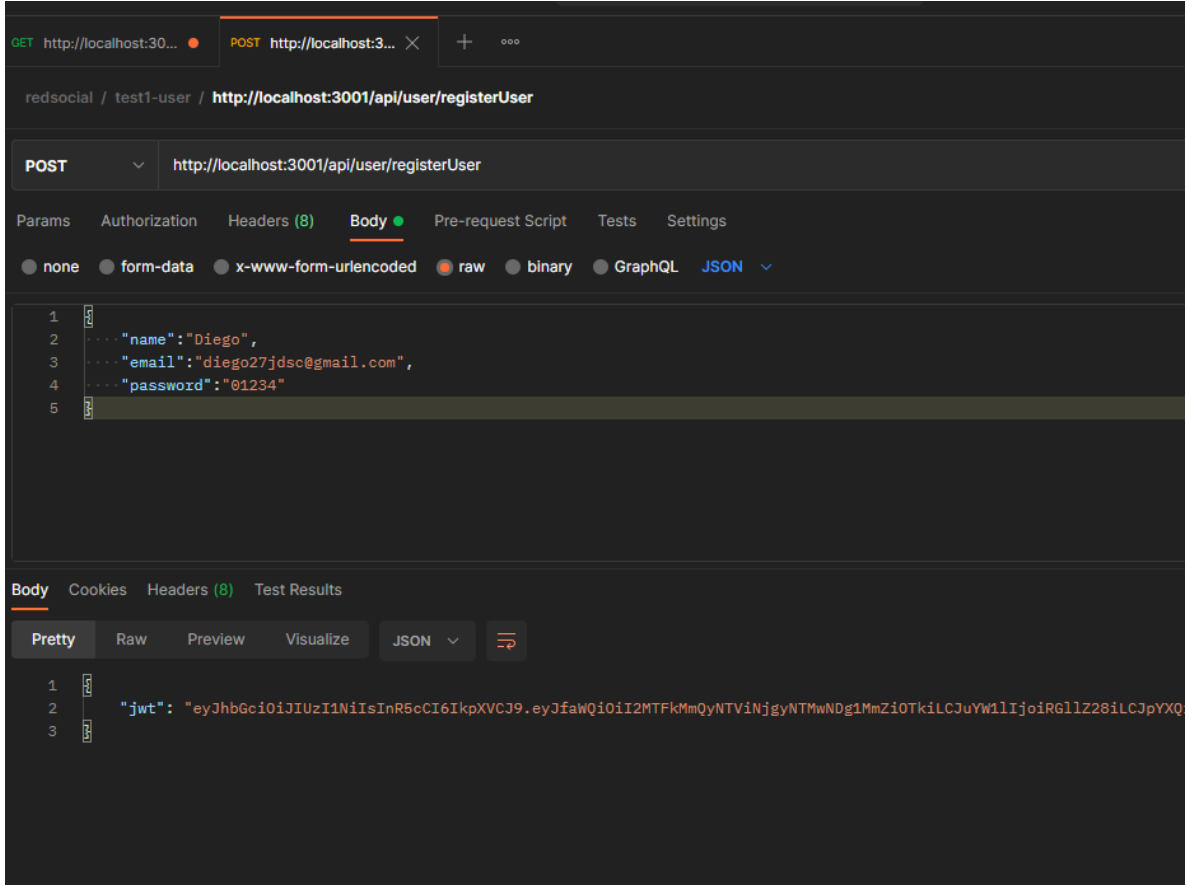


## Pantallazos, Programa Funcional

- **Usuarios:**

1)



2)

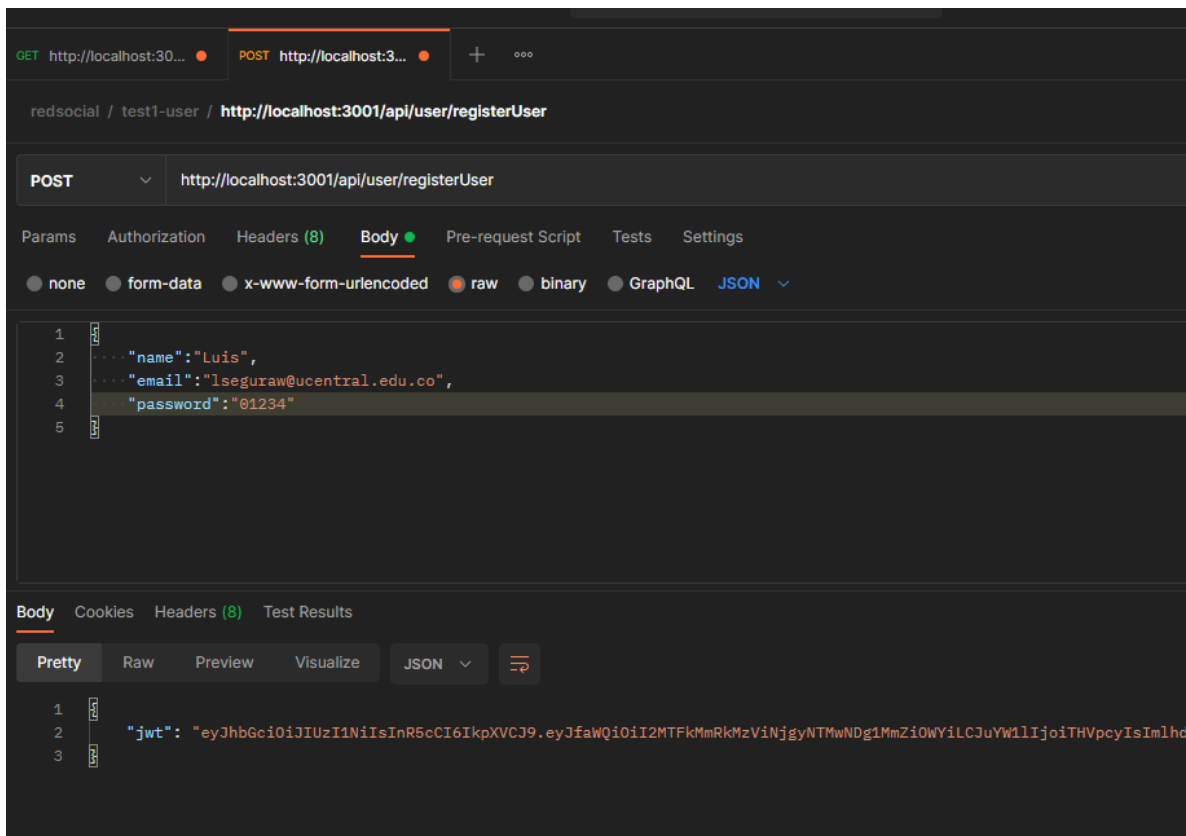
The screenshot shows a REST client interface with a dark theme. At the top, there's a breadcrumb trail: `redsocial / test1-user / http://localhost:3001/api/user/registerUser`. Below this, the request method is set to **POST** and the URL is `http://localhost:3001/api/user/registerUser`. The 'Body' tab is selected, showing a JSON payload: 

```
{  "name": "Felipe",  "email": "jramirez@ucentral.edu.co",  "password": "01234"}
```

