

# Smart Wait:3000

## Contenido

Análisis informal: .....	1
Problema: .....	1
Solución: .....	1
Uso: .....	1
Análisis detallado: .....	2
Casos de uso: .....	2
Usuario Registrado: .....	2
Cualquier Usuario: .....	2
Admins: .....	3
Diseño de la Base de Datos: .....	4
tabla Servicios: .....	4
tabla Tickets: .....	4
tabla Usuarios: .....	4
API: Endpoints: .....	5
Servicios: .....	5
Tickets: .....	9
Usuarios: .....	11

## Análisis informal:

**Problema:** El problema de tener que esperar en una cola durante largos períodos de tiempo puede ser frustrante y una pérdida de tiempo.

**Solución:** Este problema podría tener una solución que sería una aplicación que permita a los usuarios obtener un ticket de forma virtual, lo que les permitiría hacer otras cosas mientras esperan su turno. La aplicación también mostrará el turno en el que se está atendiendo a los usuarios y el servicio para el que se han registrado, lo que facilitaría la organización y mejoraría la experiencia del usuario. De esta manera, se podría evitar la molestia de tener que esperar en una cola física y aprovechar el tiempo de manera más eficiente.

**Uso:** Cuando el usuario haga clic en el botón, se le proporcionará un ticket que incluirá el nombre del servicio y su lugar en la lista de espera. Además, se le enviará un mensaje cuando solo queden tres personas por delante de él en la cola.

## Análisis detallado:

Información sobre **servicio**:

- Nombre del servicio.
- Identificador del servicio.
- Número del siguiente turno.

Información para cada **ticket**:

- Identificador del ticket.
- Identificador del servicio al que se refiere el ticket.
- Identificador del usuario que realizó la solicitud del ticket.
- Número de ticket.

Información sobre **usuarios**:

- Nombre del usuario.
- Contraseña del usuario.
- Identificador del usuario.
- Indicación si el usuario es administrador o no. (true/false)

## Casos de uso:

Usuario Registrado:

- Solicitar varios tickets de forma fácil y rápida.
- Consultar su posición en la lista de espera en cualquier momento.
- Recibir notificación cuando queden solo 3 personas por delante suya.

Cualquier Usuario:

- Solicitar un ticket de forma fácil y rápida.
- Consultar su posición en la lista de espera en cualquier momento.

## Admins:

- Realizar operaciones CRUD (crear, leer, actualizar y eliminar) en todos los aspectos de la aplicación, incluyendo tickets y usuarios.

## Diseño de la Base de Datos:

### tabla Servicios:

La tabla Servicios almacena información sobre los diferentes servicios que ofrece la empresa, incluyendo (id, nombre, numero) numero -> del atendido actual

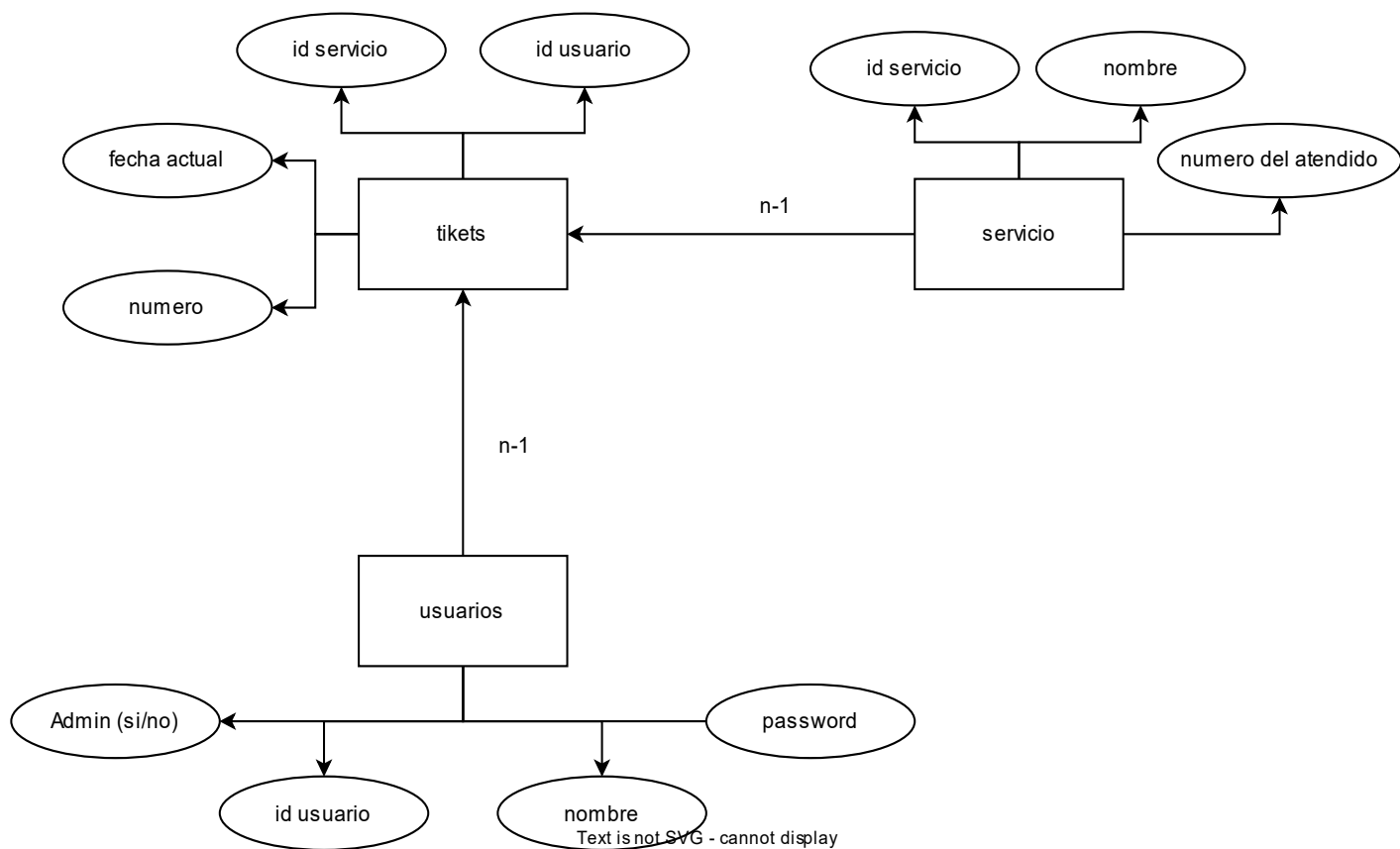
### tabla Tickets:

La tabla Tickets contiene la información de los tickets generados por los usuarios, incluyendo (id, id\_usuario (FK), id\_servicio (FK), fecha actual, numero)

### tabla Usuarios:

La tabla Usuarios almacena la información de los usuarios registrados en la aplicación, incluyendo (id, nombre, contraseña, admin (Boolean) )

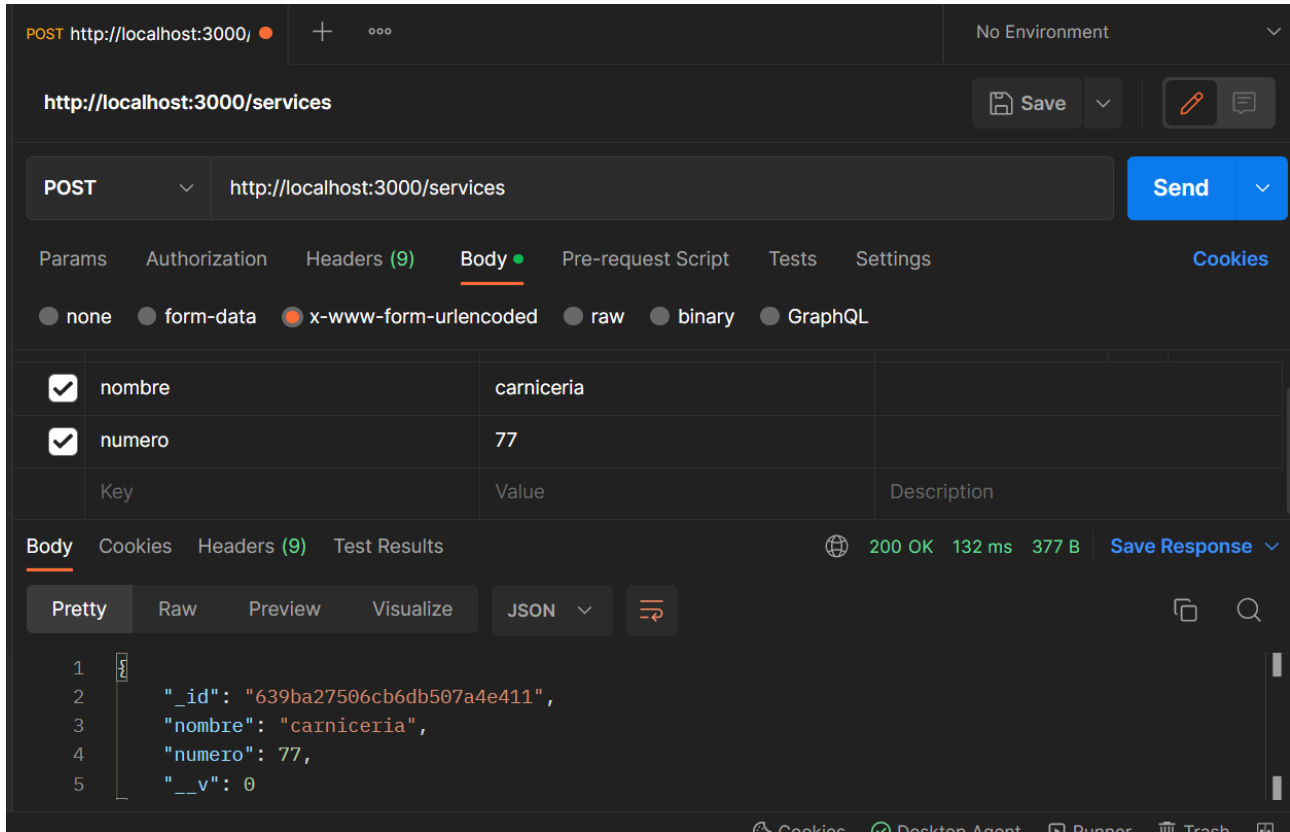
Modelo relacional:



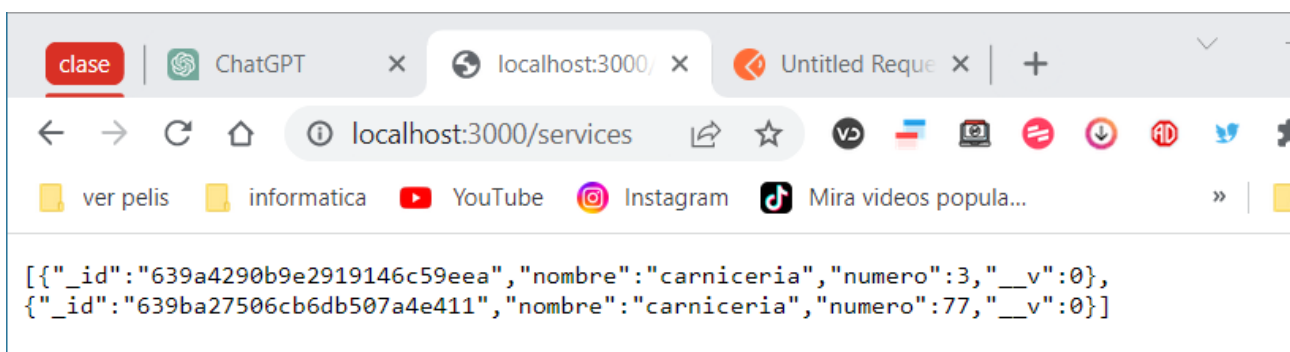
# API: Endpoints:

Servicios:

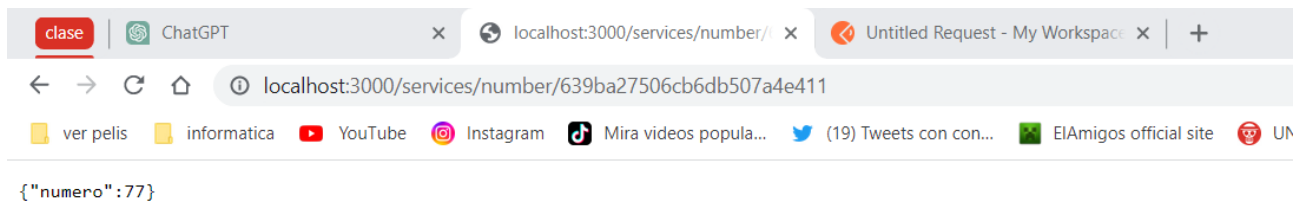
POST: api/services -> endpoint para añadir servicios.



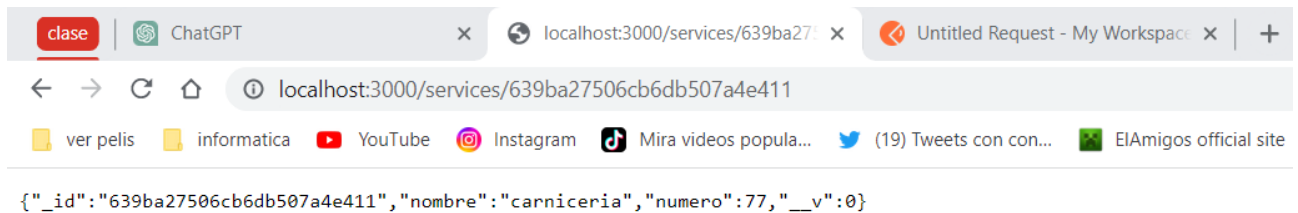
GET: api/services -> endpoint para mostrar todos los servicios y sus datos.



