado por su mentor",
3   "error": "Bad Request",
4   "statusCode": 400
5 }
```

## Profesor

### Crear Profesor

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

- Request:**
  - Method: **POST**
  - URL: `{{url_base}}/profesores`
  - Body (raw):

```
1 {
2   "nombre": "Luis Contreras",
3   "cedula": 1056325478,
4   "departamento": "Ciencias Computacionales",
5   "extension": 12345,
6   "esParEvaluador": false
7 }
```
- Response:**
  - Status: **201 Created**
  - Time: 14 ms
  - Size: 378 B
  - Body (JSON):

```
1 {
2   "id": 1,
3   "cedula": 1056325478,
4   "nombre": "Luis Contreras",
5   "departamento": "Ciencias Computacionales",
6   "extension": 12345,
7   "esParEvaluador": false
8 }
```

### Crear Profesor Invalido

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

- Request:**
  - Method: **POST**
  - URL: `{{url_base}}/profesores`
  - Body (raw):

```
1 {
2   "nombre": "Maria Herrera",
3   "cedula": 1056999999,
4   "departamento": "Matemáticas",
5   "extension": 123456,
6   "esParEvaluador": true
7 }
```
- Response:**
  - Status: **400 Bad Request**
  - Time: 37 ms
  - Size: 345 B
  - Body (JSON):

```
1 {
2   "message": "La extensión debe tener exactamente 5 digitos",
3   "error": "Bad Request",
4   "statusCode": 400
5 }
```

## Asignar Evaluación

The screenshot shows a REST client interface for a route named `AsignarEvaluador`. The request method is `PATCH` and the URL is `{{url_base}}/profesores/1/asignar-evaluador/1`. The request body is a JSON object:

```
1 {
2   "profesorId": 1,
3   "evaluacionId": 1
4 }
```

The response status is `200 OK` with a response time of `41 ms` and a size of `399 B`. The response body is a JSON object:

```
1 {
2   "id": 1,
3   "cedula": 1056325478,
4   "nombre": "Luis Contreras",
5   "departamento": "Ciencias Computacionales",
6   "extension": 12345,
7   "esParEvaluador": false,
8   "evaluaciones": [
9     {
10      "id": 1
11     }
12 ]
13 }
```

## Asignar Evaluación Inválida

The screenshot shows a REST client interface for a route named `AsignarEvaluadorInvalido`. The request method is `PATCH` and the URL is `{{url_base}}/profesores/2/asignar-evaluador/1`. The request body is empty, indicated by the text "This request does not have a body". The response status is `400 Bad Request` with a response time of `27 ms` and a size of `319 B`. The response body is a JSON object:

```
1 {
2   "message": "Profesor no encontrado",
3   "error": "Bad Request",
4   "statusCode": 400
5 }
```

# Proyecto

## Crear Proyecto

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

- Method:** POST
- URL:** `{{url_base}}/proyectos`
- Body (raw):**

```
1 {
2   "titulo": "Título del proyecto con más de 15 caracteres",
3   "area": "Área del proyecto",
4   "presupuesto": 5000,
5   "notafinal": 3,
6   "estado": 1,
7   "fechainicio": "2023-01-01T00:00:00.000Z",
8 }
```
- Response (JSON):**

```
1 {
2   "id": 1,
3   "titulo": "Título del proyecto con más de 15 caracteres",
4   "area": "Área del proyecto",
5   "presupuesto": 5000,
6   "notafinal": 3,
7   "estado": 1,
8   "fechainicio": "2023-01-01T00:00:00.000Z",
9   "fechafin": "2026-01-01T00:00:00.000Z",
10  "evaluaciones": []
11 }
```
- Status:** 201 Created
- Time:** 23 ms
- Size:** 476 B

## Crear Proyecto Inválido

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

- Method:** POST
- URL:** `{{url_base}}/proyectos`
- Body (raw):**