. The bottom section shows the response body in the 'Body' tab, displaying a JWT token: 

```
{  "jwt": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiI2MTFkMmRhNDViNjgyNTMwNDg1MmZiOWMiLCJyYW11IjoiaWVsaXB1IiwiaWF0Ijo..."}
```

3)



4)

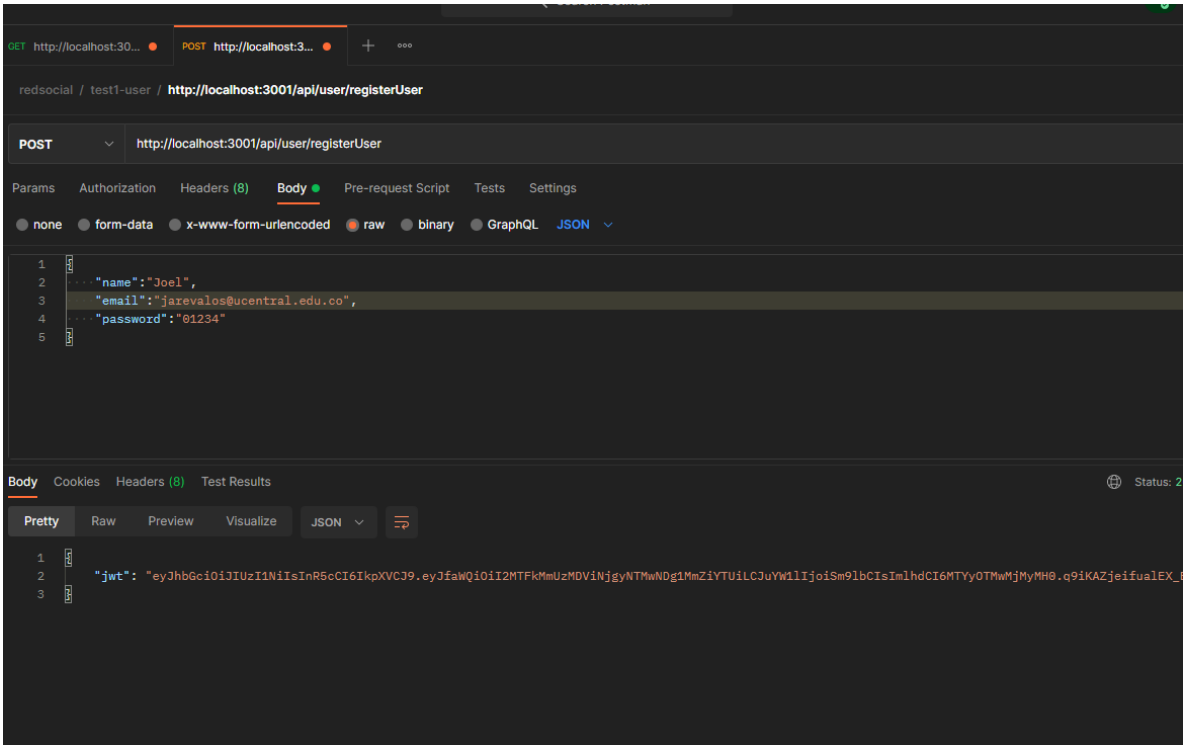
The screenshot shows a REST client interface with a dark theme. At the top, there are tabs for different requests: a GET request to `http://localhost:30...` and a selected POST request to `http://localhost:3001/api/user/registerUser`. Below the tabs, the URL bar shows `redsocial / test1-user / http://localhost:3001/api/user/registerUser`. The main area is divided into several tabs: Params, Authorization, Headers (8), Body (selected), Pre-request Script, Tests, and Settings. Under the Body tab, there are radio buttons for different content types: none, form-data, x-www-form-urlencoded, raw (selected), binary, and GraphQL. A dropdown menu shows 'JSON' is selected. The request body is displayed in a code editor with line numbers 1 to 5, containing a JSON object: 

```
1 {
2   "name": "Camilo",
3   "email": "cguerreroa@ucentral.edu.co",
4   "password": "01234"
5 }
```

 Below the request body, there are tabs for Body (selected), Cookies, Headers (8), and Test Results. Under the Body tab, there are buttons for Pretty (selected), Raw, Preview, Visualize, and a JSON dropdown. The response body is displayed in a code editor with line numbers 1 to 3, showing a JSON object with a JWT token: 

```
1 {
2   "jwt": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiI2MTFkMmRmMDViNjgyNTMwNDg1MmZiYTIiLCJyZW1lIjoia2FtaWxvIi
3 }
```

5)



