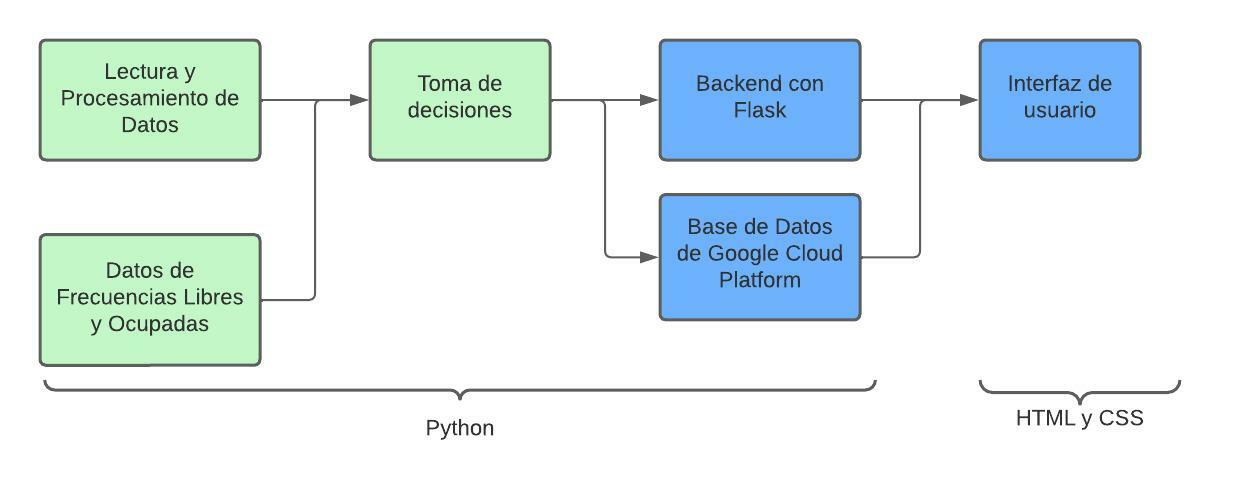
**3.1 Metodología**

Para el desarrollo de los algoritmos de este proyecto de titulación se ha dividido en dos partes como se puede ver en la figura 3.1, la primera parte (recuadros verdes) consta del desarrollo del algoritmo de lectura y procesamiento de datos, además de la toma de decisiones si la señal es legal o ilegal, para esta parte se empleará solamente Python y sus librerías ya que este lenguaje esta optimizado para el procesamiento de grandes cantidades de datos. Par este algoritmo se han desarrollado varias funciones que conforman la librería SDRSCANFM y SDRSCANTV creadas especialmente para este proyecto.

La segunda parte (recuadros azules) del desarrollo del código comprende a la creación del Backend y Frontend de la aplicación web que manejara el usuario. Para el backend se ha empleado el framework de Python Flask, ya que es un framework fácil de implementar y de desplegar en el sistema. Para el Frontend se ha empleado HTML y CSS, lenguajes ampliamente utilizados para el desarrollo de interfaces web de usuario. 

Como se puede ver en la figura 3.1 los 5 primeros bloques de la parte izquierda están desarrollados con Python e interconectados entre sí, es decir para poder tomar la decisión de si la señal es legal o ilegal es necesario los bloques anteriores de lectura y procesamiento de datos y el bloque en donde se encuentra la información de las frecuencias libres y ocupadas del espectro radioeléctrico en la región. A su vez el bloque de toma de decisiones envía un archivo JSON con toda la información previa, a los bloques de Backend con Flask y al bloque de la base de datos de Google Cloud Platform en caso de existir una señal ilegal. Finalmente, estos dos últimos bloques se conectan a la interfaz de usuario desarrollada con HTML y CSS donde el usuario puede acceder a las opciones de escaneo del espectro y las alarmas del usuario.

**3.2 Bloque de lectura, procesamiento y toma de decisiones**

Este bloque tiene la mayor cantidad de funciones, creadas específicamente para la lectura del espectro radioeléctrico y manejo de la infraestructura de los datos, para este propósito y con la finalidad de modularizar el código creado, se ha creado las librerías SDRSCANFM y SDRSCANTV, que a continuación se detallan:

**Librería SdrScanFM**

Esta librería se emplea para el procesamiento y análisis del espectro radioeléctrico del servicio de Radio FM, en donde se han creado específicamente funciones que almacenan los datos de acuerdo al ancho de banda del canal FM. A continuación, se detallará todas las funciones implementadas en esta librería.