```
1 {
2   "titulo": "proyecto",
3   "area": "Área del proyecto",
4   "presupuesto": 5000,
5   "notafinal": 3,
6   "estado": 1,
7   "fechainicio": "2023-01-01T00:00:00.000Z",
8 }
```
- Response (JSON):**

```
1 {
2   "message": "El titulo debe tener más de 15 caracteres",
3   "error": "Bad Request",
4   "statusCode": 400
5 }
```
- Status:** 400 Bad Request
- Time:** 14 ms
- Size:** 340 B

## Avanzar Proyecto

The screenshot shows a REST client interface with a tab labeled "PUT Avanz". The URL is `{{url_base}}/proyectos/proyectoid`. The request body is a JSON object: `{ "proyectoId": 1 }`. The response status is **200 OK** with a response time of 24 ms and a size of 453 B. The response body is a JSON object: `{ "id": 1, "titulo": "Título del proyecto con más de 15 caracteres", "area": "Área del proyecto", "presupuesto": 5000, "notafinal": 3, "estado": 2, "fechainicio": "2023-01-01T00:00:00.000Z", "fechafin": "2026-01-01T00:00:00.000Z" }`.

```
PUT {{url_base}}/proyectos/proyectoid
```

```
{
  "proyectoId": 1
}
```

**200 OK** • 24 ms • 453 B

```
{
  "id": 1,
  "titulo": "Título del proyecto con más de 15 caracteres",
  "area": "Área del proyecto",
  "presupuesto": 5000,
  "notafinal": 3,
  "estado": 2,
  "fechainicio": "2023-01-01T00:00:00.000Z",
  "fechafin": "2026-01-01T00:00:00.000Z"
}
```

## Avanzar Proyecto Inválido

The screenshot shows a REST client interface with a tab labeled "PUT Avanz". The URL is `{{url_base}}/proyectos/proyectoid`. The request body is a JSON object: `{ "proyectoId": 2 }`. The response status is **400 Bad Request** with a response time of 11 ms and a size of 338 B. The response body is a JSON object: `{ "message": "El proyecto ya está en su máximo estado", "error": "Bad Request", "statusCode": 400 }`.