## Usuario Mongo General

MongoDB Compass - localhost:27017/redsocial.users

Connect View Collection Help

Local

> 10 DBS 16 COLLECTIONS

☆ FAVORITE

Q Filter your data

> Diegodbscrumboard

> admin

> config

> dbscrumboard

> gamestore

> local

> pruebadb

> pruebadb2

▼ redsocial

users

> retodbscrumboard

+

> \_MONGOSH

redsocial.users Documents

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER { field: 'value' }

ADD DATA VIEW

`{`

`_id: ObjectId("611d2d255b6825304852fb99")`

`name: "Diego"`

`email: "diego27jdsc@gmail.com"`

`password: "$2b$10$F4tpw7rcikoq9pBXTd5eum.SIysEhZ./g9LRMkHmDG3rFDsGj0C"`

`__v: 0`

`{`

`_id: ObjectId("611d2da45b6825304852fb9c")`

`name: "Felipe"`

`email: "jramirez@ucentral.edu.co"`

`password: "$2b$10$b7gabkn2vxd9Nkn88KyK05lnk87BmXH/qyZ097yiFViby3SrLPF6"`

`__v: 0`

`{`

`_id: ObjectId("611d2dd35b6825304852fb9f")`

`name: "Luis"`

`email: "lseguraw@ucentral.edu.co"`

`password: "$2b$10$Apko0ISMYQJNlbnGfQ0D4ea/HU.sQcuKGhVYf.3mueQ880./219ck"`

`__v: 0`

`{`

`_id: ObjectId("611d2df05b6825304852fba2")`

`name: "Camilo"`

`email: "cguerreroa@ucentral.edu.co"`

`password: "$2b$10$W1C2ZH16QT/X4002G5gNOCFVXwe8UN9MmKSwGuFvVKzW2S0iTBa"`

`__v: 0`

`{`

`_id: ObjectId("611d2e305b6825304852fba5")`

`name: "Joel"`

`email: "jarevalos@ucentral.edu.co"`

`password: "$2b$10$NKG9zXQdof18yIpQaz0EAu.lQjC7ifF6.CeGp0xEksBVCdwg4Gasm"`

`__v: 0`

# Login usuarios y post

## 1) Login con Diego

The screenshot displays a REST client interface with a dark theme. At the top, a tab shows a POST request to `http://localhost:3001/api/auth/login`. Below the tab, the request details are visible, including the method (POST) and the URL. The 'Body' tab is selected, showing a JSON payload with the following structure:

```
1 {
2   "email": "diego27jdsc@gmail.com",
3   "password": "01234"
4 }
```

Below the request body, the response is shown in the 'Body' tab. The response is a JSON object containing a JWT token:

```
1 {
2   "jwtToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiI2MTFkMmQyNTViNjgyNTMwNDg1MmZiOTk1LCJ1bm50IjoiZm9udCJ9"
3 }
```

## Post con Diego

a)

The screenshot shows a REST client interface with a POST request to `http://localhost:3001/api/post/savePost`. The request body is a JSON object with `text` and `hashtag` fields. The response is a JSON object with a `result` field containing user and post details.

**Request:**

```
POST http://localhost:3001/api/post/savePost
```

**Body:**

```
{  "text": "Quien pa jugar lol",  "hashtag": "#vamosamanquear"}
```

**Response:**

```
{  "result": {    "_id": "611d30155b6825304852fb99",    "userId": "611d2d255b6825304852fb99",    "text": "Quien pa jugar lol",    "hashtag": "#vamosamanquear",    "date": "2021-08-18T16:06:45.840Z",    "__v": 0  }}
```



b)

The screenshot displays a REST client interface with a POST request to `http://localhost:3001/api/post/savePost`. The request body is a JSON object with the following structure:

```
1 {
2   "text": "Hoy es día de farra",
3   "hashtag": "#QuienCopea"
4 }
```

The response body is also in JSON format, showing the saved post details:

```
1 {
2   "result": {
3     "_id": "611d306b5b6825304852fbad",
4     "userId": "611d2d255b6825304852fb99",
5     "text": "Hoy es día de farra",
6     "hashtag": "#QuienCopea",
7     "date": "2021-08-18T16:08:11.805Z",
8     "__v": 0
9   }
10 }
```

c)

The screenshot shows a REST client interface with a POST request to `http://localhost:3001/api/post/savePost`. The request body is a JSON object with `text` and `hashtag` fields. The response is also a JSON object with a `result` field containing a full post object.

**Request:**

```
POST http://localhost:3001/api/post/savePost
```

**Body:**

```
{  1  "text": "Lunes de odio los lunes",
  2  "hashtag": "#lunes"
  3  }
```

**Response:**

```
{  1  "result": {
  2    "_id": "611d308f5b6825304852fbb0",
  3    "userId": "611d2d255b6825304852fb99",
  4    "text": "Lunes de odio los lunes",
  5    "hashtag": "#lunes",
  6    "date": "2021-08-18T16:08:47.253Z",
  7    "__v": 0
  8  }
  9  }
```

d)

The screenshot shows a REST client interface with a POST request to `http://localhost:3001/api/post/savePost`. The request body is a JSON object with `text` and `hashtag` fields. The response body is a JSON object with a `result` field containing a nested object with various fields including `_id`, `userId`, `text`, `hashtag`, `date`, and `__v`.

POST `http://localhost:3001/api/post/savePost`

Params Authorization Headers (9) Body Pre-request Script Tests Settings

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

```
1 {
2   "text": "viernes de siluetas",
3   "hashtag": "#simsons"
4 }
```

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "result": {
3     "_id": "611d30a95b6825304852fbb3",
4     "userId": "611d2d255b6825304852fb99",
5     "text": "viernes de siluetas",
6     "hashtag": "#simsons",
7     "date": "2021-08-18T16:09:13.648Z",
8     "__v": 0
9   }
10 }
```

e)

The screenshot shows a REST client interface with a POST request to `http://localhost:3001/api/post/savePost`. The request body is a JSON object with `text` and `hashtag` fields. The response is also in JSON format, showing a successful save with a `result` object containing an `_id`, `userId`, `text`, `hashtag`, `date`, and `__v` (version).

**Request:**

