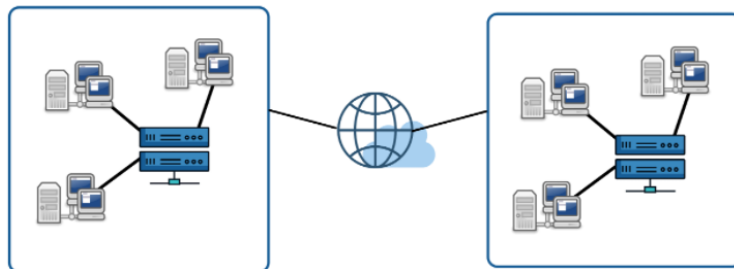


# Sistemas Distribuidos

## Práctica 2 - "Fun with Queues"

2021/22

- ELVI MIHAI SABAU SABAU - 51254875L
- Iván Sabater Domínguez - 48797056A
- 



## Índice.

Tecnologías usadas.	2
Estructura del Sistema.	3
Dependencias.	3
Script.	4
Registry.	4
Visitor.	5
Servidor web.	5
Controlador Kafka.	5
Navegador web.	6
Sensor.	6
Waiting Time Server.	7
Engine.	7
Guía de Despliegue.	8
Si tenemos las imágenes en un .tar	8
No tenemos las imágenes, tenemos que generarlas.	8
Engine.	9
Registry.	10
Sensor.	11
Visitor.	12
Waiting Time Server.	13
Orden de lanzamiento.	13
Capturas del funcionamiento.	14

# Tecnologías usadas.

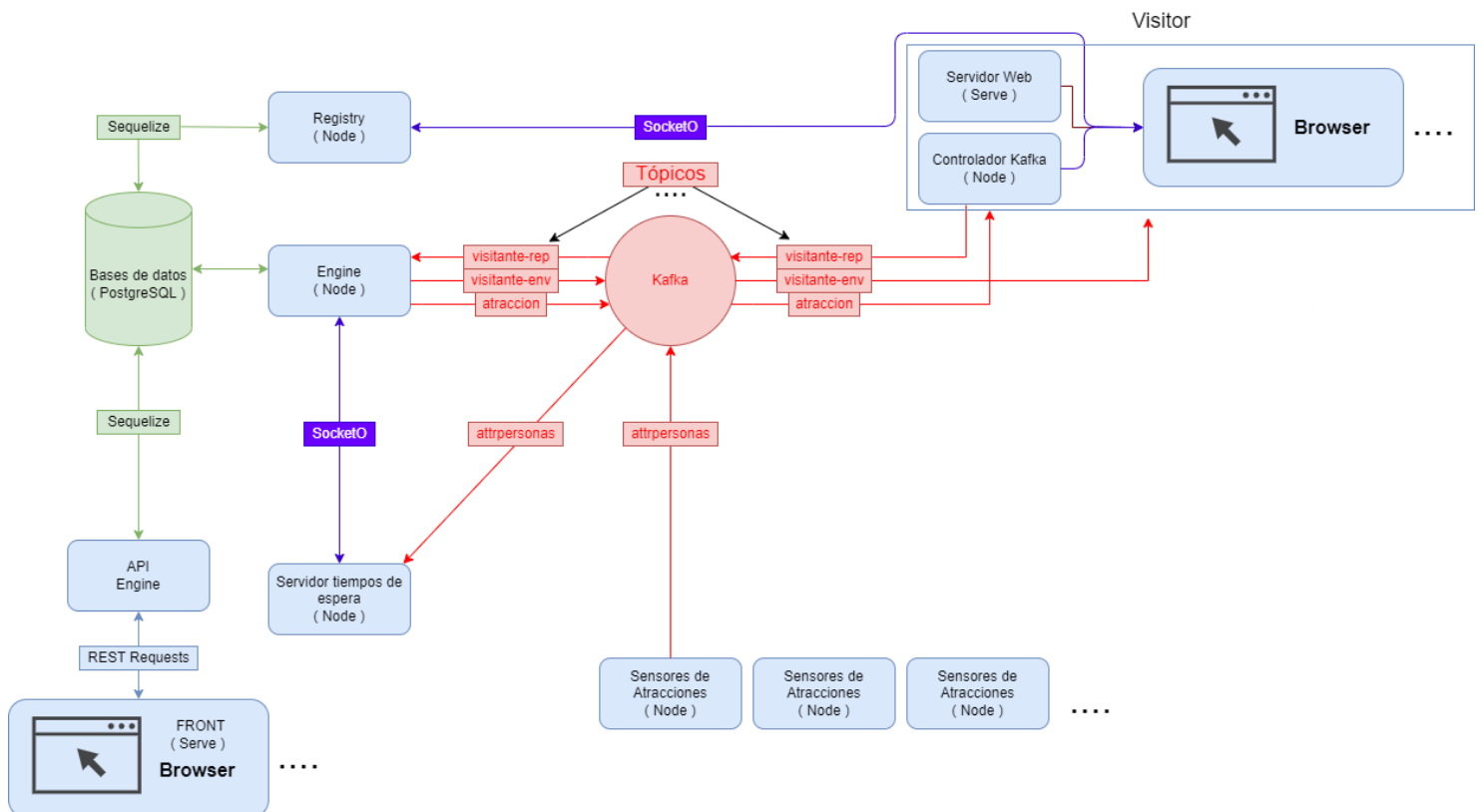
En este apartado se listan las tecnologías utilizadas en detalle, y para que se han usado en la práctica.

