

**CHALLENGER: TRANSITO\_APP**

**MANUAL FUNCIONAL E INSTRUCTIVO DE CONFIGURACIÓN DEL SISTEMA TRANSITO\_APP**

**Versión: v01**

**Fecha:05/03/2023**

1. **Repositorio GITHUB:**

[**https://github.com/edr122/transito-app**](https://github.com/edr122/transito-app)

1. **Imagen de la aplicación en Dockerhub:**

[**https://hub.docker.com/r/edr122/transito\_app**](https://hub.docker.com/r/edr122/transito_app)

1. **SUPUESTOS:**

* En el punto 1.c del documento menciona un único número identificatorio , en el cual se ha asumido que es mismo **id** (int autoincremental) de tabla.
* Adicional se ha agregado un campo **usuario\_ap**p en la tabla **oficial** para que sirva como input para generar el token.
* También se ha creado una tabla infracción que está relacionado con un oficial (quien registra la infracción) y con un vehículo.

1. **Ejecución de la aplicación desde el repositorio de github:**

### Requisitos generales

* python 3.8
* mysql 5.7 o superior
* Instalar *Pipenv* => pip install pipenv

### Instalación del sistema

Se va desplegar en un entorno virtual de python. Clonamos el proyecto:

git clone https://github.com/edr122/transito-app.git  
cd transito-app

Creamos el entorno virtual

pipenv shell

Instalamos todos requerimientos (dependencias)

pipenv install -r requirements.txt

Ejecutar el proyecto

python main.py

Nota: El proyecto se va ejecutar en el puerto 3000 *Ejemplo:http://localhost:3000/*

### Cargar la base de datos

Diagrama de base de datos

Ejecutar el siguiente script <db_estructura.sql> para cargar la información

**IMPORTANTE:** Actualizar la configuración de base de datos en el archivo <db.py>

# Mysql Settings  
app.config['MYSQL\_HOST'] = 'localhost'  
app.config['MYSQL\_USER'] = 'xxxx'  
app.config['MYSQL\_PASSWORD'] = 'xxxx'  
app.config['MYSQL\_DB'] = 'xxxx'

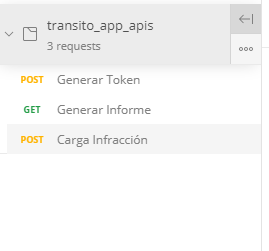
1. **Ejecución de la aplicación desde el docker:**

La imagen de la aplicación en dockerhub está como [edr122/transito\_app](https://hub.docker.com/r/edr122/transito_app). Para desplegar conjuntamente con la base datos tenemos que ejecutar el archivo <docker-compose.yml>. Este archivo lo podemos encontrar en el repositorio de github. Para ejecutar el entorno lanzamos el siguiente comando.

docker-compose up -d

luego ingresamos a la siguiente dirección: *http://localhost:5000/* y listo.

1. **Colección de APIS:**

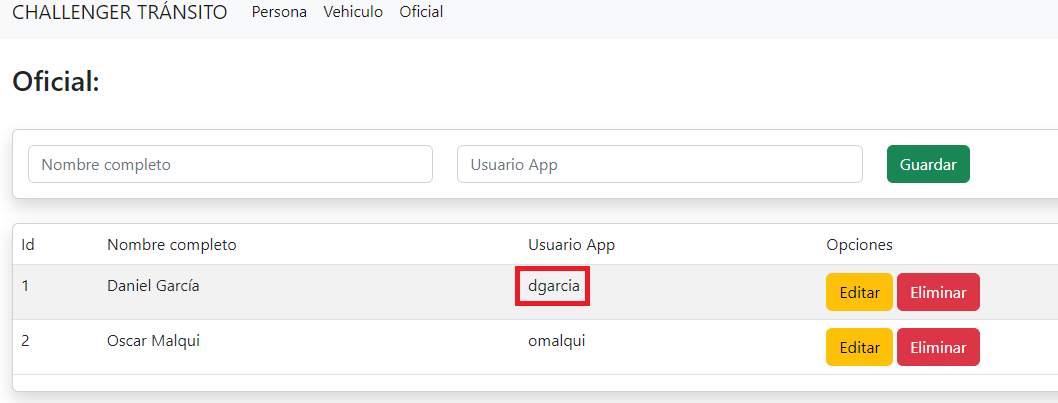


Del repositorio de github existe una colección (transito\_app\_apis.postman\_collection.json) de postman de las APIS disponibles.