```
POST http://localhost:3001/api/post/savePost
```

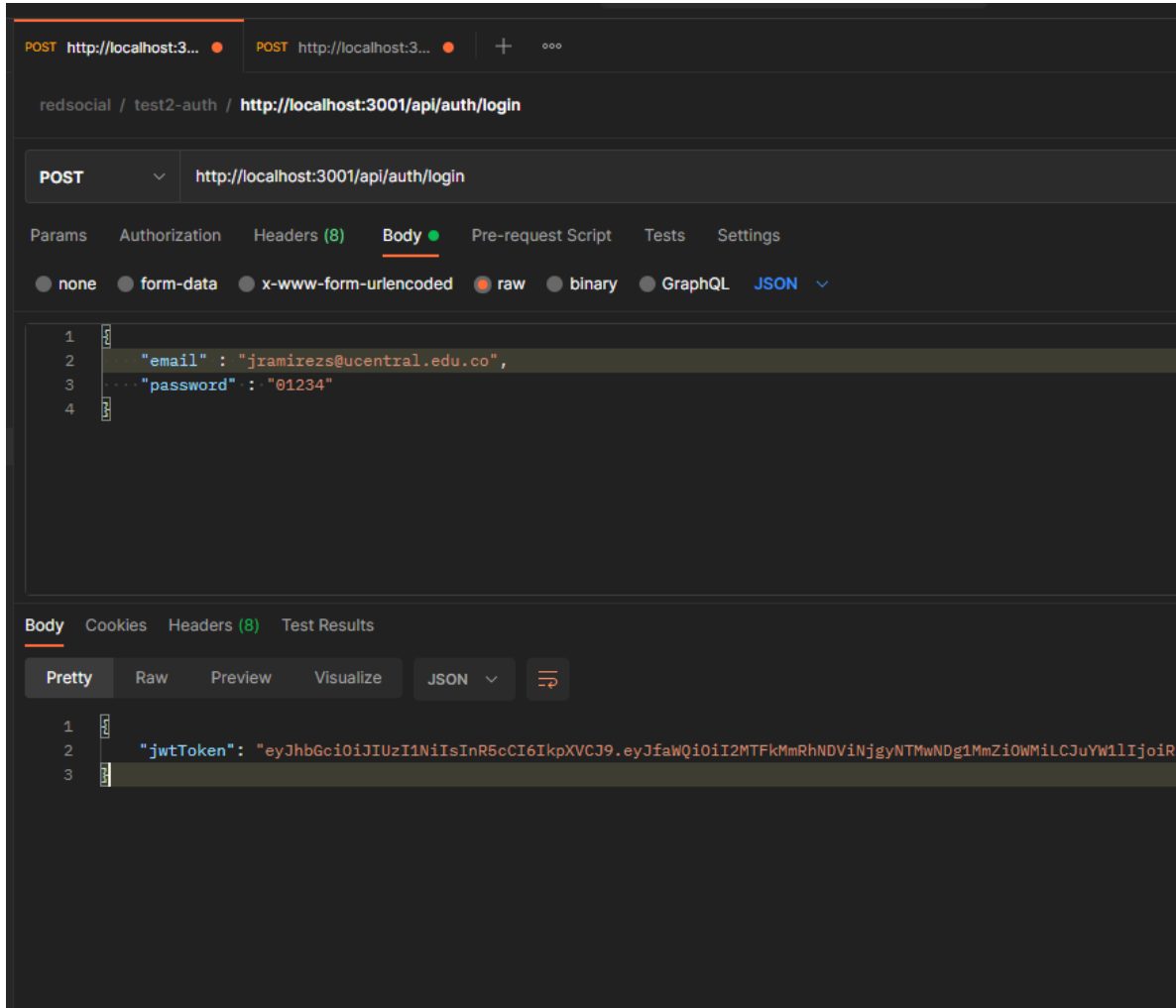
**Body:**

```
{  "text": "Hoy toca vivir",  "hashtag": "#searma"}
```

**Response:**

```
{  "result": {    "_id": "611d30d05b6825304852fbb6",    "userId": "611d2d255b6825304852fb99",    "text": "Hoy toca vivir",    "hashtag": "#searma",    "date": "2021-08-18T16:09:52.456Z",    "__v": 0  }}
```

## 2) Login con Felipe



# Post con Felipe

a)

The screenshot shows a REST client interface with a POST request to `http://localhost:3001/api/post/savePost`. The request body is a JSON object with `text` and `hashtag` fields. The response is a JSON object with a `result` field containing an object with `_id`, `userId`, `text`, `hashtag`, `date`, and `__v` fields.

POST `http://localhost:3001/api/post/savePost`

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings

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

```
1 {
2   "text": "Como es csgo",
3   "hashtag": "#jugar"
4 }
```

**Body** Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "result": {
3     "_id": "611d312c5b6825304852fbba",
4     "userId": "611d2da45b6825304852fb9c",
5     "text": "Como es csgo",
6     "hashtag": "#jugar",
7     "date": "2021-08-18T16:11:24.689Z",
8     "__v": 0
9   }
10 }
```

b)

The screenshot displays a REST client interface with a POST request to `http://localhost:3001/api/post/savePost`. The request body is a JSON object with `text` and `hashtag` fields. The response is also in JSON format, showing a successful save with a `result` object containing an `_id`, `userId`, the saved `text` and `hashtag`, a `date`, and a version number `__v`.

**Request:**

```
POST http://localhost:3001/api/post/savePost
```

**Body:**

```
{  "text": "me toca estudiar noooooo",  "hashtag": "#ucentral"}
```

**Response:**

```
{  "result": {    "_id": "611d31835b6825304852fbdb",    "userId": "611d2da45b6825304852fb9c",    "text": "me toca estudiar noooooo",    "hashtag": "#ucentral",    "date": "2021-08-18T16:12:51.938Z",    "__v": 0  }}}
```

c)

The screenshot displays a REST client interface with a dark theme. At the top, a breadcrumb trail shows 'redsocial / test3-post / http://localhost:3001/api/post/saveTask'. Below this, the request configuration is shown for a POST method to 'http://localhost:3001/api/post/savePost'. The 'Body' tab is selected, showing a JSON payload: 

```
{  "text": "pase stikers de whatss",  "hashtag": "#stickers"}
```

. The bottom section shows the response body in 'Pretty' format, displaying a JSON object: 

```
{  "result": {    "_id": "611d31a25b6825304852fbc0",    "userId": "611d2da45b6825304852fb9c",    "text": "pase stikers de whatss",    "hashtag": "#stickers",    "date": "2021-08-18T16:13:22.532Z",    "__v": 0  }}
```



d)

The screenshot shows a REST client interface with a POST request to `http://localhost:3001/api/post/savePost`. The request body is a JSON object with `text` and `hashtag` fields. The response is a JSON object with a `result` field containing user and post details.

redsocial / test3-post / `http://localhost:3001/api/post/saveTask`

**POST** `http://localhost:3001/api/post/savePost`

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings

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

```
1 {
2   "text": "quien pa ver Harry potter",
3   "hashtag": "#espeliargus"
4 }
```