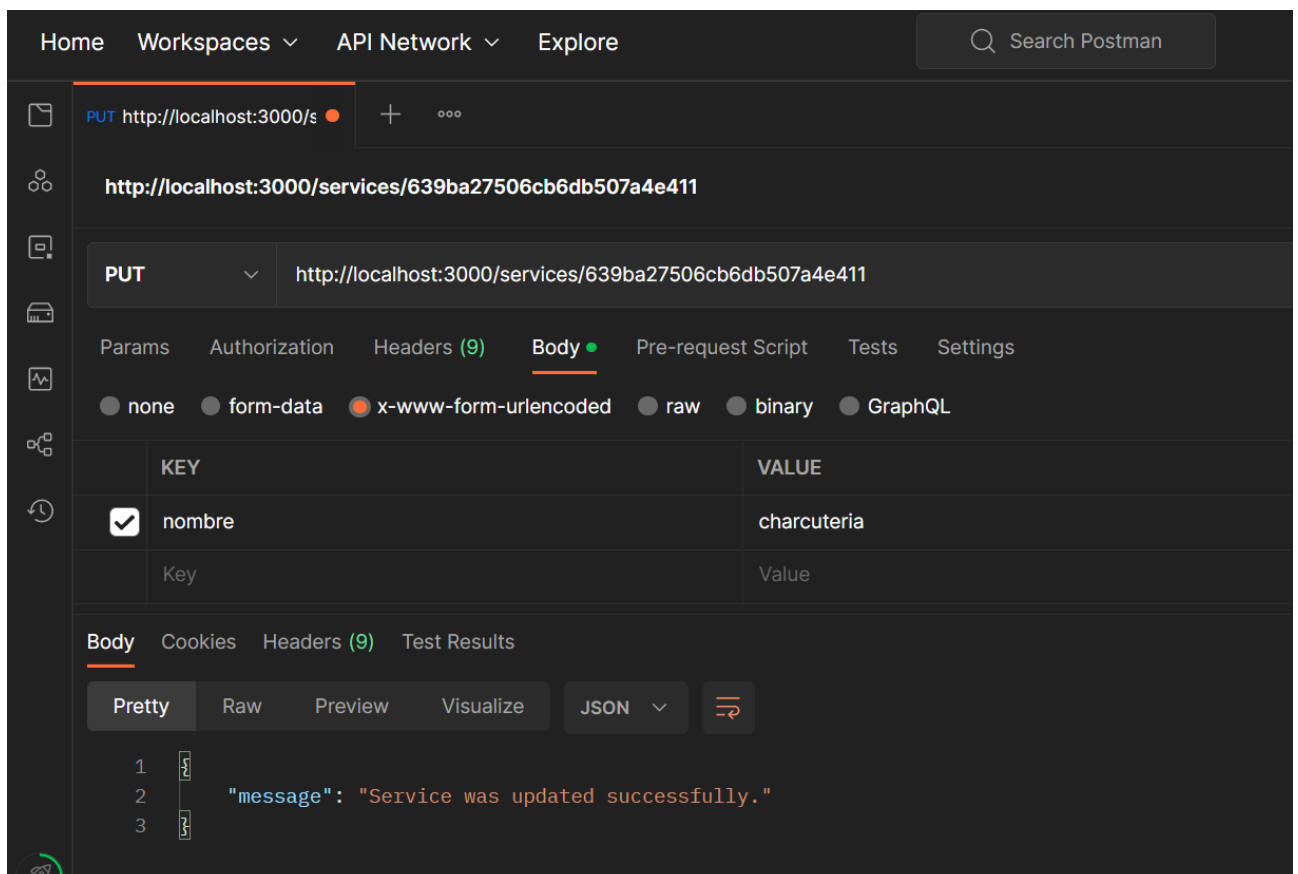
GET: api/services/number/:id -> endpoint para mostrar el numero del atendido actual de un servicio específico.

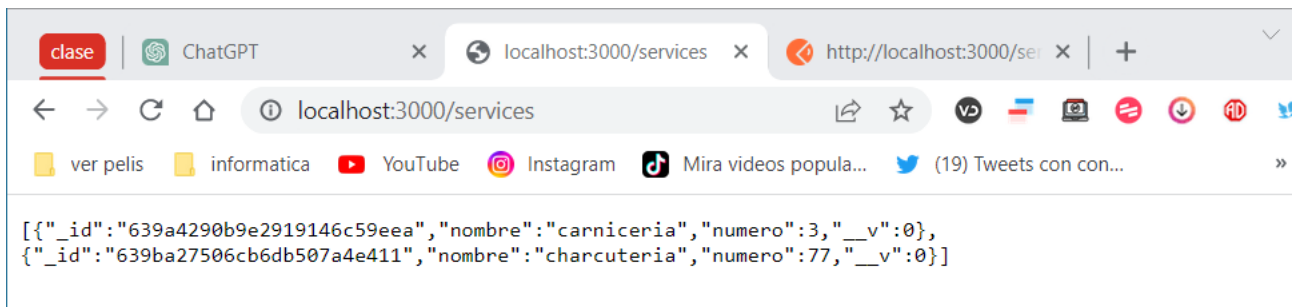


GET: api/services/:id -> endpoint para mostrar el un servicio específico y sus datos.