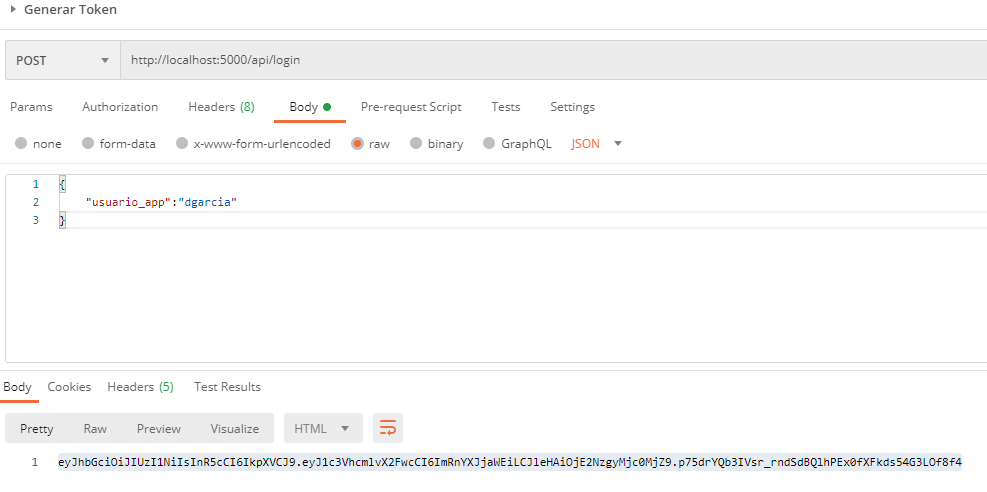
1. **Generar token para registrar una infracción:**

Como se mencionó en el **punto 3,** se agregado un campo que es usuario\_app en la tabla oficial para generar el token.

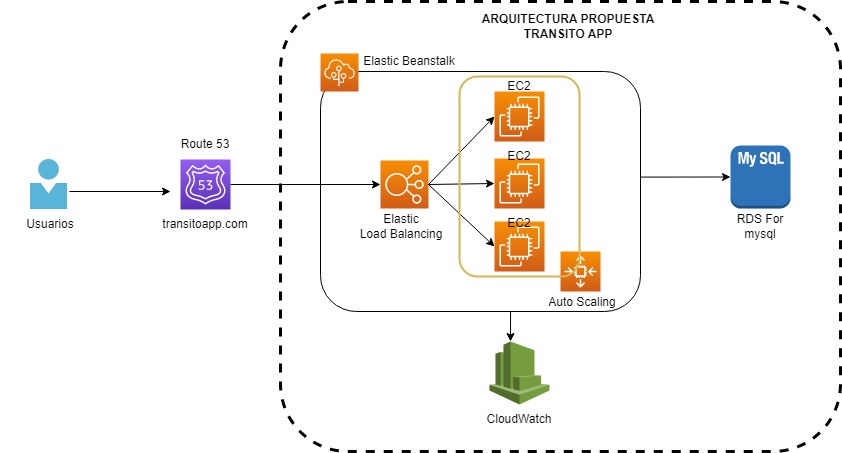
Entonces para generar el token cogemos el usuario\_app de un oficial creado por ejemplo (dgarcia).



Vamos a Postman y usamos ese usuario\_app en el endpoint Generar Token (<http://localhost:5000/api/login>), y generamos el token.



1. **Arquitectura AWS:**

****