**Body** Cookies Headers (8) Test Results

Pretty Raw Preview Visualize **JSON**

```
1 {
2   "result": {
3     "_id": "611d31c35b6825304852fbc3",
4     "userId": "611d2da45b6825304852fb9c",
5     "text": "quien pa ver Harry potter",
6     "hashtag": "#espeliargus",
7     "date": "2021-08-18T16:13:55.260Z",
8     "__v": 0
9   }
10 }
```

e)

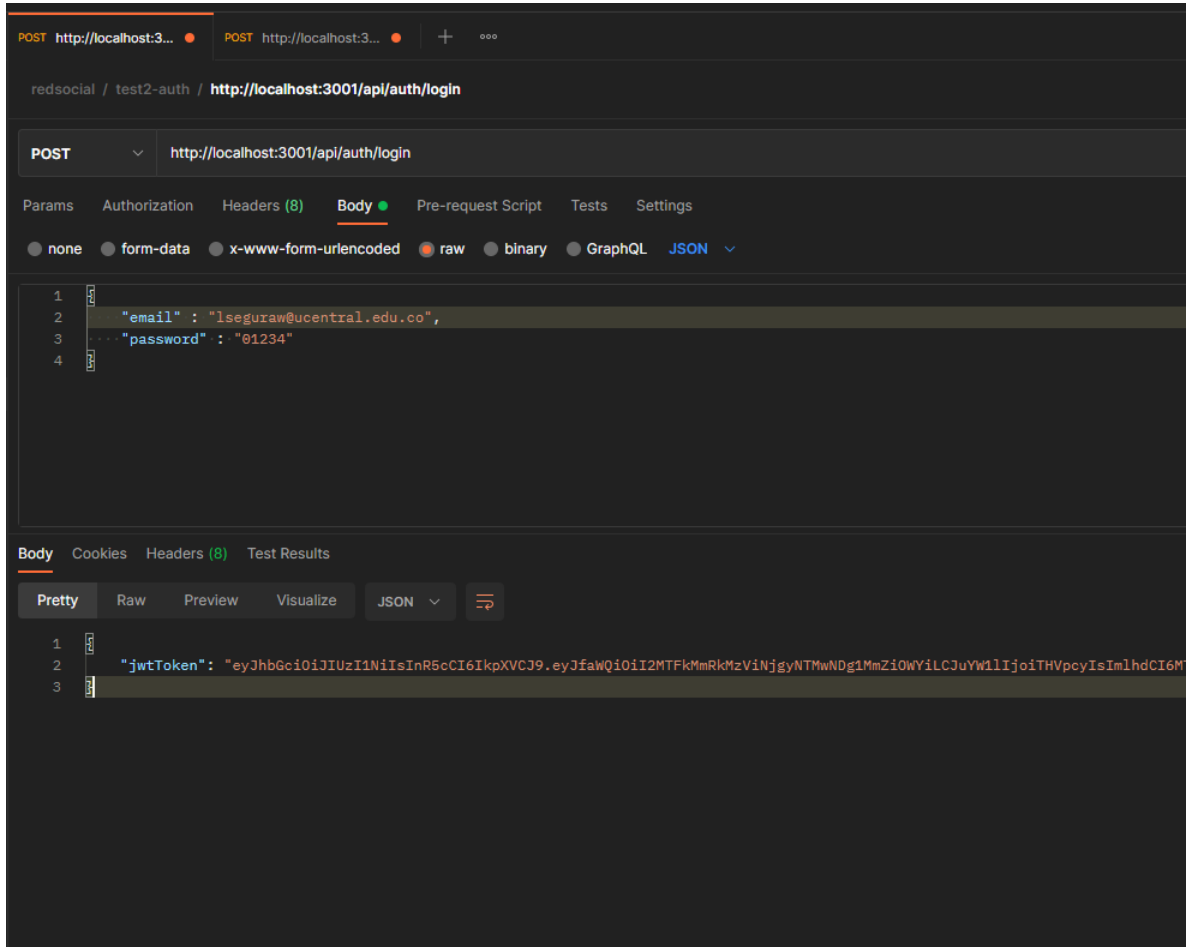
The screenshot displays a REST client interface with a dark theme. At the top, there are two tabs for POST requests to `http://localhost:3001/api/post/saveTask` and `http://localhost:3001/api/post/savePost`. The second tab is active, showing the request details. The request method is POST, and the URL is `http://localhost:3001/api/post/savePost`. The request body is set to JSON and contains the following data:

```
{  "text": "no me gusto game of thrones",  "hashtag": "#seriepaila"}
```

Below the request details, the response is shown in the 'Body' tab. The response is a JSON object with the following structure:

```
{  "result": {    "_id": "611d31eb5b6825304852fbc6",    "userId": "611d2da45b6825304852fb9c",    "text": "no me gusto game of thrones",    "hashtag": "#seriepaila",    "date": "2021-08-18T16:14:35.198Z",    "__v": 0  }}
```

### 3) Login con Luis



## Post con Luis

a)

The screenshot displays a REST client interface with a POST request to `http://localhost:3001/api/post/savePost`. The request body is a JSON object with the following structure:

```
{  "text": "como es pa farrear hasta el amanecer",  "hashtag": "#lescopeo"}
```

The response body is a JSON object with the following structure:

```
{  "result": {    "_id": "611d32625b6825304852fbca",    "userId": "611d2dd35b6825304852fb9f",    "text": "como es pa farrear hasta el amanecer",    "hashtag": "#lescopeo",    "date": "2021-08-18T16:16:34.712Z",    "__v": 0  }}
```

b)

The screenshot displays a REST client interface with a dark theme. At the top, there are two tabs for POST requests to `http://localhost:3001/api/post/saveTask` and `http://localhost:3001/api/post/savePost`. The second tab is active. Below the tabs, the URL bar shows `http://localhost:3001/api/post/savePost`. The interface includes tabs for Params, Authorization, Headers (9), Body, Pre-request Script, Tests, and Settings. The Body tab is selected, and the content type is set to JSON. The request body is a JSON object with the following structure:

```
{  "text": "Miercoles de portarse bien :(",  "hashtag": "#miercoles"}
```