- [PostgreSQL](#): Se ha usado PostgreSQL para la base de datos.
- [Kafka](#): Se ha usado Kafka como servidor de mensajería e intermediario entre aplicaciones.
- [JavaScript & Node](#): Se ha usado el lenguaje de programación JavaScript, y el entorno de ejecución Node.
- [Sequelize](#): Se ha usado Sequelize como librería ORM para el control de transacciones entre la aplicación y la base de datos.
- [Bcrypt](#): Se ha usado la librería Bcrypt para “[Saltear](#)” las contraseñas de los usuarios en la base de datos, de esta manera nunca se expone la contraseña en texto plano en el sistema.
- [Socket.IO](#): Se ha usado la librería Socket.IO para el control y flujo de datos de sockets entre cliente - servidor.
- [Socket.IO-encrypt](#): Se ha usado esta librería middleware en conjunto con Socket.IO para encriptar la transmisión de datos entre cliente - servidor usando un “[Secret](#)”, usando un encriptado simétrico (el mismo secret sirve tanto para encriptar como para desencriptar, en ningún momento dicho secret se transmite por un medio, y ambas partes conocen dicho secret de antemano).
- [kafka-node](#): Se ha usado la librería kafka-node para el control de mensajes y conexión con el servidor de Kafka.
- [ReactJS](#): Se ha usado React como framework front-end para el desarrollo de las interfaces de usuario usando páginas web.
- [Concurrently](#): Se ha usado la librería Concurrently para la ejecución de dos comandos simultáneos independientemente del sistema operativo en el que se ejecuta, este se usa para ejecutar en el mismo contenedor Serve y el controlador de Kafka para los Visitantes.
- [Serve](#): Se ha usado la librería Serve para crear un pequeño servidor web el cual sirve la página web creada con React.
- [https](#): Se ha usado la librería interna de Node https para realizar el servidor REST con certificados en el Registry.
- [aes-encryption](#): Se ha usado esta librería para encriptar simétricamente con clave compartida las comunicaciones entre kafka y la plataforma.

- **Docker & docker-compose**: Se ha usado el sistema de encapsulación y contenerización Docker para el despliegue del sistema de manera distribuida, y docker-compose para desplegar Kafka y la Base de datos Postgres.

## Estructura del Sistema.

La estructura de nuestro sistema sería el siguiente, usando las tecnologías mencionadas en el diagrama para su ejecución.



## Dependencias.

Las dependencias generales del sistema son:

- SO ^Windows 7, ^Ubuntu 14.04 o derivados, Mac OS.
- Node ^v10.19.
- Docker ^20.10.8.
- docker-compose ^1.25.0.

## Script.

Hemos creado un script de lanzamiento en nuestro package.json para cada módulo de la aplicación para facilitar y estandarizar el lanzamiento de la aplicación.

```
"scripts": {  
  "start": "node ./FWQ_Sensor.js"  
},
```

De esta manera en vez que tener que, por ejemplo en el sensor, ejecutar el comando:

```
node ./FWQ_Sensor.js [parámetros]
```

Vamos a usar en todos los módulos:

```
npm start [parámetros]
```

Este script se usa tanto en producción como en despliegue dentro de los contenedores.

## Registry.

En el registry hemos añadido la función de generar logs.

El registry guarda en la base de datos todas las peticiones de gestión de cuentas que le envía el visitor ( registro, autenticación, edición, desconexión ).