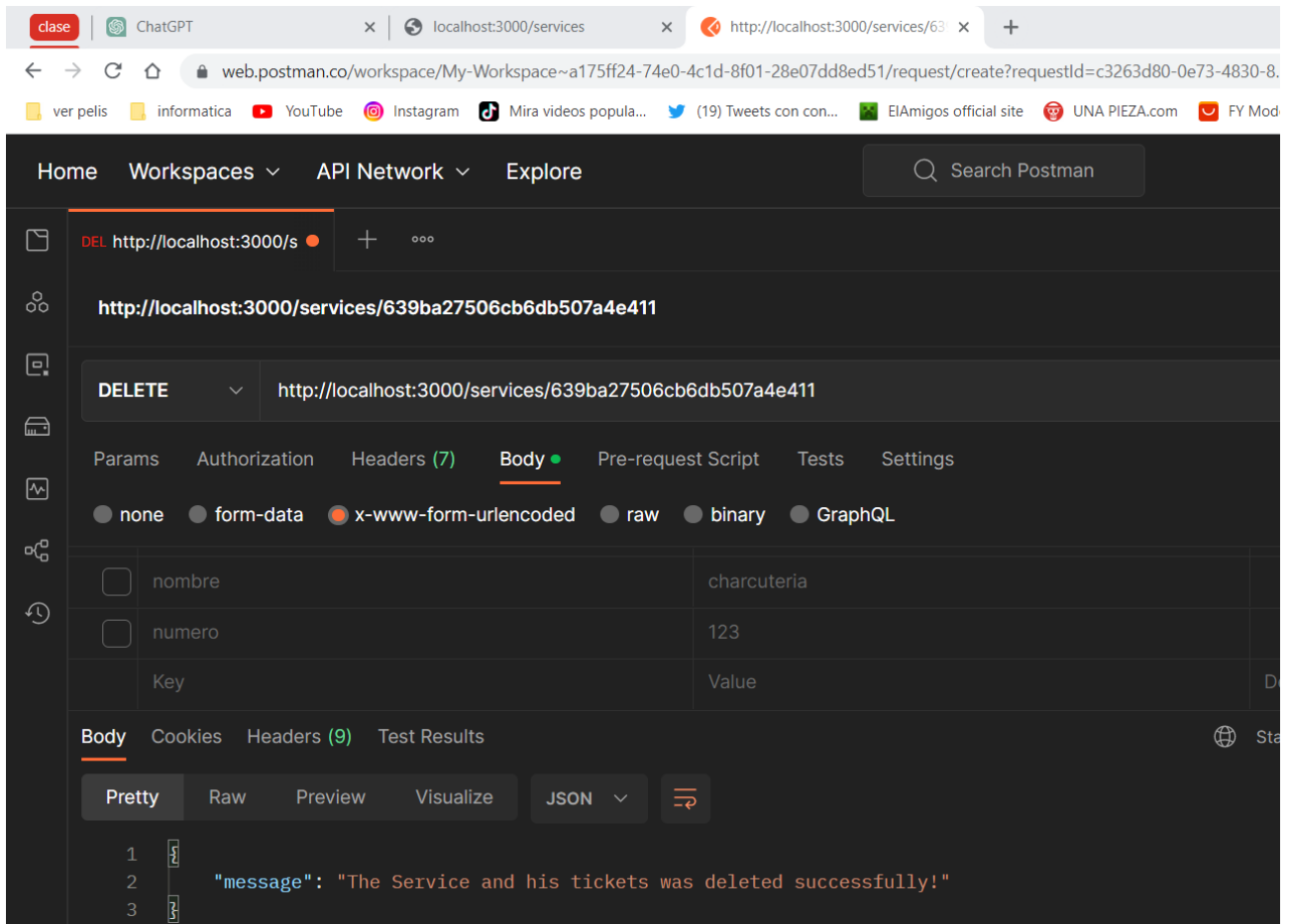


PUT: api/services -> endpoint para actualizar un servicio.





DELETE: api/services/:id -> endpoint para borrar un servicio.



DELETE: api/services -> endpoint para borrar todos los servicios.

HomeWorkspacesAPI NetworkExplore

Search Postman

Invite

Upgrade

DELhttp://localhost:3000/s

+

...

No Environment

http://localhost:3000/services

Save

</>

DELETE

http://localhost:3000/services

Send

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Settings

Cookies

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

nombre

correos

numero

123

Key

Value

Description

Body

Cookies

Headers (9)

Test Results

Status: 200 OK

Time: 102 ms

Size: 351 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

1

2

3

"message": "4 Service were deleted successfully!"



## Tickets:

POST: api/tickets -> endpoint para añadir tickets.

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

- Method:** POST
- URL:** http://localhost:3000/tickets
- Body:** x-www-form-urlencoded
- Params Table:**

KEY	VALUE	DESCRIPTION
numero	639	
id_servicio	639be025a2040c99fb824f23	
id_usuario	639bc95c3fc423fb54d00963	
- Status:** 200 OK
- Time:** 107 ms
- Size:** 516 B
- Body Content (JSON):**

```
{  "numero": 639,  "id_servicio": "639be025a2040c99fb824f23",  "id_usuario": "639bc95c3fc423fb54d00963",  "_id": "639be053a2040c99fb824f27",  "createdAt": "2022-12-16T03:04:51.949Z",  "updatedAt": "2022-12-16T03:04:51.949Z",  "__v": 0}
```

GET: api/tickets -> endpoint para mostrar todos los tickets y sus datos.

The screenshot shows a web browser displaying the JSON response for the GET request to http://localhost:3000/tickets:

```
[{"_id": "639be053a2040c99fb824f27", "numero": 639, "id_servicio": "639be025a2040c99fb824f23", "id_usuario": "639bc95c3fc423fb54d00963", "createdAt": "2022-12-16T03:04:51.949Z", "updatedAt": "2022-12-16T03:04:51.949Z", "__v": 0}]
```

GET: api/tickets/:id -> endpoint para mostrar un ticket específico y el servicio e usuario al que pertenece.

The screenshot shows a web browser displaying the JSON response for the GET request to http://localhost:3000/tickets/639be053a2040c99fb824f27:

```
{ "message": "Tu numero es el 639 para el servicio: charcuteria Fecha actual: Fri Dec 16 2022 03:04:51 GMT+0000 (hora estándar de Europa occidental) perteneciente a pascal" }
```

PUT: api/tickets -> endpoint para actualizar un tickets.