Below the request body, the response section is visible, showing the Body tab selected. The response is a JSON object with the following structure:

```
{  "result": {    "_id": "611d327d5b6825304852fbcd",    "userId": "611d2dd35b6825304852fb9f",    "text": "Miercoles de portarse bien :(",    "hashtag": "#miercoles",    "date": "2021-08-18T16:17:01.613Z",    "__v": 0  }}
```

c)

The screenshot displays a REST client interface with a dark theme. At the top, there are two tabs for POST requests to `http://localhost:3001/api/post/saveTask` and `http://localhost:3001/api/post/savePost`. The second tab is active. Below the tabs, the URL `http://localhost:3001/api/post/savePost` is shown. The interface includes tabs for Params, Authorization, Headers (9), Body, Pre-request Script, Tests, and Settings. The Body tab is selected, and the request body is set to raw JSON. The request body contains the following JSON:

```
{  "text": "Me estan estorbando 80k hableme",  "hashtag": "#caigan"}
```

Below the request body, the response body is shown in the 'Body' tab. The response is a JSON object with the following structure:

```
{  "result": {    "_id": "611d329b5b6825304852fbd0",    "userId": "611d2dd35b6825304852fb9f",    "text": "Me estan estorbando 80k hableme",    "hashtag": "#caigan",    "date": "2021-08-18T16:17:31.276Z",    "__v": 0  }}
```

d)

The screenshot shows a REST client interface with a POST request to `http://localhost:3001/api/post/saveTask`. The request body is a JSON object with `text` and `hashtag` fields. The response is a JSON object with a `result` field containing an object with `_id`, `userId`, `text`, `hashtag`, `date`, and `__v` properties.

POST `http://localhost:3001/api/post/saveTask`

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings

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

```
1 {
2   "text": "hoy un accidente que visaje la vida mi so",
3   "hashtag": "#visaje"
4 }
```

**Body** Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "result": {
3     "_id": "611d32c05b6825304852fbd3",
4     "userId": "611d2dd35b6825304852fb9f",
5     "text": "hoy un accidente que visaje la vida mi so",
6     "hashtag": "#visaje",
7     "date": "2021-08-18T16:18:08.220Z",
8     "__v": 0
9   }
10 }
```

e)

The screenshot shows a REST client interface with a POST request to `http://localhost:3001/api/post/savePost`. The request body is a JSON object with `text` and `hashtag` fields. The response body is a JSON object with a `result` field containing an object with `_id`, `userId`, `text`, `hashtag`, `date`, and `__v` fields.

POST `http://localhost:3001/api/post/savePost`

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings

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

```
1 {
2   ... "text": "como se llama el meme del pikachu haciendo :0 ?",
3   ... "hashtag": "#pikachu"
4 }
```

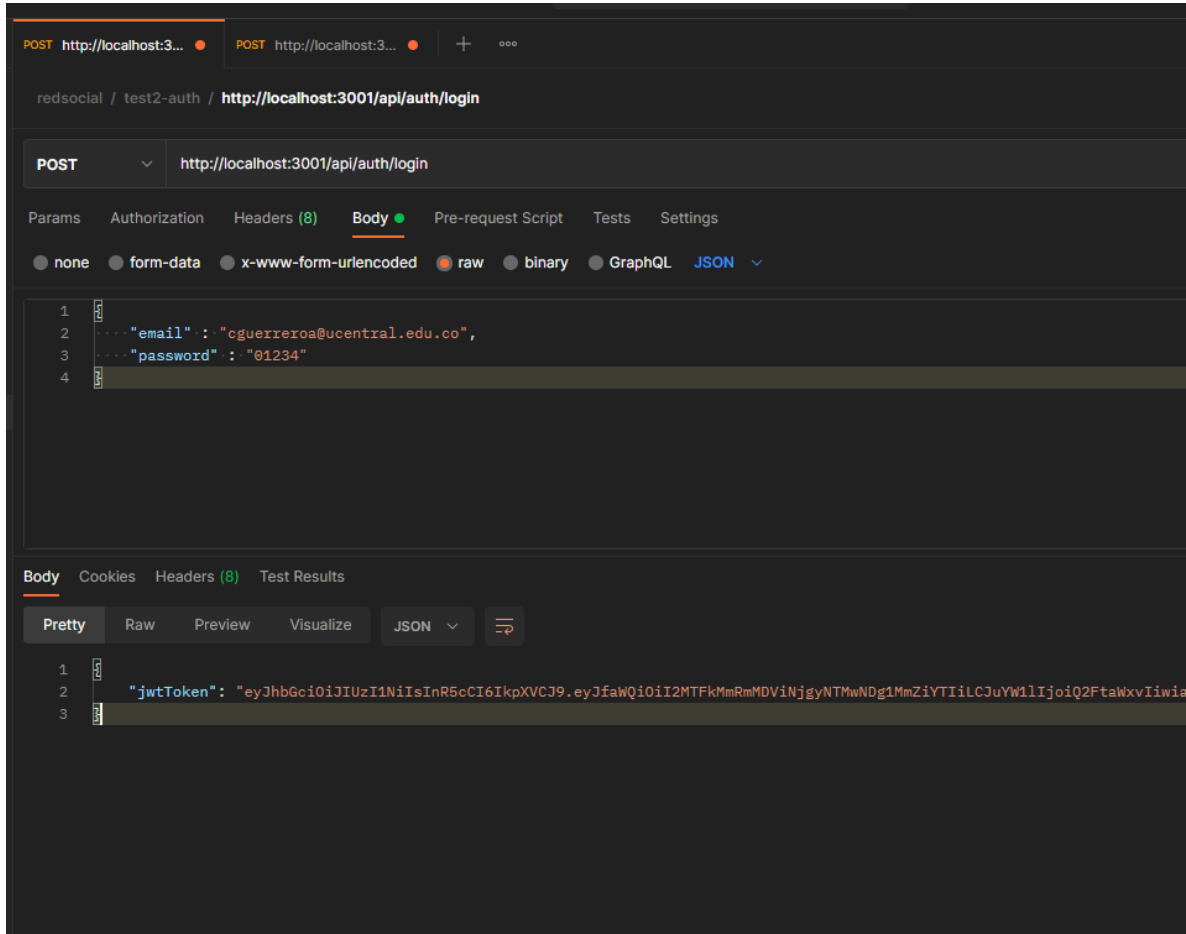
**Body** Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON ▾ ↻