Primero en el código se tiene que importar todas las librerías necesarias para la lectura y procesamiento de los datos, estas librerías nos permiten un fácil manejo y manipulación de la infraestructura de los datos leídos con el RTL-SDR.

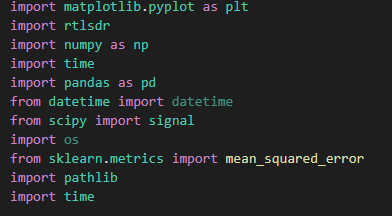


Figura 3.2.1 Librerías importadas

**Función setup:** La primera función creada se muestra en la figura 3.2.2, donde recibe como parámetros de entrada a la frecuencia máxima y mínima que se pretende analizar. Esta función es la encargada de configurar todos los parámetros que se emplearan en la lectura del espectro radioeléctrico como la tasa de muestreo (rate\_best), la cantidad de muestras para calcular el Power Spectral Density (npsd\_res) y la cantidad de veces que se calcula el PSD para su promedio (npsd\_avg), además se configura el tamaño de los arreglos que almacenaran los datos que nos da el RTL-SDR, como por ejemplo las variables, samples, psd\_array, freq\_array y psd\_total. Es importante configurar estos parámetros con los valores exactos de las muestras obtenidas, ya que si el número no es exacto se va a tener un error en las dimensiones de los arrays al momento de almacenar en variables los datos leídos del RTL-SDR. Finalmente, se retorna todas estas variables configuradas, para que se proceda a realizar la lectura del espectro radioeléctrico.



Figura 3.2.2 Función de configuración de parámetros

**Función readsdr:** La segunda función implementada recibe como parámetros las variables configuradas en la función setup, esta función es la encargada de inicializar el objeto sdr (sdr=rtlsdr.RtlSdr( ) ) dentro del código de acuerdo a los parámetros configurados anteriormente, además una vez iniciado el objeto sdr, se puede configurar la ganancia del RTL-SDR con la variable sdr.gain. El objetivo de esta función es realizar una lectura de las muestras obtenidas con el RTL-SDR y procesarlas con la función psd para obtener la Densidad Espectral de Potencia (PSD) a partir de las muestras obtenidas. En la figura 3.2.3 se puede observar este proceso en donde se trabaja con dos bucles for, el primero que es un bucle for anidado primero recorre el arreglo freq, obteniendo los valores de las muestras en cada frecuencia asignada para después volver a realizar este proceso un determinado número y el segundo bucle for trasforma las muestras en valores de la Densidad Espectral de Potencia y la gráfica. El parámetro sdr.close nos permite cerrar el objeto sdr, para poder realizar otra lectura con el mismo objeto. Es importante incluir este comando debido a que si no se cierra el objeto el RTL-SDR seguirá activado y no podrá realizar nuevas lecturas. Finalmente, todos estos valores son almacenados, procesados y retornados en el dataframe data.