Postman interface showing a PUT request to `http://localhost:3000/tickets/639be053a2040c99fb824f27`. The request body is in x-www-form-urlencoded format with the following data:

KEY	VALUE	DESCRIPTION
numero	639	
id_servicio	639be025a2040c99fb824f23	
id_usuario	639bc9863fc423fb54d00967	

The response status is 200 OK, Time: 106 ms, Size: 347 B. The response body is:

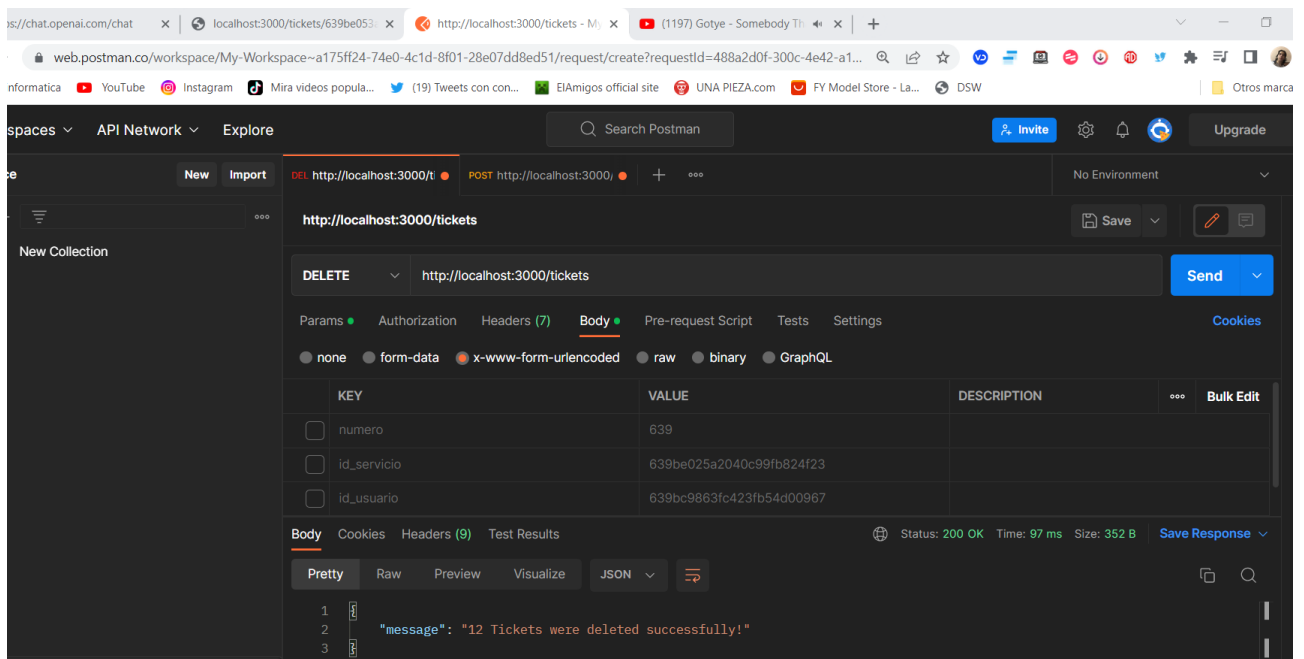
```
{
  "message": "Ticket was updated successfully."
}
```

Browser screenshot showing the response message: `{"message": "Tu numero es el 639 para el servicio: charcuteria Fecha actual: Fri Dec 16 2022 03:04:51 GMT+0000 (hora estándar de Europa occidental) perteneciente a manolo"}`.

DELETE: `api/tickets/:id` -> endpoint para borrar un ticket.