A la hora de iniciar sesión, el Registry recibe la contraseña del usuario, la encripta, y la compara con la versión previamente encriptada guardada en la base de datos.

```
SELECT * FROM "users" LIMIT 50 (0.001 s) Modificar
```

<input type="checkbox"/> Modify	id	name	password	x_ac
<input type="checkbox"/> modificar	1	a	\$2b\$10\$QCGCT8/nzK7sdWGSSjOUQu2MbYm2QTwSYz1p/GIN2F3zIbYaw0R2K	10

## Visitor.

En visitor hemos añadido la función de realizar la comunicación con el registry mediante una API Rest o un socket dependiendo de lo que quiera el visitor.

A la hora de comunicarnos mediante una API se realiza con el protocolo TLS. Utilizando los certificados para conectarnos.

Y a la hora de conectarnos con sockets lo hacemos mediante sockets seguros utilizando encriptado simétrico con clave compartida.

## Controlador Kafka.

En nuestro caso encriptamos la comunicación de kafka cifrando antes de enviar los datos, y descifrando cuando se van a utilizar los datos recibidos. Esto lo hacemos con AES.

## Front

Nuestro front es básicamente igual que la interfaz del mapa del visitor, pero añadiendo la opción de modificar las ciudades para leer de la API OpenWeather.

Además de tener la opción de mostrar los logs recibidos por el API\_ENGINE, y de borrarlos todos si se quiere.

## Api Engine

Se encarga de leer de la base de datos los información que dejan tanto el registry como el Engine. Y mandarla a front como un log.

Además si desde el front se cambia el nombre de una ciudad la api engine se encargará de mandarlo a la base de datos para que el engine utilice esa información y deje pasar a los clientes a las atracciones.

Entonces el Api engine se encargará de responder al front, las atracciones, los logs, las ciudades y los usuarios.

Además de tener un botón el front que se encarga de borrar todos los logs.

#	Dirección	Acción	Datos	Fecha
1	172.20.42.151	registro	{\"name\":\"a\",\"password\":\"a\"}	2021-12-15T17:07:10.801Z
2	172.20.42.151	autenticar	{\"id\":\"1\",\"name\":\"a\",\"password\":\"a\",\"actual\":\"1\",\"y_destino\":\"1\",\"y_destino\":\"1\",\"y_destino\":\"1\",\"logged\":\"false\",\"updatedAt\":\"2021-12-15T17:07:11.012Z\",\"createdAt\":\"2021-12-15T17:07:11.012Z\"}	2021-12-15T17:07:31.927Z
3	172.20.42.151	desautenticar	{\"user\":{\"id\":\"1\",\"name\":\"a\",\"password\":\"2b8105449bADf5b0a0qTfcl0yqF088gPhv7C374U.en9k0p4WzEiW9\"},\"actual\":\"1\",\"y_destino\":\"14\",\"y_destino\":\"14\",\"y_destino\":\"14\",\"logged\":\"true\",\"createdAt\":\"2021-12-15T17:07:11.012Z\",\"updatedAt\":\"2021-12-15T17:29:54.622Z\"}	2021-12-15T17:29:55.175Z
4	172.20.42.151	desautenticar	{\"user\":{\"id\":\"1\",\"name\":\"a\",\"password\":\"2b8105449bADf5b0a0qTfcl0yqF088gPhv7C374U.en9k0p4WzEiW9\"},\"actual\":\"1\",\"y_destino\":\"14\",\"y_destino\":\"14\",\"y_destino\":\"14\",\"logged\":\"true\",\"createdAt\":\"2021-12-15T17:07:11.012Z\",\"updatedAt\":\"2021-12-15T17:29:54.622Z\"}	2021-12-15T17:29:55.414Z
5	172.20.42.151	autenticar	{\"name\":\"a\",\"password\":\"a\"}	2021-12-15T17:30:01.456Z
6	172.20.42.151	autenticar	{\"name\":\"a\",\"password\":\"a\"}	2021-12-15T17:45:26.118Z
7	172.20.42.151	editar	{\"userId\":\"1\",\"requestbody\":{\"id\":\"1\",\"name\":\"a\",\"password\":\"a\",\"actual\":\"14\",\"y_destino\":\"14\",\"y_destino\":\"14\",\"logged\":\"true\",\"createdAt\":\"2021-12-15T17:07:11.012Z\",\"updatedAt\":\"2021-12-15T17:36:51.866Z\"}}	2021-12-15T17:45:33.256Z
8	172.20.42.151	desautenticar	{\"user\":{\"id\":\"1\",\"name\":\"a\",\"password\":\"2b810588fJdWm1w0HWT5UJN1u7P9v0b0TATV0g0K300JMyt0\"},\"actual\":\"16\",\"y_destino\":\"14\",\"y_destino\":\"14\",\"y_destino\":\"14\",\"logged\":\"true\",\"createdAt\":\"2021-12-15T17:07:11.012Z\",\"updatedAt\":\"2021-12-15T17:45:34.887Z\"}	2021-12-15T17:45:35.064Z

