

## FARMA-APP

### Requisitos funcionales

<b>RF 001</b>	<b>Buscar medicamento</b>
<b>Detalle</b>	El aplicativo buscara un medicamento en diferentes páginas web.
<b>Precondición</b>	El usuario debe haber ingresado al aplicativo
<b>Secuencia</b>	<ol style="list-style-type: none"><li>1. Ingresar dirección del usuario</li><li>2. Ingresar el nombre del medicamento</li><li>3. Elegir de una lista la presentación del medicamento</li><li>4. Botón Buscar</li><li>5. Botón micrófono</li></ol>
<b>Criterios</b>	El sistema debe buscar medicamentos que contengan los criterios ingresados por el usuario

<b>RF 002</b>	<b>Resultado de la búsqueda</b>
<b>Detalle</b>	Mostrar los resultados relevantes a la búsqueda realizada por el usuario.
<b>Precondición</b>	El usuario debe haber ingresado un medicamento y una presentación
<b>Secuencia</b>	<ol style="list-style-type: none"><li>1. El sistema procesa la consulta de búsqueda ingresada por el usuario.</li><li>2. Buscar en cada url de las farmacias y comparar ubicación y precio</li><li>3. Mostrar lista del medicamento encontrado poniendo al comienzo de la lista el más cercano y económico para usuario</li><li>4. El usuario puede seleccionar uno de la lista para ver más detalles</li></ol>
<b>Criterios</b>	<ul style="list-style-type: none"><li>- El sistema debe mostrar una lista de medicamentos encontrados</li><li>- El usuario debe poder seleccionar un medicamento para ver más detalles</li><li>- El sistema debe mostrar información detallada del medicamento seleccionado, como nombre, presentación, laboratorio, precio, disponibilidad.</li><li>- El sistema debe mostrar la ubicación de la farmacia y si tiene domicilio.</li></ul>

<b>RF 003</b>	<b>Ordenamiento</b>
<b>Detalle</b>	El aplicativo debe permitir el ordenamiento de los resultados según diferentes criterios.
<b>Precondición</b>	El usuario debe haber realizado una búsqueda en el sistema.
<b>Secuencia</b>	<ol style="list-style-type: none"><li>1. El usuario selecciona la lista de elementos que desea ordenar.</li><li>2. El usuario selecciona el criterio de ordenamiento deseado (como el precio, la disponibilidad, laboratorio, etc.)</li><li>3. El sistema ordena la lista de elementos en base al criterio seleccionado.</li><li>4. El sistema muestra la lista de elementos ordenada al usuario.</li></ol>
<b>Criterios</b>	<ul style="list-style-type: none"><li>- El sistema debe permitir al usuario ordenar la lista de elementos en base a al menos tres criterios diferentes.</li><li>- El sistema debe mostrar la lista de elementos ordenada de manera clara y fácil de entender para el usuario.</li><li>- El sistema debe guardar la selección de ordenamiento del usuario para la próxima vez que acceda a la lista de elementos.</li></ul>

<b>RF 004</b>	<b>Notificaciones</b>
<b>Detalle</b>	El aplicativo debe permitir la configuración de notificaciones para alertar al usuario cuando se encuentra un medicamento en una página web específica o cuando hay una oferta disponible.
<b>Precondición</b>	El usuario debe tener una cuenta en el sistema y haber otorgado permiso para recibir notificaciones.
<b>Secuencia</b>	<ol style="list-style-type: none"> <li>1. El sistema informa de ofertas y medicamentos disponibles al usuario.</li> <li>2. El usuario recibe la notificación en su dispositivo móvil o en su correo electrónico.</li> <li>3. El usuario puede abrir la notificación para leer el mensaje completo o descartarla.</li> </ol>
<b>Criterios</b>	<ul style="list-style-type: none"> <li>- El sistema debe ser capaz de enviar notificaciones a los usuarios en tiempo real.</li> <li>- Las notificaciones deben ser claras y concisas, para no confundir al usuario.</li> <li>- El usuario debe tener la opción de desactivar las notificaciones en cualquier momento.</li> </ul>