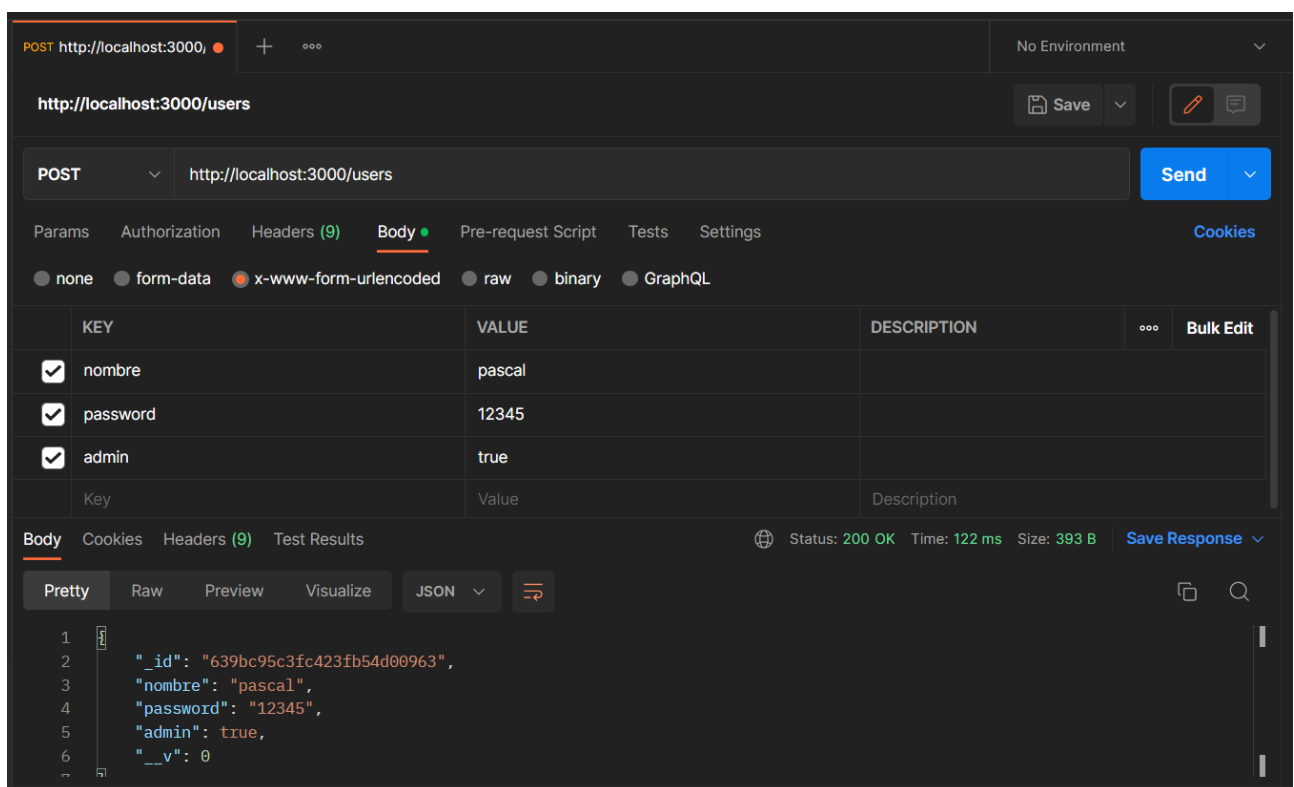
Browser screenshot showing the response message: `{"message": "Tu numero es el 639 para el servicio: charcuteria Fecha actual: Fri Dec 16 2022 03:04:51 GMT+0000 (hora estándar de Europa occidental) perteneciente a manolo"}`.

DELETE: `api/tickets` -> endpoint para borrar todos los tickets.

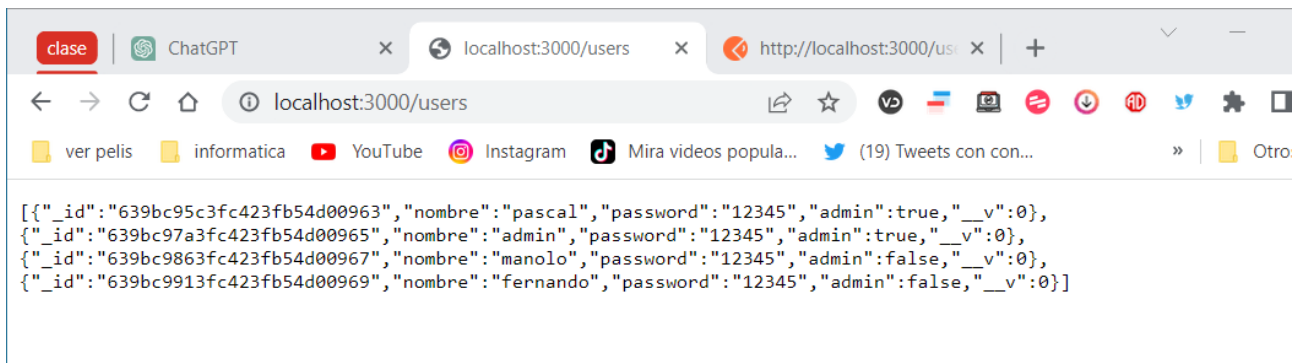


## Usuarios:

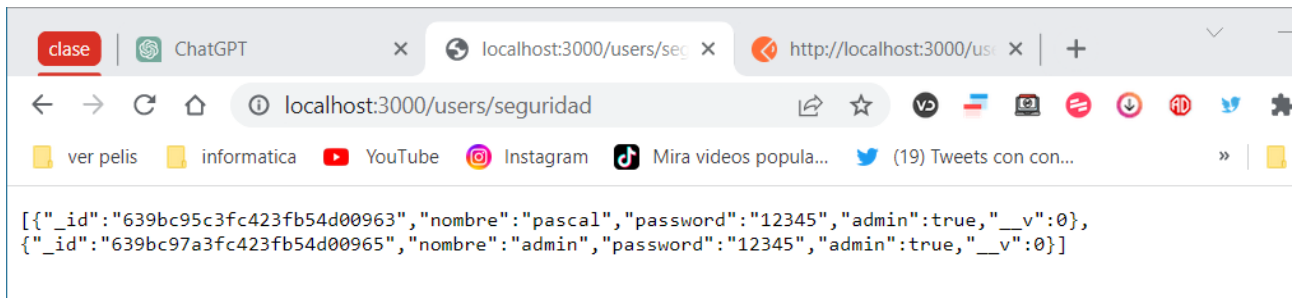
POST: `api/users` -> endpoint para añadir usuarios.