[Limpiar Logs](#)

## Engine.

Hemos añadido en el engine los cifrados en el envío de información a kafka. Y la petición vía API a OpenWeather para recibir la información de las ciudades introducidas en el front. Además controlará los sectores del mapa para apagar o encender las atracciones, según la temperatura recibida.

# Guía de Despliegue.

Para desplegar, se necesitará la imagen encapsulada de cada módulo de docker.

En el engine y visitor reciben un parámetro extra llamado AES\_KEY además el engine recibe un parámetro de la API\_KEY para openWeather.

Los módulos del front y api\_engine se despliegan de la misma manera que el resto de módulos hechos en docker. Los parámetros y un ejemplo del comando para arrancarlo se encontrarán en los docker files correspondientes de cada módulo en el código proporcionado.

## Orden de lanzamiento.

No hay un orden obligatorio de lanzamiento, se pueden arrancar los módulos en el orden que se quiera.

## Capturas del funcionamiento.

Prueba del uso de HTTPS en las peticiones desde el visitor al registry.

The image shows a web application interface on the left and its DevTools network tab on the right.

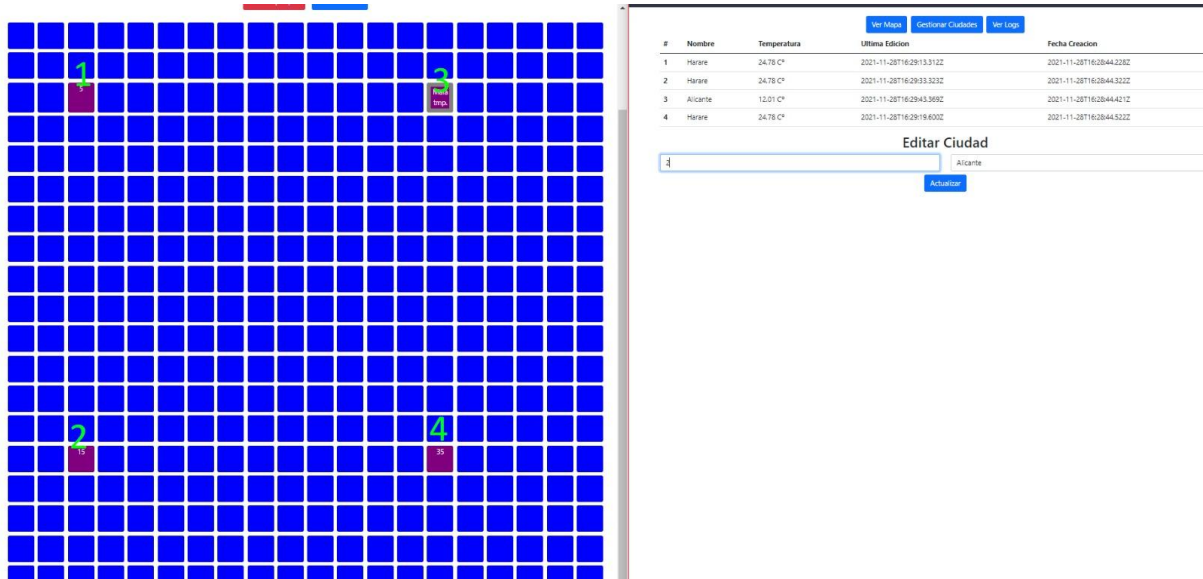
**Web Application Interface:**

- Buttons: "Usar Sockets" (red), "Usar REST" (green), "Salir del parque" (red), "Editar usuario" (blue).
- Text: "Nombre: aa ID: 1", "Siguiete Posición: 1, 14 Destino: 2, 14", "Password: 52b5105B5NF3DwinmIviseUWT5dJuN.i27FYx3bdTATV0gaJFJX3d0LMyiea", "Logged: true".
- Grid: A 10x10 grid of blue squares. One square at row 2, column 10 is green and labeled "aa". Another square at row 3, column 9 is red and labeled "30".