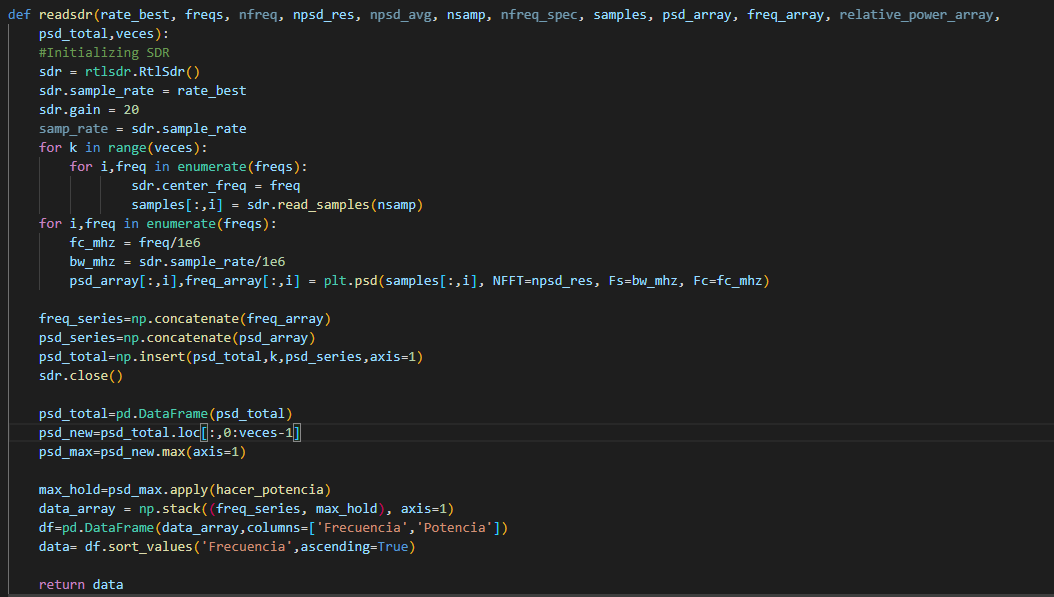


Figura 3.2.3 Función de lectura del espectro radioeléctrico

**Función hacer\_potencia:** En la función anterior se puede observar que los datos obtenidos son procesados por una función hacer\_potencia, esta función permite transformar los datos de PSD obtenidos a decibelios (dBm). Dentro de la función esta la fórmula que permite esta transformación.

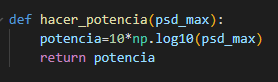


Figura 3.2.4 Función de trasformación de potencia

**Funcion canal\_filter:** Esta función recibe como parámetros los datos obtenidos de la función readsdr, además de los valores de la frecuencia máxima y mínima del canal. Esta función tiene como objetivo filtrar los datos de cada canal a analizar y almacenarlos en la variable data\_canal. Este proceso es importante ya que nos permite analizar canal por canal, y no el conjunto de datos entero, generando menos carga computacional y menos tiempo de procesamiento del algoritmo.

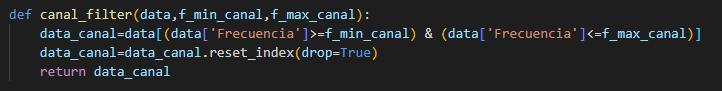


Figura 3.2.5 Función de filtrado de canal

**Función minima\_senal\_detectable:** Esta función recibe como parámetros a los datos por canal obtenidos en la lectura del espectro radioeléctrico, y las procesa con la función detection\_limit, esto con el fin de crear una señal de referencia que es la señal que se va a comparar para la toma de decisiones posteriores en el código.

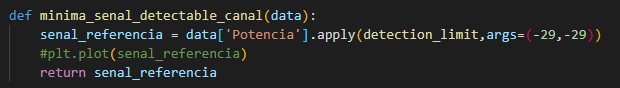
****

Figura 3.2.6 Función de mínima señal detectable

**Función detection\_limit:** Esta función es creada específicamente para la función minima\_senal\_detectable\_canal, donde los parámetros de entrada en esta función, es cada valor de potencia dentro del dataframe data\_canal y el umbral corresponde al valor de la potencia mínima que debe tener el canal para que sus valores de potencia no sean cambiados. Es decir, esta función permite un alisamiento del espectro del canal a partir del valor del umbral, como se ve en el código si el valor de la potencia es menor al umbral este valor se convierte en el valor de la constante y en caso contrario el valor de la potencia no cambia. El objetivo de esta función es eliminar el ruido dentro del canal ya que este ruido no permite una correcta comparación de las señales.

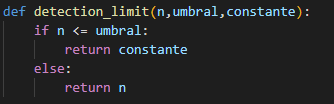
****

Figura 3.2.7 Función detection\_limit

**Función crear\_senal\_comparacion:** Esta función toma como entrada a la señal de referencia y el umbral de la mínima señal detectable. El propósito de esta función es crear una señal que permita la comparación estadística con la señal de referencia y determinar si la señal es legal o ilegal. Para esto se necesita que la señal de comparación tenga las mismas dimensiones que la señal de referencia y que los valores dentro de esta señal creada sean los valores del umbral que anteriormente definimos.

