

Manual técnico para el sistema Gestión de Accesos y Aforo

Camilo Andrés Gutierrez Torres
Edixon Alirio Salas Medina
Francisco Suárez
Pontificia Universidad Javeriana Cali
Cali, Colombia
Diciembre 2020

1 Instrucciones de instalación

El sistema está montado sobre un servidor en Ohio - Estados Unidos que pertenece a la empresa Amazon a través de Amazon Web Services, no es necesario que se instale localmente en las máquinas de los usuarios, debido a que es una página web a la cual se puede ingresar a través de la IP : <http://3.20.225.166/>.

Para que el sistema funcione, los usuarios deben estar conectados a Internet y la conexión a la página web puede hacerse ya sea desde dispositivos móviles o de escritorio.

2 Descripción de la implementación

El backend del sistema fue desarrollado con el framework Flask de python, y se utilizaron diferentes librerías como flask_cors, la cual es una extensión de Flask para manejar Cross Origin Resource Sharing (CORS) [1], y PYjwt, la cual sirve para crear tokens a través de JSON Web Tokens, los cuales son información codificada que puede decodificarse haciendo el proceso inverso [2]. Además, al usar el motor de base de datos MongoDB, usamos la librería pymongo para comunicar python con la base de datos. Para realizar el frontend usamos angular, el cual es un framework para realizar proyectos front-end desarrollado por Google [3].

El sistema se encuentra alojado en un servidor con 3 instancias de AWS, el cual se puede acceder a través de una IP ubicada en Estados Unidos. En una

de las instancias está el frontend, en otra está el backend y en otra está la base de datos.

Para implementar algunas funcionalidades del sistema, se utilizaron las siguientes librerías de Angular:

- ngx-cookie-service.
- jwt-decode.
- file-saver.
- XLSX.
- html2pdf.js.

3 Listado de Archivos Fuente

El sistema cuenta con dos carpetas principales para el backend y el frontend, llamadas Backend y gestionAforoCOVID. En la carpeta Backend, nos podemos encontrar con 5 carpetas, correspondientes a cada servicio: Parámetros, Pruebas, Usuarios, Reportes, Visitas. Dentro de cada una de ellas se encuentran dos carpetas y un archivo main.py, las carpetas corresponden al llamado a la instancia de base de datos y a las constantes del sistema. El archivo main.py es el que se ejecuta para poner a funcionar el servicio que se requiera.

Dentro de la carpeta gestionAforoCOVID correspondiente al frontend de la aplicación, podemos encontrarnos con un proyecto de angular el cual, entre otras cosas, contiene una carpeta llamada src, dentro de la cual hay una carpeta llamada app, dentro de esta carpeta, podemos encontrarnos con los componentes del sistema, la carpeta landing-page contiene los componentes correspondientes a la página principal del sistema, la carpeta services contiene los servicios que envían peticiones http al backend y envían o devuelven información, la carpeta user contiene los componentes correspondientes a las secciones de ingreso de usuarios y registro de usuarios, dentro de la carpeta correspondiente a ese registro llamada "user-options-registration", podemos encontrar opciones de registro según el tipo de usuario. Finalmente, tendremos la carpeta user-profile, en la cual se encuentran todos los componentes que corresponden a los perfiles de los usuarios.

4 Bibliografía

- [1] "Flask-CORS — Flask-Cors 3.0.9 documentation", Flask-cors.readthedocs.io, 2020. [Online]. Available: <https://flask-cors.readthedocs.io/en/latest/>. [Accessed: 07- Dec- 2020]

[2] "Welcome to PyJWT — PyJWT 1.7.1 documentation", Pyjwt.readthedocs.io, 2020. [Online]. Available: <https://pyjwt.readthedocs.io/en/stable/>. [Accessed: 07- Dec- 2020]

[3] "Angular", Angular.io, 2020. [Online]. Available: <https://angular.io/docs>. [Accessed: 07- Dec- 2020]