### Requisitos no funcionales

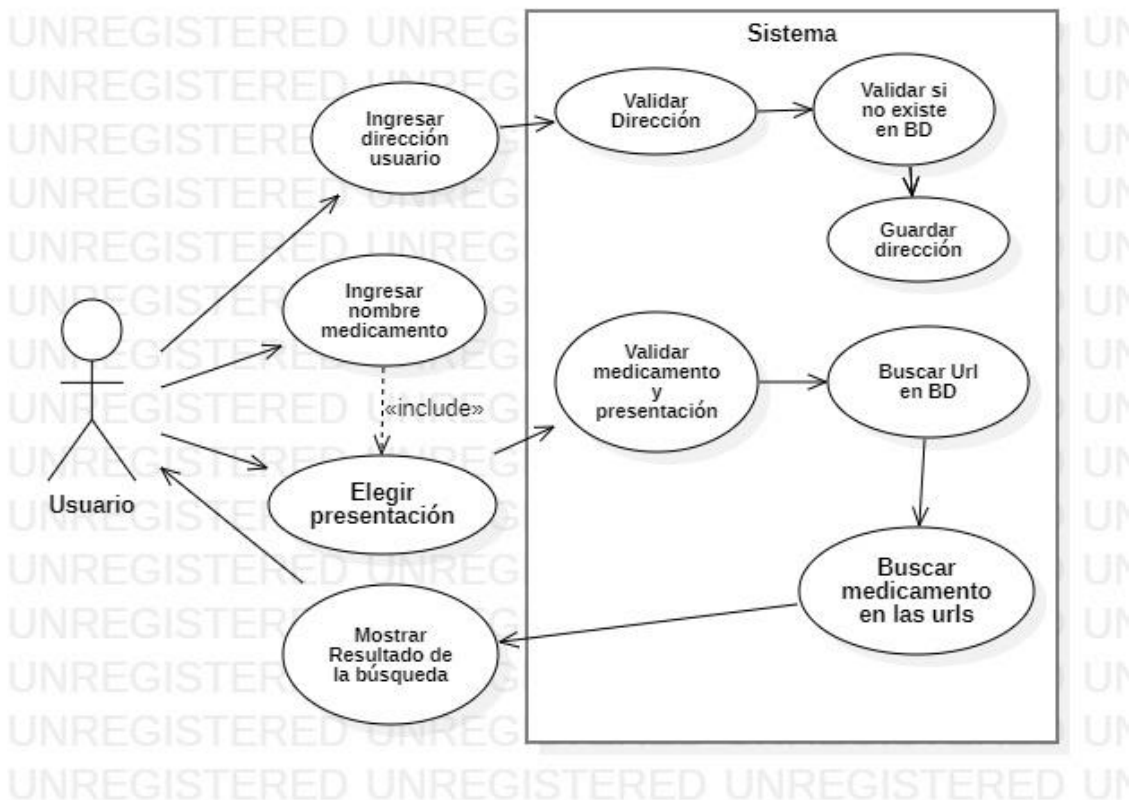
<b>RNF 001</b>	<b>Seguridad</b>
<b>Detalle</b>	El aplicativo debe garantizar la seguridad de los datos del usuario
<b>Actividad</b>	<ol style="list-style-type: none"> <li>1. Identificar los requisitos de seguridad para el sistema.</li> <li>2. Implementar medidas de seguridad adecuadas, como el cifrado y la autenticación.</li> <li>3. Realizar pruebas de penetración y evaluaciones de vulnerabilidades para detectar posibles debilidades en el sistema.</li> </ol>

<b>RNF 002</b>	<b>Mantenibilidad</b>
<b>Detalle</b>	El aplicativo debe ser fácil de mantener y actualizar, con una arquitectura modular y un código limpio y estructurado.
<b>Actividad</b>	<ol style="list-style-type: none"> <li>1. Establecer estándares de codificación y documentación para asegurar la comprensión y facilidad de modificación de código.</li> <li>2. Realizar pruebas y depuraciones periódicas para detectar errores y fallos que puedan afectar la estabilidad del sistema.</li> <li>3. Proporcionar una arquitectura clara y bien estructurada para facilitar el mantenimiento.</li> <li>4. Proporcionar una documentación detallada y actualizada sobre el sistema para facilitar su mantenimiento y futuras mejoras.</li> <li>5. Establecer procesos de actualización y parches para mantener el sistema actualizado y corregir fallos de seguridad.</li> </ol>