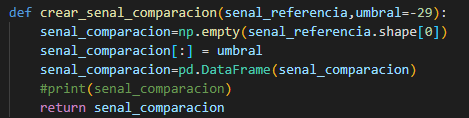
****

Figura 3.2.8 Función de creación de señal a comparar

**Función comparación\_senales:** Esta función toma como entrada a las señales que van a compararse para tomar la decisión si existe una señal ilegal o no. Lo primero que se realiza es una comparación de dimensiones de los dataframes a comparar y en caso de no ser de la misma dimensión se los convierte a la misma dimensión. La función comparación implícita en el código de esta función nos da los valores de la correlación y el rmse. Finalmente se devuelven los valores tanto de la correlación como del rmse.

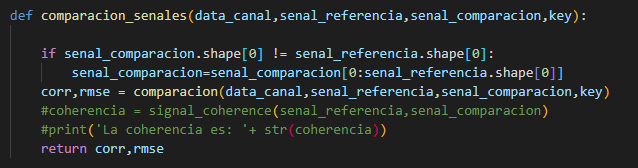
****

Figura 3.2.9 Función de comparación de señales

**Función comparación:** Esta función es la encargada de calcular los valores de la correlación y el rmse de las señales de referencia y comparación, para este proceso se ha implementado una validación de la correlación debido a que si las señales no varían entre si el valor de la correlación va a ser indeterminado originando errores en el código. Además, para el rmse se ha implementado una validación en donde se calcula este valor múltiples veces y se toma el mínimo. Estas validaciones permiten que la comparación de las señales sea más robusta dando así menos errores al momento de tomar decisiones.

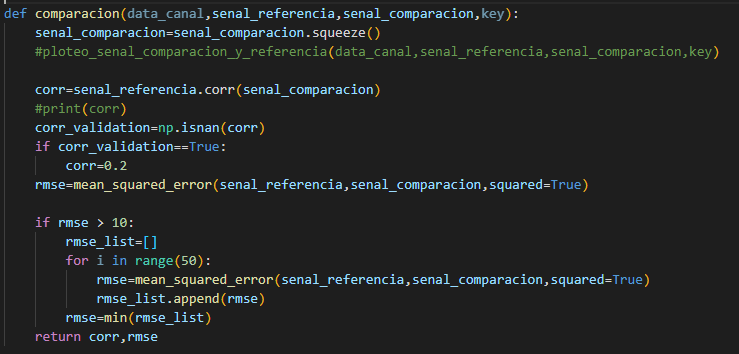
****

Figura 3.2.10 Función comparación

**Función procesamiento:** La función procesamiento es la función principal del algoritmo, donde se ejecutan todas las demás funciones anteriormente descritas, y almacena los valores de la correlación y rmse por canal, posteriormente esta función ejecuta la comparación para determinar si la señal es deseada o no. Para esto se utiliza el condicional if, en donde si la correlación es menor a 0.25 y el rmse es mayor a 0.01 el algoritmo determina que la señal es no deseada y procede a almacenar la información de la potencia y frecuencia en un diccionario para posteriormente retornar esta variable. Además, esta función crea la gráfica del canal analizado y de la frecuencia ilegal en caso de existir. Esta grafica es almacenada en un directorio del proyecto y es empleada para mostrarla en el Frontend de la aplicación web. Finalmente, el diccionario que se retorna tiene los valores de los datos de la potencia espectro radioeléctrico, la señal no deseada y la decisión si existe o no señal ilegal. Retornando el valor de 0 si no existe señal ilegal y 1 si existe señal ilegal.