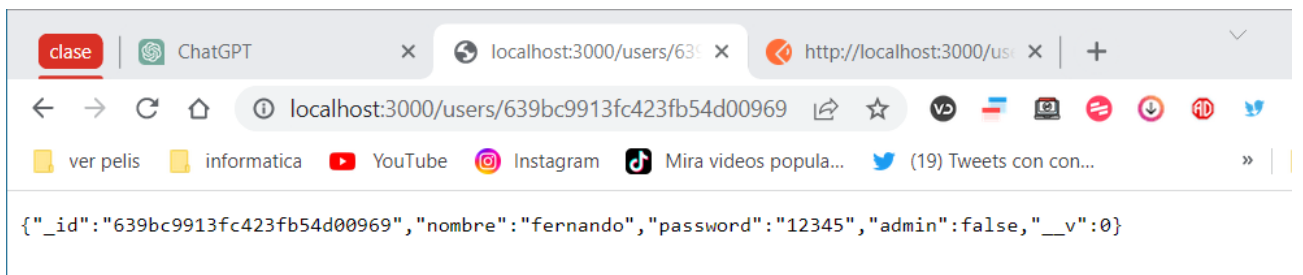
GET: `api/users` -> endpoint para mostrar todos los usuarios y sus datos.



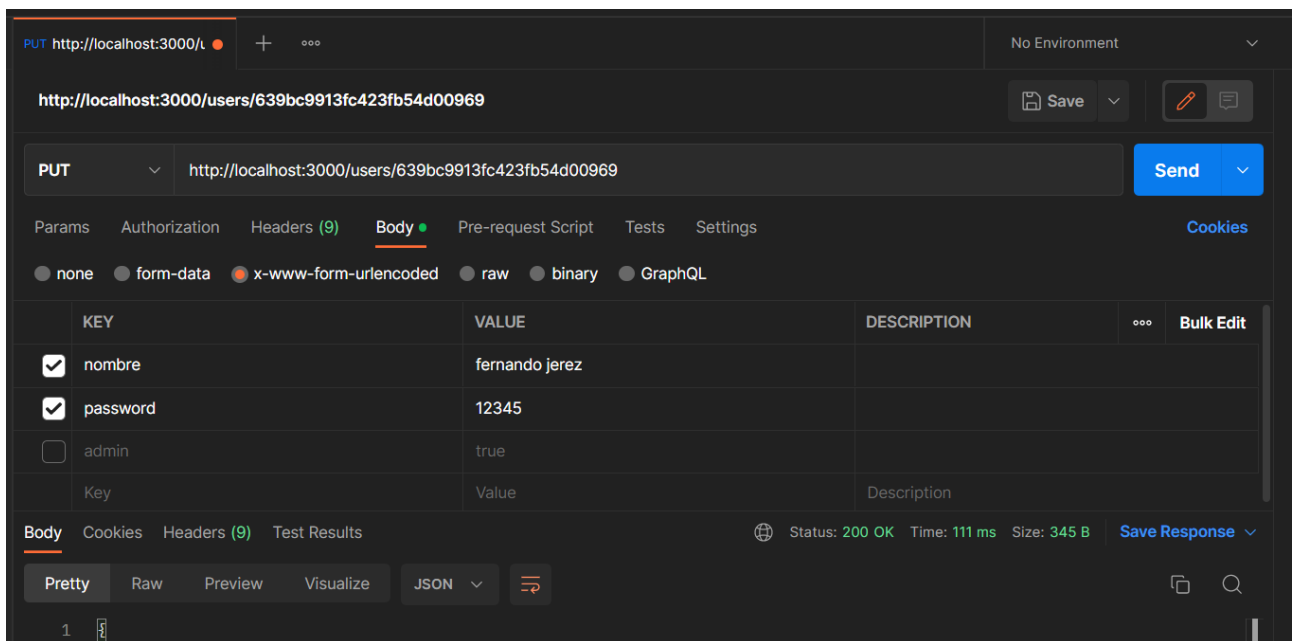
GET: api/users/seguridad -> endpoint para mostrar todos los usuarios Administradores.



GET: api/users/:id -> endpoint para mostrar el un usuario específico.



PUT: api/users -> endpoint para actualizar un usuario.



localhost:3000/users/639bc9913fc423fb54d00969

```
{ "_id": "639bc9913fc423fb54d00969", "nombre": "fernando jerez", "password": "12345", "admin": false, "__v": 0 }
```

DELETE: api/users/:id -> endpoint para borrar un usuario.

DEL http://localhost:3000/u

http://localhost:3000/users/639bc9913fc423fb54d00969

DELETE http://localhost:3000/users/639bc9913fc423fb54d00969

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

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

	KEY	VALUE	DESCRIPTION
<input type="checkbox"/>	nombre	fernando jerez	
<input type="checkbox"/>	password	12345	
<input type="checkbox"/>	admin	true	
	Key	Value	Description

Body Cookies Headers (9) Test Results Status: 200 OK Time: 188 ms Size: 365 B Save Response

Pretty Raw Preview Visualize JSON

```
1  
2 "message": "The User and his tickets was deleted successfully!"  
3
```

localhost:3000/users

```
[{"_id": "639bc95c3fc423fb54d00963", "nombre": "pascal", "password": "12345", "admin": true, "__v": 0}, {"_id": "639bc97a3fc423fb54d00965", "nombre": "admin", "password": "12345", "admin": true, "__v": 0}, {"_id": "639bc9863fc423fb54d00967", "nombre": "manolo", "password": "12345", "admin": false, "__v": 0}]
```