```
1 {
2   "result": {
3     "_id": "611d330d5b6825304852fbd6",
4     "userId": "611d2dd35b6825304852fb9f",
5     "text": "como se llama el meme del pikachu haciendo :0 ?",
6     "hashtag": "#pikachu",
7     "date": "2021-08-18T16:19:25.364Z",
8     "__v": 0
9   }
10 }
```



#### 4) Login de Camilo



## Post de Camilo

a)

The screenshot displays a REST client interface with a POST request to `http://localhost:3001/api/post/savePost`. The request body is a JSON object with the following structure:

```
{  "text": "roten canciones sad pls :(",  "hashtag": "#sad"}
```

The response body, shown in 'Pretty' view, is a JSON object representing the saved post:

```
{  "result": {    "_id": "611d33785b6825304852fbda",    "userId": "611d2df05b6825304852fba2",    "text": "roten canciones sad pls :(",    "hashtag": "#sad",    "date": "2021-08-18T16:21:12.817Z",    "__v": 0  }}
```

b)

The screenshot shows a REST client interface with a POST request to `http://localhost:3001/api/post/savePost`. The request body is a JSON object with `text` and `hashtag` fields. The response body is a JSON object with a `result` field containing an object with `_id`, `userId`, `text`, `hashtag`, `date`, and `__v` fields.

POST `http://localhost:3001/api/post/savePost`

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings

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

```
1 {
2   "text": "barca o real, matense en los comentarios xd",
3   "hashtag": "#pelea"
4 }
```

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "result": {
3     "_id": "611d339b5b6825304852fbdd",
4     "userId": "611d2df05b6825304852fba2",
5     "text": "barca o real, matense en los comentarios xd",
6     "hashtag": "#pelea",
7     "date": "2021-08-18T16:21:47.231Z",
8     "__v": 0
9   }
10 }
```

c)

The screenshot displays a REST client interface with a POST request to `http://localhost:3001/api/post/savePost`. The request body is a JSON object with `text` and `hashtag` fields. The response body shows a successful result with fields like `_id`, `userId`, `text`, `hashtag`, `date`, and `__v`.

**Request:**

```
POST http://localhost:3001/api/post/savePost
```

**Body:**

```
{  "text": "legalicen muchas cosas jashjashajsh",  "hashtag": "#ligal"}
```

**Response:**

```
{  "result": {    "_id": "611d33e15b6825304852fbe0",    "userId": "611d2df05b6825304852fba2",    "text": "legalicen muchas cosas jashjashajsh",    "hashtag": "#ligal",    "date": "2021-08-18T16:22:57.597Z",    "__v": 0  }}
```

d)

The screenshot displays a REST client interface with a POST request to `http://localhost:3001/api/post/savePost`. The request body is a JSON object with `text` and `hashtag` fields. The response body shows a successful result with a `_id`, `userId`, `text`, `hashtag`, `date`, and `__v` field.

POST `http://localhost:3001/api/post/savePost`

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings

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

```
1 {
2   "text": "Señores me pusieron correa F",
3   "hashtag": "#novia"
4 }
```

**Body** Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "result": {
3     "_id": "611d33fc5b6825304852fbc3",
4     "userId": "611d2df05b6825304852fba2",
5     "text": "Señores me pusieron correa F",
6     "hashtag": "#novia",
7     "date": "2021-08-18T16:23:24.049Z",
8     "__v": 0
9   }
10 }
```

e)

The screenshot shows a REST client interface with a POST request to `http://localhost:3001/api/post/savePost`. The request body is a JSON object with `text` and `hashtag` fields. The response body is a JSON object with a `result` field containing an object with `_id`, `userId`, `text`, `hashtag`, `date`, and `__v` fields.

POST `http://localhost:3001/api/post/savePost`

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings

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

```
1 {
2   "text": "Ponganme vacuna pero para olvidarla a ella",
3   "hashtag": "#nopuedeser"
4 }
```

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "result": {
3     "_id": "611d341d5b6825304852fbe6",
4     "userId": "611d2df05b6825304852fba2",
5     "text": "Ponganme vacuna pero para olvidarla a ella",
6     "hashtag": "#nopuedeser",
7     "date": "2021-08-18T16:23:57.112Z",
8     "__v": 0
9   }
10 }
```

## 5) Login de Joel

The screenshot shows a REST client interface with a POST request to `http://localhost:3001/api/auth/login`. The request body is a JSON object containing email and password. The response body is a JSON object containing a JWT token.