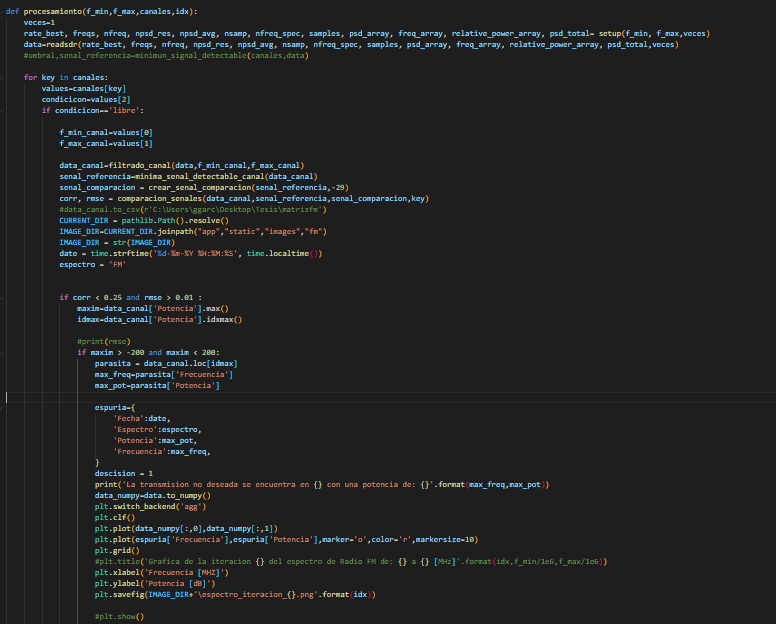


Figura 3.2.11 Función procesamiento

**Función procesamiento\_diccionarios:** Esta función toma como entrada el diccionario retornado de la función procesamiento y devuelve una estructura de diccionario apta para cargar estos datos en la base de datos de Google Cloud Platform.

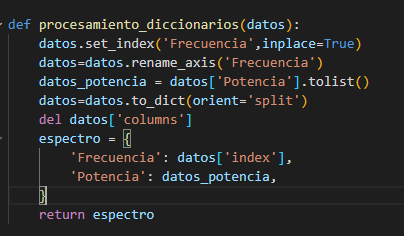


Figura 3.2.12 Función de procesamiento de diccionarios

**Procesamiento de los canales FM en el algoritmo**

Para un correcto procesamiento de los canales FM que están siendo empleados es necesario especificar que canales están libres y cuales están ocupados, con esto, es posible definir que el algoritmo analice aquellos canales en donde no existen transmisiones legales de estaciones de radio FM. El algoritmo desarrollado procesa la información de todo el espectro radioeléctrico FM en 5 iteraciones, con 20 canales cada una, esta división se realiza con el objetivo de dividir en pequeñas partes el espectro FM y procesarlas con las funciones desarrolladas disminuyendo la complejidad computacional en el algoritmo. Además, la representación gráfica de 20 canales FM es más clara que la representación del espectro FM en su totalidad. Con estas consideraciones en las siguientes tablas se especifica las frecuencias máximas y mínimas que se han tomado de los canales para el desarrollo del algoritmo de detección.

**Primera iteración:**

Frecuencia Mínima: 88 MHz

Frecuencia Máxima: 92 MHz

\begin{table}[H]

\centering

\caption{Canales correspondientes a la primera iteración del análisis del espectro FM }

\label{tab:150}

\begin{tabular}{cccc}

\hline

\textit{Canal FM} & Frecuencia máxima {[}MHz{]} & Frecuencia mínima {[}MHz{]} & \begin{tabular}[c]{@{}c@{}}Disponibilidad\end{tabular} \\ \hline