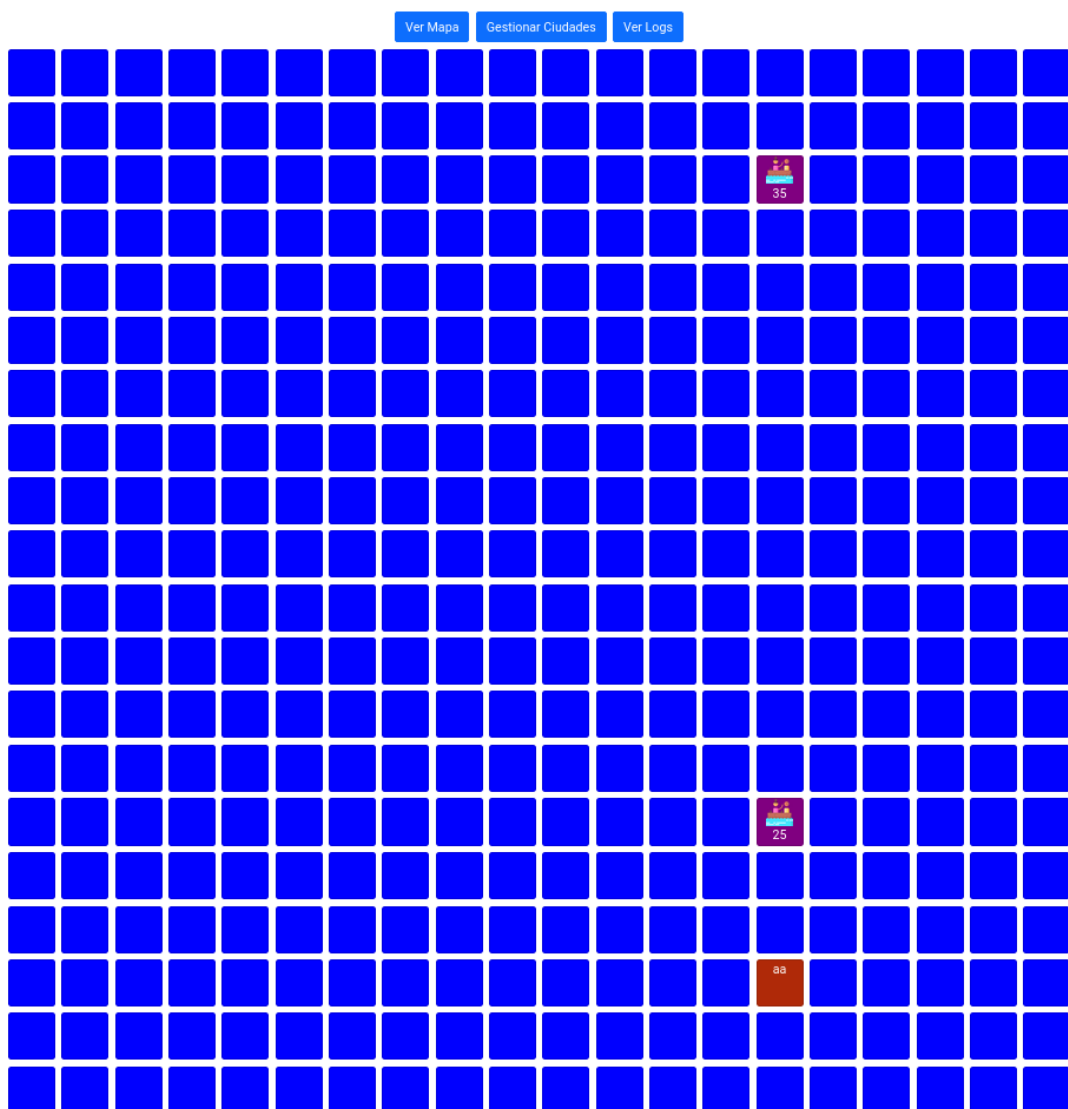
**DevTools Network Tab:**

- Filter: "auth".
- Request: "auth".
- Request URL: "https://172.20.42.151:9090/auth".
- Request Method: "POST".
- Status Code: "200 OK".
- Remote Address: "172.20.42.151:9090".
- Referrer Policy: "strict-origin-when-cross-origin".
- Response Headers: "Access-Control-Allow-Origin: \*", "Connection: keep-alive", "Content-Length: 240", "Content-Type: application/json; charset=utf-8", "Date: Wed, 15 Dec 2021 17:53:27 GMT", "ETag: W/"f8-hI8pg2uiCaEbdg3oTMM2efDK0M0\"", "Keep-Alive: timeout=5", "X-Powered-By: Express".
- Request Headers: "Accept: application/json, text/plain, /\*", "Accept-Encoding: gzip, deflate, br", "Accept-Language: es-ES,es;q=0.9", "Connection: keep-alive", "Content-Length: 28", "Content-Type: application/json", "Host: 172.20.42.151:9090", "Origin: http://172.20.42.152:5000/", "Referer: http://172.20.42.152:5000/", "sec-ch-ua: \"Chromium\";v=94\", \"Google Chrome\";v=94\", \";Not A", "sec-ch-ua-mobile: ?0", "sec-ch-ua-platform: \"Linux\"", "Sec-Fetch-Dest: empty".

Muestra del mapa, y de las ciudades por cuadrante.



Muestra del mapa del Front.



Muestra de la lista de Logs, enseñando la id, la MAC:IP, la acción, los datos, y la fecha.

Ver Mapa

Seleccionar Ciudad

Ver Logs

#	Dirección	Acción	Datos	Fecha
1	::ffff:172.20.42.151	registro	{\"name\":\"a\",\"password\":\"a\"}	2021-12-15T17:07:10.801Z
2	::ffff:172.20.42.151	autenticar	{\"id\":\"1\",\"name\":\"a\",\"password\":\"a\",\"x_actual\":\"y\",\"y_destino\":\"y\",\"logged\":\"false\",\"updatedAt\":\"2021-12-15T17:07:11.011Z\",\"createdAt\":\"2021-12-15T17:07:11.011Z\"}	2021-12-15T17:07:31.927Z
