

## Reporte Técnico

León Villapún, Luis Alfredo  
A01322275

Canto García, Armando  
A01322361

Rodiles Legaspi, Ricardo  
A01325081

Tovar Muñoz de Cote, Alejandro  
A01328484

May 3, 2018

## Contents

|          |                         |          |
|----------|-------------------------|----------|
| <b>1</b> | <b>Introducción</b>     | <b>2</b> |
| <b>2</b> | <b>Github</b>           | <b>2</b> |
| <b>3</b> | <b>Trello</b>           | <b>2</b> |
| <b>4</b> | <b>Database</b>         | <b>2</b> |
| <b>5</b> | <b>REST API</b>         | <b>2</b> |
| <b>6</b> | <b>Frontend Docker</b>  | <b>3</b> |
| <b>7</b> | <b>Frontend Testing</b> | <b>3</b> |
| <b>8</b> | <b>Backend Testing</b>  | <b>3</b> |
| <b>9</b> | <b>Jenkins</b>          | <b>3</b> |

## 1 Introducción

Dentro del marco del curso de Laboratorio de Desarrollo Web del ITESM Campus Puebla, se nos requirió hacer como proyecto final una plataforma web con temática a elegir.

La temática elegida para el proyecto fue una página de viajes. La idea en general es la de tener una página que fomente el turismo en Puebla y se puedan organizar viajes.

Así, se concibió un proyecto viable e interesante, que cumple con los objetivos del curso.

## 2 Github

La liga para el repositorio de github es la siguiente: <https://github.com/Action52/PueblaViajes>.

Aquí realizamos la organización de acuerdo a lo establecido por el forking model.

## 3 Trello

## 4 Database

La base de datos fue montada en AWS con la ayuda de su funcionalidad RDS, para ser más específicos, como una base de Postgres. Se puede acceder a la base de datos con las siguientes credenciales:

Usuario: viajesPuebla

Password: viajes123

Puerto: 5432

Host: dbviajes.c72vt9uachn0.us-east-2.rds.amazonaws.com

Nombre de la base: viajesPuebla

## 5 REST API

Contamos con varios endpoints para que la aplicación funcione adecuadamente.

/viajes/ despliega todos los viajes o crea uno nuevo, dependiendo de la petición.

/viajes/id/ despliega la información de uno en específico, lo modifica o lo borra, dependiendo de la petición.

/users/ despliega todos los usuarios o crea uno nuevo, dependiendo de la petición.

/users/id/ despliega la información de uno en específico, lo modifica o lo borra, dependiendo de la petición.

## **6 Frontend Docker**

## **7 Frontend Testing**

## **8 Backend Testing**

El testing del Backend se desarrolló con la ayuda que trae integrada Django. Se desarrollaron diversos test. Para probarlos simplemente se necesita correr el `manage.py` con la instrucción `test`.

## **9 Jenkins**

Jenkins fue integrado de manera exitosa en la instancia de nuestro proyecto. Para acceder al mismo: <http://18.217.0.33:8080/>