canal 1 &88 &88.19 &usado

\\

canal 2 &88.2 &88.39 &libre

\\

canal 3 &88.4 &88.59 &usado

\\

canal 4 &88.625 &88.79 &libre

\\

canal 5 &88.8 &88.99 &libre

\\

canal 6 &89 &89.19 &libre

\\

canal 7 &89.25 &89.39 &libre

\\

canal 8 &89.4 &89.59 &libre

\\

canal 9 &89.6 &89.79 &usado

\\

canal 10 &89.85 &89.9999 &libre

\\

canal 11 &90 &90.19 &usado

\\

canal 12 &90.2 &90.39 &libre

\\

canal 13 &90.4 &90.59 &usado

\\

canal 14 &90.6 &90.79 &libre

\\

canal 15 &90.8 &90.99 &usado

\\

canal 16 &91 &91.19 &libre

\\

canal 17 &91.2 &91.39 &usado

\\

canal 18 &91.4 &91.59 &libre

\\

canal 19 &91.6 &91.79 &usado

\\

canal 20 &91.8 &91.95 &libre

\end{tabular}

\end{table}

|  |  |  |  |
| --- | --- | --- | --- |
| Canal FM | Frecuencia Maxima (MHz) | Frecuencia Minima (MHz) | Disponibilidad |
| canal 1 | 88 | 88.19 | usado |
| canal 2 | 88.2 | 88.39 | libre |
| canal 3 | 88.4 | 88.59 | usado |
| canal 4 | 88.625 | 88.79 | libre |
| canal 5 | 88.8 | 88.99 | libre |
| canal 6 | 89 | 89.19 | libre |
| canal 7 | 89.25 | 89.39 | libre |
| canal 8 | 89.4 | 89.59 | libre |
| canal 9 | 89.6 | 89.79 | usado |
| canal 10 | 89.85 | 89.9999 | libre |
| canal 11 | 90 | 90.19 | usado |
| canal 12 | 90.2 | 90.39 | libre |
| canal 13 | 90.4 | 90.59 | usado |
| canal 14 | 90.6 | 90.79 | libre |
| canal 15 | 90.8 | 90.99 | usado |
| canal 16 | 91 | 91.19 | libre |
| canal 17 | 91.2 | 91.39 | usado |
| canal 18 | 91.4 | 91.59 | libre |
| canal 19 | 91.6 | 91.79 | usado |
| canal 20 | 91.8 | 91.95 | libre |

**Segunda Iteración:**

Frecuencia Mínima: 92 MHz

Frecuencia Máxima: 96 MHz

|  |  |  |  |
| --- | --- | --- | --- |
| canal 21 | 92 | 92.19 | usado |
| canal 22 | 92.25 | 92.39 | usado |
| canal 23 | 92.4 | 92.59 | usado |
| canal 24 | 92.69 | 92.79 | libre |
| canal 25 | 92.8 | 92.99 | libre |
| canal 26 | 93 | 93.15 | libre |
| canal 27 | 93.2 | 93.39 | usado |
| canal 28 | 93.45 | 93.59 | libre |
| canal 29 | 93.6 | 93.79 | usado |
| canal 30 | 93.8 | 93.95 | libre |
| canal 31 | 94 | 94.19 | usado |
| canal 32 | 94.2 | 94.39 | libre |
| canal 33 | 94.4 | 94.59 | usado |
| canal 34 | 94.6 | 94.79 | libre |
| canal 35 | 94.8 | 94.99 | usado |
| canal 36 | 95 | 95.19 | libre |
| canal 37 | 95.2 | 95.39 | usado |
| canal 38 | 95.4 | 95.59 | libre |
| canal 39 | 95.6 | 95.79 | usado |
| canal 40 | 95.8 | 95.99 | libre |

**Tercera Iteración:**

Frecuencia Mínima: 96 MHz

Frecuencia Máxima: 100 MHz

|  |  |  |  |
| --- | --- | --- | --- |
| canal 41 | 96 | 96.19 | usado |
| canal 42 | 96.27 | 96.39 | libre |
| canal 43 | 96.4 | 96.59 | usado |
| canal 44 | 96.6 | 96.79 | libre |
| canal 45 | 96.8 | 96.99 | usado |
| canal 46 | 97 | 97.19 | libre |
| canal 47 | 97.2 | 97.39 | usado |
| canal 48 | 97.4 | 97.59 | libre |
| canal 49 | 97.6 | 97.79 | usado |
| canal 50 | 97.8 | 97.94 | libre |
| canal 51 | 98 | 98.19 | usado |
| canal 52 | 98.225 | 98.39 | libre |
| canal 53 | 98.4 | 98.59 | usado |
| canal 54 | 98.6 | 98.79 | libre |
| canal 55 | 98.8 | 98.99 | usado |
| canal 56 | 99.025 | 99.19 | libre |
| canal 57 | 99.2 | 99.39 | usado |
| canal 58 | 99.425 | 99.59 | libre |
| canal 59 | 99.6 | 99.79 | usado |
| canal 60 | 99.8 | 99.99 | libre |

