# Reporte configuración del servidor App Covid

Mario Alberto Encinas Cardona

Facultad de Ciencias de la Computación, Benemérita Universidad Autónoma de Puebla. México

mario.encinasc@alumno.buap.mx

**Abstract.** El avance de las tecnologías computacionales ha permitido la recopilación de datos acerca de la pandemia causada por el virus SARS-CoV-2; la gran cantidad de datos recopilados, y el fácil acceso a estos ha facilitado a los investigadores la tarea de analizar el comportamiento del virus y su propagación.

Sin embargo, a pesar de la capacidad de los equipos de cómputo actuales, el gran volumen de datos disponibles resulta difícil de procesar, a menos que se posean las herramientas necesarias. Aunque muchas herramientas para el procesamiento de grandes volúmenes de datos son de uso libre, la curva de aprendizaje que estas poseen resulta prohibitiva, pues se requiere un amplio conocimiento técnico para poder aprovechar todas las ventajas que ofrecen.

Es debido a esto que se ha propuesto la creación de una herramienta cuyo objetivo es facilitar el análisis y procesamiento de los datos proporcionados por el gobierno de México respecto a los individuos afectados por la enfermedad COVID-19.

## 1 Introducción

A finales del año 2019 en Wuhan se detectaron los primeros casos de la enfermedad COVID-19, a pesar de las medidas aplicadas en la ciudad el virus se propagó a otras regiones de China y rápidamente se extendió al resto del mundo.

A pesar de los conocimientos actuales tanto en epidemiología como en biología el virus demostró ser extremadamente difícil de manejar; y rápidamente se convirtió en una de las peores pandemias de los últimos años.

Incluso con lo anteriormente descrito, es un hecho que las tecnologías y conocimientos modernos han logrado apalear en gran medida los efectos que pudo haber tenido un virus de tales características en la población mundial.

Aun queda un largo camino por recorrer en el manejo y contención de pandemias antes que podamos contenerlas con total eficacia, por lo que diversas instituciones se han dedicado a la recopilación de datos acerca de los infectados con el virus SARS-CoV-2, y junto a este volumen titánico de datos surge la necesidad de analizar la información que se encuentra dentro de ellos; sin embargo dada la gran cantidad disponible, es imposible analizarlos adecuadamente a menos que se posea el conjunto de herramientas adecuado; sin embargo existe un problema.

Las herramientas para análisis y procesamiento de datos modernas son capaces de detectar patrones tan complejos que resultaría imposible detectarlos por cualquier otro método; sin embargo, aunque se posea el conocimiento teórico necesario para interpretar correctamente cualquier resultado, también es necesario poseer

un conocimiento técnico relativamente especializado para poder aprovechar completamente estas tecnologías.

Con el objetivo de Facilitar tareas de análisis a los datos disponibles por parte de especialistas que no posean conocimiento técnico acerca de las herramientas necesarias, se ha propuesto la creación de una Web App que permita realizar tareas de reconocimiento de patrones sin que el creciente volumen de información resulte en un impedimento.

Se ha decidido implementar la herramienta como una aplicación web para permitir que la calidad de los resultados, así como el tiempo que demore el sistema en entregar una respuesta no sea dependiente del equipo perteneciente al investigador que realice el proceso.

Así mismo se ha decidido utilizar el lenguaje de programación Python y el gestor de base de datos MySQL, el motivo de esto es la disponibilidad de bibliotecas gratuitas para análisis de datos en Python, y la accesibilidad del gestor MySQL de uso gratuito, hacen de los sistemas basados en dichas tecnologías relativamente fáciles de mantener, sin que se comprometa la calidad de las herramientas

## 2 Diseño.

Para permitir la creación de una aplicación amigable con el usuario, sin complicar las tareas de desarrollo se optó por la utilización de un framework; tras consultar con diversas fuentes se decidió utilizar Django, los principales motivos de esta decisión fueron su robustez, la gran disponibilidad de la documentación existente y por supuesto, su orientación a Python.

El framework Django permite la creación de Web Apps basadas en el patrón de arquitectura Model Template View (MTV), de este modo se separa de forma efectiva la lógica que maneja las vistas, la interfaz de usuario y el manejo de la base de datos, la implementación de una arquitectura de este estilo permite que la aplicación posea un diseño modular facilitando las tareas de mantenimiento, sin embargo debido a las necesidades del sistema se tuvo que realizar una modificación a dicho patrón de arquitectura.