**Request:**

```
POST http://localhost:3001/api/auth/login
```

**Body:**

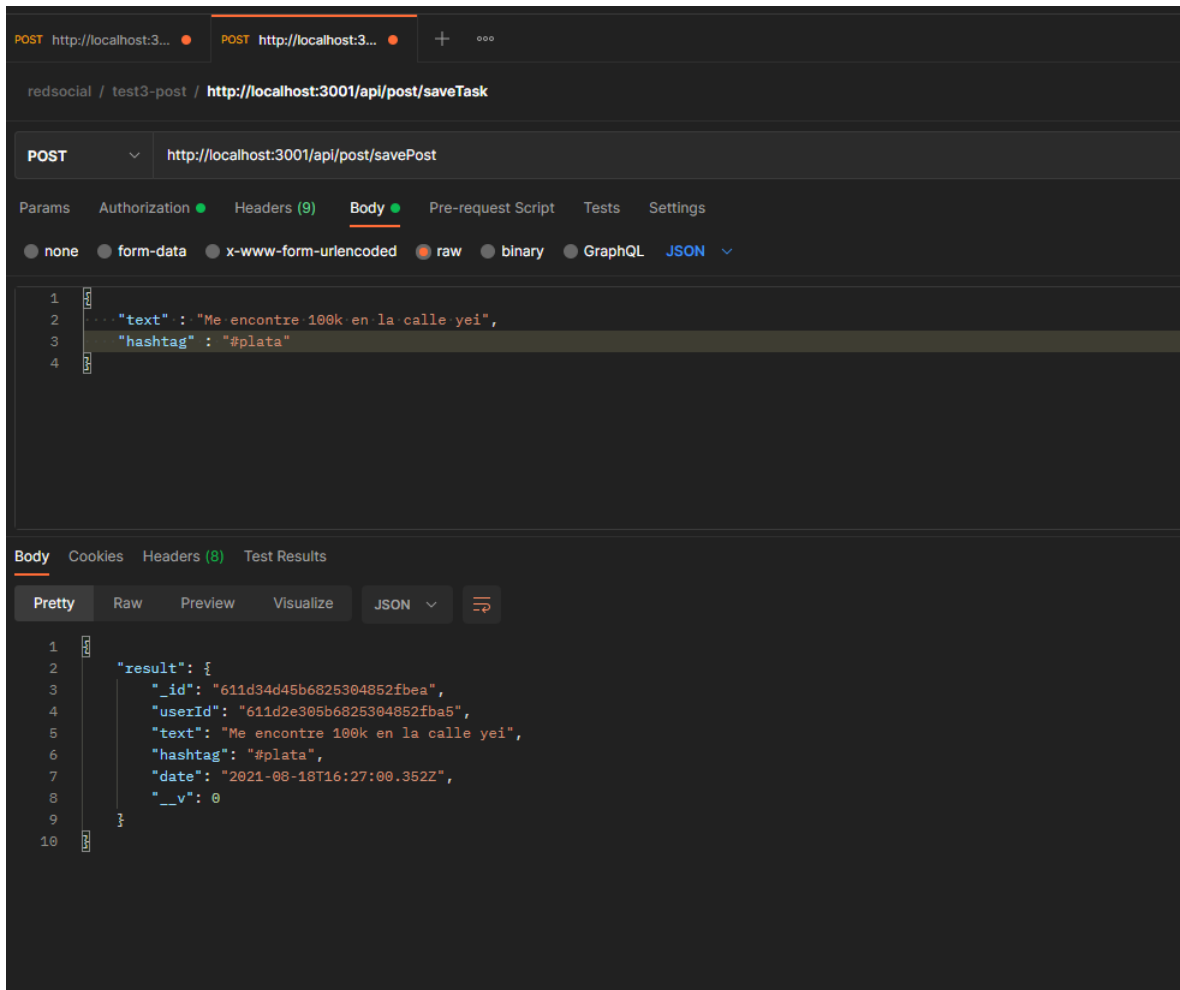
```
{  "email": "jarevalos@ucentral.edu.co",  "password": "01234"}
```

**Response:**

```
{  "jwtToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiI2MTFkMmUzMjYNTmVjgyNTMwNDg1MmZiYTUuLCJyZW11IjoiaS91b21234"}
```

## Post de Joel

a)





b)

The screenshot displays a REST client interface with a POST request to `http://localhost:3001/api/post/saveTask`. The request body is a JSON object with `text` and `hashtag` fields. The response is also in JSON format, showing a `result` object with various fields including `_id`, `userId`, `text`, `hashtag`, `date`, and `__v`.

**Request:**

```
POST http://localhost:3001/api/post/saveTask
```

**Body:**

```
{  "text": "Ya se programar hackeo facebook",  "hashtag": "#emprendedor"}
```

**Response:**

```
{  "result": {    "_id": "611d34f75b6825304852fbed",    "userId": "611d2e305b6825304852fba5",    "text": "Ya se programar hackeo facebook",    "hashtag": "#emprendedor",    "date": "2021-08-18T16:27:35.829Z",    "__v": 0  }}
```

c)

The screenshot shows a REST client interface with a POST request to `http://localhost:3001/api/post/savePost`. The request body is a JSON object with `text` and `hashtag` fields. The response is also a JSON object with a `result` field containing an object with `_id`, `userId`, `text`, `hashtag`, `date`, and `__v` fields.

POST http://localhost:3001/api/post/savePost

Params Authorization Headers (9) Body Pre-request Script Tests Settings

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

```
1 {
2   "text": "a 30k les formateo pc",
3   "hashtag": "#llameme"
4 }
```

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "result": {
3     "_id": "611d35145b6825304852fbf0",
4     "userId": "611d2e305b6825304852fba5",
5     "text": "a 30k les formateo pc",
6     "hashtag": "#llameme",
7     "date": "2021-08-18T16:28:04.079Z",
8     "__v": 0
9   }
10 }
```

d)

The screenshot shows a REST client interface with a POST request to `http://localhost:3001/api/post/savePost`. The request body is a JSON object with `text` and `hashtag` fields. The response is a JSON object with a `result` field containing an object with `_id`, `userId`, `text`, `hashtag`, `date`, and `__v` fields.

POST `http://localhost:3001/api/post/savePost`

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings

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

```
1 {
2   "text": "estoy que me bebo la fuente de paris",
3   "hashtag": "#trago"
4 }
```

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "result": {
3     "_id": "611d352c5b6825304852fbf3",
4     "userId": "611d2e305b6825304852fba5",
5     "text": "estoy que me bebo la fuente de paris",
6     "hashtag": "#trago",
7     "date": "2021-08-18T16:28:28.137Z",
8     "__v": 0
9   }
10 }
```

e)

The screenshot shows a REST client interface with a dark theme. At the top, there are tabs for different requests, with the active one being a POST to `http://localhost:3001/api/post/savePost`. Below the tabs, the 'Body' tab is selected, showing a JSON request body: `{ "text": "parce quiero escuchar rock pasen plays", "hashtag": "#play" }`. The bottom section shows the 'Test Results' tab, displaying the JSON response: `{ "result": { "_id": "611d3555b6825304852fbf6", "userId": "611d2e305b6825304852fba5", "text": "parce quiero escuchar rock pasen plays", "hashtag": "#play", "date": "2021-08-18T16:29:09.846Z", "__v": 0 } } }`.

POST http://localhost:3001/api/post/savePost

Params Authorization Headers (9) Body Pre-request Script Tests Settings

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

```
1 {
2   "text": "parce quiero escuchar rock pasen plays",
3   "hashtag": "#play"
4 }
```

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "result": {
3     "_id": "611d3555b6825304852fbf6",
4     "userId": "611d2e305b6825304852fba5",
5     "text": "parce quiero escuchar rock pasen plays",
6     "hashtag": "#play",
7     "date": "2021-08-18T16:29:09.846Z",
8     "__v": 0
9   }
10 }
```