```
PUT {{url_base}}/proyectos/proyectoid
```

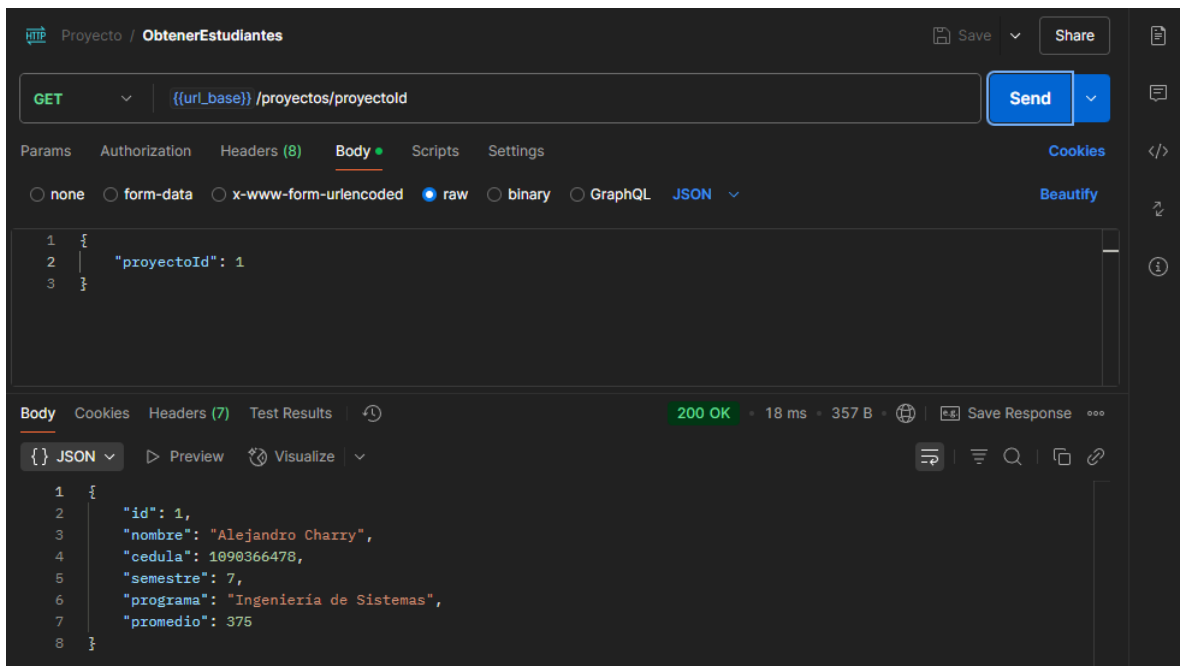
```
{
  "proyectoId": 2
}
```

**400 Bad Request** • 11 ms • 338 B

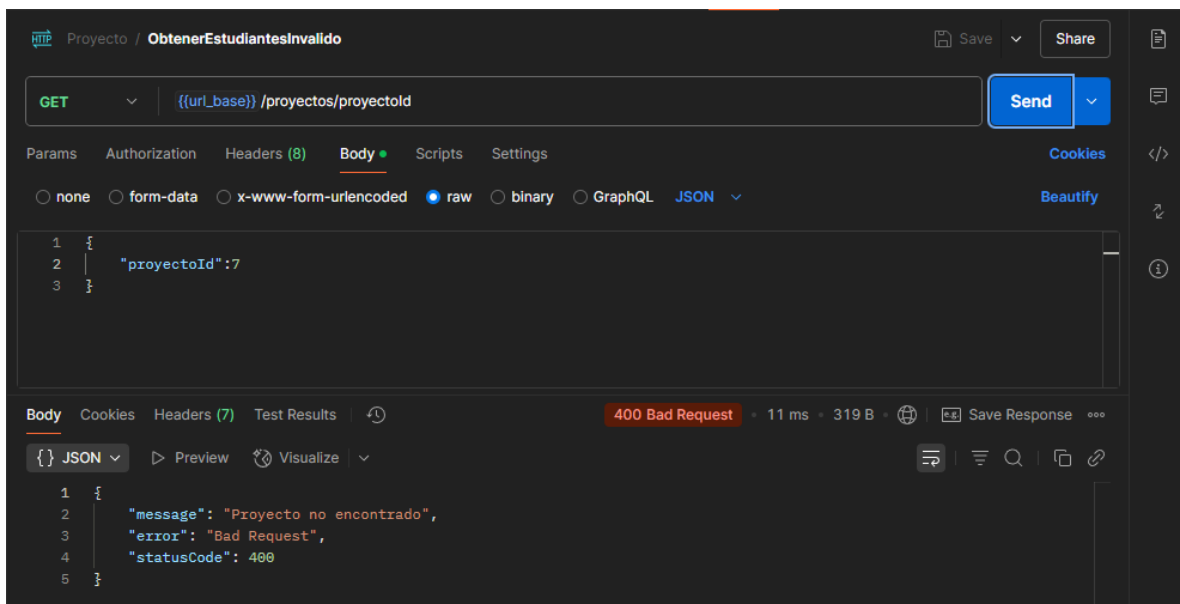
```
{
  "message": "El proyecto ya está en su máximo estado",
  "error": "Bad Request",
  "statusCode": 400
}
```



## Find estudiante



## Find Estudiante Inválido



## Justificaciones

Para las pruebas postman, se realizó la colección completa bajo un environment con variable `{{url_base}}` = <http://localhost:3000/>. Adicionalmente, los esquemas de cada prueba para cada clase, es el mismo definido dentro de los services que se crearon con faker para las pruebas de lógica. Este mismo esquema responde a las entidades

correspondientes de cada clase, donde se definieron los atributos obligatorios según el enunciado del parcial.