Django permite la utilización de una gran cantidad de motores de base de datos, así mismo la flexibilidad del framework permite que desde la página web se pueda ejecutar prácticamente cualquier script o función creada en Python esto hace posible la utilización archivos para el almacenamiento y manejo de datos, por ejemplo, archivos con extensión CSV o XLS para el almacenamiento de tablas, sin embargo, debido a la naturaleza del proyecto se decidió utilizar el motor de bases de datos MySQL.

Debido a la configuración del servidor solamente es posible acceder a la base de datos desde localhost, sin embargo, esto no representa ningún impedimento o dificultad para el desarrollo del proyecto, así mismo existen dos métodos para establecer una conexión con la base de datos: El método mas sencillo es por medio del framework Django y las funciones que ofrece para manejo de bases de datos MySQL, la segunda opción es usando bibliotecas de Python, en este caso se decidió utilizar “mysqlclient”.

El servidor proporcionado es administrado por un tercero, y únicamente se ha proporcionado acceso por medio del protocolo FTP, de modo que no es posible modificar la configuración del servidor, sin embargo es posible contactar con el administrador y solicitar que ejecute tareas que requieran el uso de terminal; no obstante debido al uso del framework Django, la configuración de la aplicación y el servidor se encuentran separadas y es posible modificar la configuración de la aplicación sin necesidad de solicitar al administrador que realicé los cambios, aunque será necesario reiniciar el servidor para poder visualizar los cambios en la página web.

**NOTA:** Las credenciales de acceso al servidor mediante el protocolo FTP no se adjuntan en el presente documento por motivos de seguridad,

Favor de contactar a Mario Alberto Encinas Cardona: **mario.encinasc@alumno.buap.mx.**

### 2.1 Arquitectura.

Dada la naturaleza de la aplicación es necesario variar un poco el funcionamiento de la arquitectura MTV, la variante será denominada MLTV (Model, Logic, Template, View), el motivo de esta decisión se debe a la alta complejidad que poseen varios de los algoritmos que se emplearán en la aplicación, siendo necesario separarlos en una capa extra.

**Model:** Contiene la información de la base de datos, como acceder a los datos y almacenar nuevos datos.

**Logic:** Contiene las funciones y métodos necesarios para realizar las tareas de administración y procesamiento de los datos, esto incluye algoritmos de aprendizaje automático y funciones para filtrar y actualizar los datos recientes.

**Template:** Contiene archivos relacionados a la visualización de los datos y la forma en que estos son presentados.

**View (no confundir con el View del modelo MVC):** Define que información se mostrará al usuario y define el template apropiado.

### 2.2 Base de datos

El diseño del modelo para la base de datos es bastante simple gracias a que los datos proporcionados por el gobierno se encuentran originalmente organizados dentro de una sola tabla, sin embargo, para facilitar tanto las consultas como la interpretación de los datos el modelo se encuentra organizado por medio de tres tablas relacionadas.

Dado que Django no permite la creación de entidades débiles las tres tablas poseen una llave primaria correspondiente con el identificador del caso al que pertenecen.

Cada una de las tres tablas que conforman el modelo para el almacenamiento de datos que manejará la app está orientado a organizar un tipo de información respecto a los casos de COVID-19 en México. La información que almacenada está agrupada de la siguiente manera: antecedentes de enfermedades del paciente, información del caso, y datos médicos del caso.

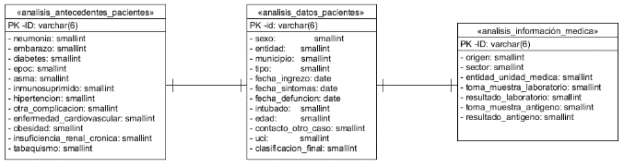


**Fig. 1.** Diagrama entidad relación de la base de datos implementada en el servidor.

### 2.3 Versiones y bibliotecas

El servidor tiene instalado Python 3 y como tal cualquier biblioteca, dependencia y script debe estar desarrollado dentro de dicha versión, específicamente Python 3.7.10, mientras que la versión de Django instalada es la 3.2.4. Si bien Python posee problemas de retrocompatibilidad, es posible utilizar versiones similares y no es estrictamente necesario usar exactamente las mismas versiones de las bibliotecas instaladas.

La lista completa de bibliotecas de Python instaladas dentro del servidor se encuentra en el archivo “pip.requeriments.txt”.

## 3 Implementación.

Si bien el servidor se encuentra implementado y configurado mediante el framework Django, no se imposibilita la implementación directa de otras tecnologías; específicamente la utilización de funciones Python que compilen directamente código HTML, sin embargo, no se recomienda la utilización de dicha alternativa.