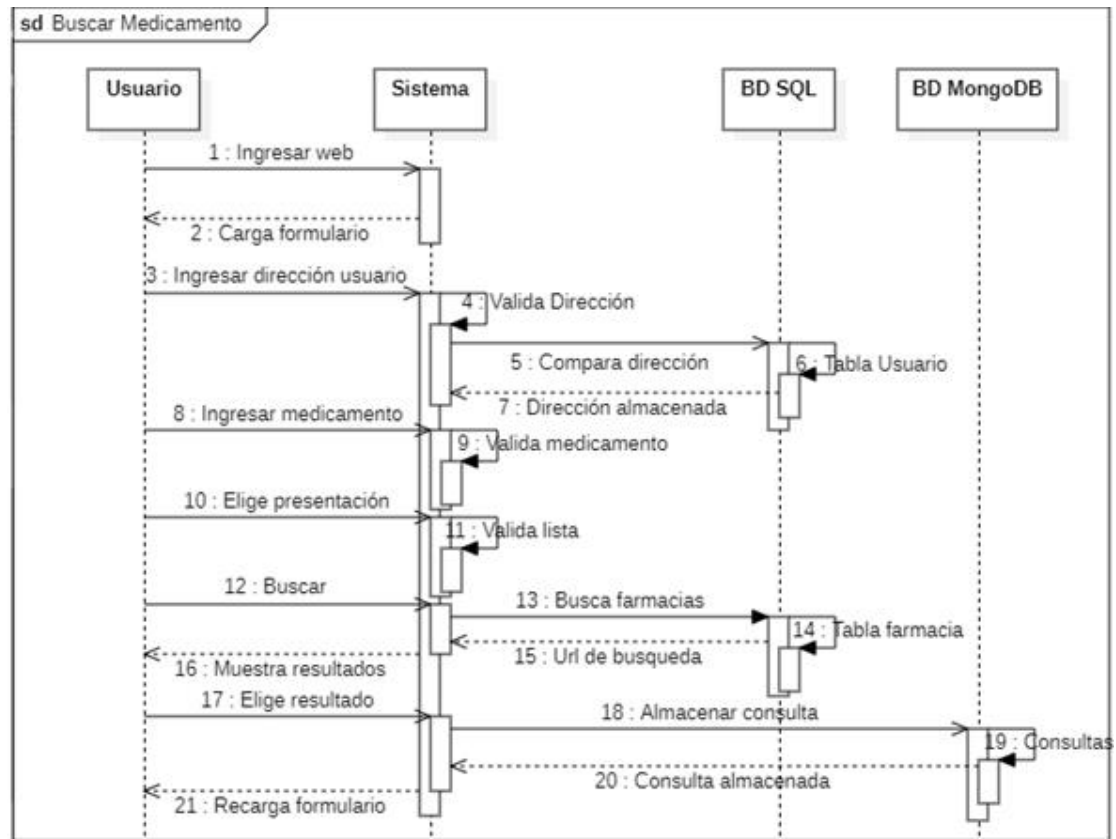
<b>RNF 003</b>	<b>Usabilidad</b>
<b>Detalle</b>	La plataforma debe poseer una interfaz sencilla e intuitiva para el usuario.
<b>Actividad</b>	<ol style="list-style-type: none"> <li>1. Velocidad para la ejecución.</li> <li>2. Fácil de usar y aprender.</li> <li>3. Respuesta inmediata para las modificaciones.</li> <li>4. En caso de errores el sistema genera mensaje claros e informativos.</li> </ol>

<b>RNF 004</b>	<b>Configurar Mapa</b>
<b>Detalle</b>	La plataforma debe contar con la API Google Maps.
<b>Actividad</b>	<ol style="list-style-type: none"> <li>1. Crear usuario API.</li> <li>2. Obtener key de conexión.</li> <li>3. Configurar el puerto HTTPS.</li> <li>4. Configurar visualización en el navegador Chrome.</li> </ol>