3	::ffff:172.20.42.151	desautenticar	{\"user\":{\"id\":\"1\",\"name\":\"a\",\"password\":\"\$2b\$10\$49HADf\$0namqTnlcy8FuB8bpX7C374J.zeV9kxpuN7zeEIW9\",\"x_actual\":\"y\",\"y_destino\":\"y\",\"logged\":\"true\",\"createdAt\":\"2021-12-15T17:07:11.011Z\",\"updatedAt\":\"2021-12-15T17:29:54.622Z\"}}	2021-12-15T17:29:55.175Z
4	::ffff:172.20.42.151	desautenticar	{\"user\":{\"id\":\"1\",\"name\":\"a\",\"password\":\"\$2b\$10\$49HADf\$0namqTnlcy8FuB8bpX7C374J.zeV9kxpuN7zeEIW9\",\"x_actual\":\"y\",\"y_destino\":\"y\",\"logged\":\"true\",\"createdAt\":\"2021-12-15T17:07:11.011Z\",\"updatedAt\":\"2021-12-15T17:29:54.622Z\"}}	2021-12-15T17:29:55.414Z
5	::ffff:172.20.42.151	autenticar	{\"name\":\"a\",\"password\":\"a\"}	2021-12-15T17:30:01.456Z
6	::ffff:172.20.42.151	autenticar	{\"name\":\"a\",\"password\":\"a\"}	2021-12-15T17:45:26.118Z
7	::ffff:172.20.42.151	editar	{\"userId\":\"1\",\"requestbody\":{\"id\":\"1\",\"name\":\"a\",\"password\":\"a\",\"x_actual\":\"y\",\"y_destino\":\"y\",\"logged\":\"true\",\"createdAt\":\"2021-12-15T17:07:11.011Z\",\"updatedAt\":\"2021-12-15T17:36:51.866Z\"}}	2021-12-15T17:45:33.256Z
8	::ffff:172.20.42.151	desautenticar	{\"user\":{\"id\":\"1\",\"name\":\"a\",\"password\":\"\$2b\$10\$8\$NFJ0wimtlwvslWITS6JUnLk7PvVbYbYtATVUpuJ5J0dLMyleI\",\"x_actual\":\"y\",\"y_destino\":\"y\",\"logged\":\"true\",\"createdAt\":\"2021-12-15T17:07:11.011Z\",\"updatedAt\":\"2021-12-15T17:45:34.687Z\"}}	2021-12-15T17:45:35.064Z

Limpia Logs