La configuración original del servidor implementa múltiples archivos alojados en la carpeta raíz, como el caso de “Passenger_wsgi.py”, las carpetas “tmp” y “public, además de dos archivos

ocultos y el archivo “app-covid.log”.

La creación del proyecto usando framework se realizó directamente sobre la carpeta raíz, de modo que es aquí donde se encuentra el archivo “manage.py” el cual es el encargado de administrar y hacer efectivos los cambios aplicados tanto al modelo para la base de datos, como la configuración del servidor.

La carpeta principal del proyecto es “appcovid”, dentro de esta se encuentran los archivos de configuración y el archivo central para la definición de las direcciones dentro de la pagina web; actualmente los archivos de esta carpeta se encuentran configurados para el correcto funcionamiento del servidor y no es necesario modificar ningún elemento.

Por otra parte, la carpeta “analisis” contiene los archivos de la aplicación, cualquier cambio que se desee aplicar al contenido de la pagina web debe ser implementado en este archivo.

Dentro de la carpeta “análisis” se encuentran diversos archivos y carpetas, algunos de los más importantes son los siguientes:

- **urls.py:** Archivo de definición de direcciones de la aplicación, debido a la configuración de Django todas las direcciones definidas en este directorio son relativas a url del servidor (http://app-covid.siycise.org/).
- **models.py:** define el modelo utilizado para las tablas en la base de datos (Nota: debido a la configuración de Django todas las tablas definidas dentro de este archivo se almacenarán en la base de datos con el prefijo “análisis_”).
- **templates:** Se usa para almacenar los archivos HTML.
- **static:** se usa para definir los archivos de formato (CSS, javascript, etc.)
- **migrations:** Esta carpeta es utilizada por el sistema para almacenar scripts que permiten modificar la estructura de la base de datos.
- **data:** Se utiliza para almacenar el archivo CSV descargado desde la página del gobierno.
- **logic:** contiene los archivos py con los cuales se realizan las tareas correspondientes a la capa lógica de la aplicación, la presencia de un archivo “__init.__py” epermite que Python identifique dicha carpeta como un paquete, lo que permite importar los archivos en su interior como cualquier otro paquete de las bibliotecas de Python.

La instalación y configuración de Python fue realizada siguiendo la documentación oficial de Django.

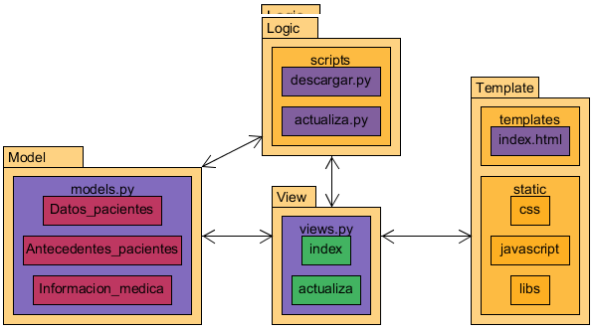


***Fig. 2.*** *Modelo MLTV de la aplicación, únicamente se visualizan las carpetas(amarillo), archivos(morado), clases(magenta) y funciones(verde) implementadas hasta ahora en el servidor. Todos los archivos y carpetas en el modelo se encuentran dentro de la carpeta “análisis” esto debido a que la totalidad de los archivos propios de la aplicación se encuentran ahí.*

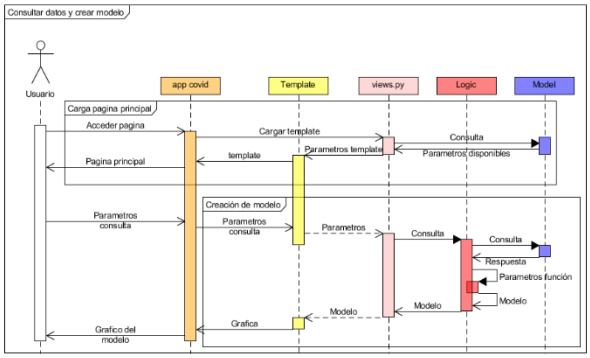


***FIG 3:*** *Diagrama de secuencias para el proceso de creación de un modelo basado en algoritmos de reconocimiento de patrones para el análisis de los datos disponibles.*

## References


1. Mysqlclient: https://mysqlclient.readthedocs.io
2. Django: https://docs.djangoproject.com/en/3.2/