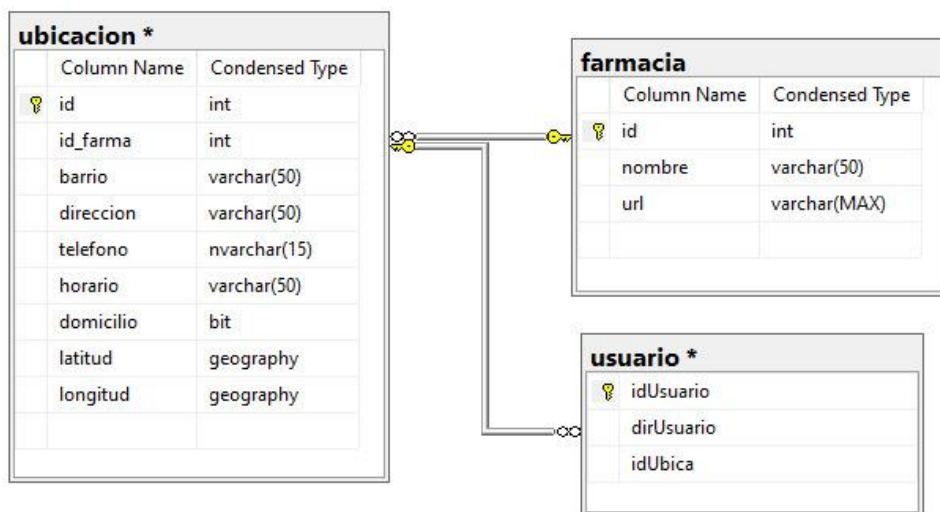
## CASO DE USO



## DIAGRAMA DE SECUENCIAS



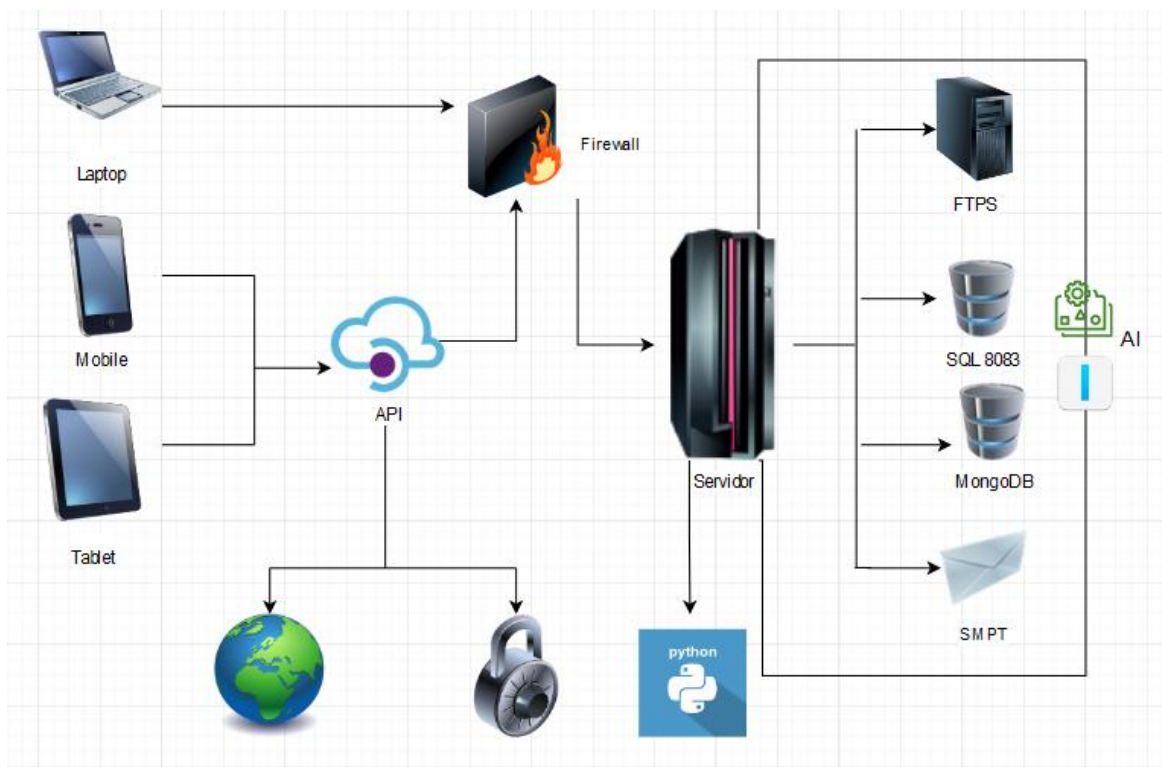
## MODELO RELACIONAL



## COLECCIÓN MONGODB

```
} _id: ObjectId('642dfb333914157eaaa577a6')  
  producto: "Dolex 500"  
  Precio: 1500  
  ubica: "Farmatodo"  
  fecha: "2023/04/05"  
  usuario: "Calle 15 12a-10"
```

## DIAGRAMA DE ARQUITECTURA



## MOCKUP PRINCIPAL

# FARMA-APP



Tu Dirección

Nombre del Medicamento



Presentación del Medicamento

**BUSCAR**

## MOCKUP DE LA BÚSQUEDA

# FARMA-APP



Medicamento

### Resultado de la Búsqueda

Farmacia

Presentación

Precio

Dirección

Link


## Stack

### LENGUAJE DE PROGRAMACIÓN:

- Python: es un lenguaje de programación de alto nivel y fácil de aprender que se utiliza ampliamente en proyectos de web scraping debido a sus bibliotecas integradas para el procesamiento de datos y la automatización de tareas.

## LIBRERÍAS

- BeautifulSoup: es una biblioteca de Python utilizada para extraer datos de HTML y XML.
- Selenium: es una herramienta de automatización de pruebas que se utiliza comúnmente para el web scraping.
- Pandas: es una biblioteca de Python utilizada para el análisis de datos.
- AVMSpeechMath
- speech\_recognition
- basemap
- geopy

## GESTOR DE BASE DE DATOS

- MongoDB: es una base de datos NoSQL popular para almacenar datos de webscraping. Es escalable, flexible y fácil de usar para proyectos de webscraping.
- MySQL: es un sistema de administración de bases de datos relacionales. Es un software de código abierto desarrollado por Oracle.

## FRAMEWORKS

- Scrapy: es un framework de Python diseñado para el web scraping.
- Flask: es un framework de Python para crear aplicaciones web.

## CONTROL DE VERSIONES

- GitHub