**Cuarta Iteración:**

Frecuencia Mínima: 100 MHz

Frecuencia Máxima: 104 MHz

|  |  |  |  |
| --- | --- | --- | --- |
| canal 61 | 100 | 100.19 | usado |
| canal 62 | 100.2 | 100.39 | libre |
| canal 63 | 100.4 | 100.59 | usado |
| canal 64 | 100.65 | 100.79 | libre |
| canal 65 | 100.8 | 100.99 | libre |
| canal 66 | 101 | 101.19 | libre |
| canal 67 | 101.2 | 101.39 | usado |
| canal 68 | 101.4 | 101.59 | libre |
| canal 69 | 101.6 | 101.79 | usado |
| canal 70 | 101.8 | 101.99 | libre |
| canal 71 | 102 | 102.19 | usado |
| canal 72 | 102.2 | 102.39 | libre |
| canal 73 | 102.4 | 102.59 | usado |
| canal 74 | 102.6 | 102.79 | libre |
| canal 75 | 102.8 | 102.99 | usado |
| canal 76 | 103 | 103.19 | libre |
| canal 77 | 103.2 | 103.39 | usado |
| canal 78 | 103.41 | 103.59 | libre |
| canal 79 | 103.6 | 103.79 | usado |
| canal 80 | 103.8 | 103.99 | libre |

**Quinta Iteración:**

Frecuencia Mínima: 104 MHz

Frecuencia Máxima: 108 MHz

|  |  |  |  |
| --- | --- | --- | --- |
| canal 81 | 104 | 104.19 | usado |
| canal 82 | 104.2 | 104.39 | libre |
| canal 83 | 104.4 | 104.59 | usado |
| canal 84 | 104.65 | 104.79 | libre |
| canal 85 | 104.8 | 104.99 | usado |
| canal 86 | 105 | 105.19 | libre |
| canal 87 | 105.2 | 105.39 | usado |
| canal 88 | 105.4 | 105.59 | libre |
| canal 89 | 105.6 | 105.79 | usado |
| canal 90 | 105.8 | 105.99 | libre |
| canal 91 | 106 | 106.19 | usado |
| canal 92 | 106.2 | 106.39 | libre |
| canal 93 | 106.4 | 106.59 | usado |
| canal 94 | 106.6 | 106.79 | libre |
| canal 95 | 106.8 | 106.99 | usado |
| canal 96 | 107 | 107.19 | libre |
| canal 97 | 107.2 | 107.39 | usado |
| canal 98 | 107.4 | 107.59 | libre |
| canal 99 | 107.6 | 107.79 | usado |
| canal 100 | 107.8 | 107.99 | libre |

Para resumir el proceso de lectura, procesamiento y visualización de datos, la siguiente figura muestra el diagrama de bloques que sigue el algoritmo en donde en la parte superior y los bloques en anaranjado muestran la etapa de configuración del dispositivo RTL-SDR y lectura del espectro radioeléctrico. Seguido de la etapa de análisis por canal, expresada en los bloques de color celeste en donde se muestra las etapas de filtrado del canal, mínima señal detectable, creación de la señal a comparar, y finalmente la comparación estadística que nos arroja el resultado de si existe o no una transmisión no deseada, en el último bloque de color morado se muestra la etapa de visualización del espectro radioeléctrico analizado.

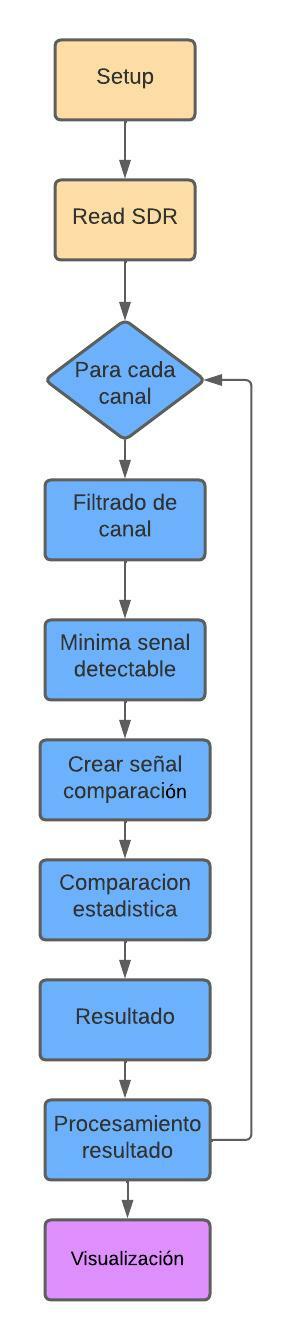


Figura SN. Diagrama de bloques lectura, procesamiento y visualización

**Librería SdrScanTV**

La librería creada para el análisis de las frecuencias de TV, emplea las mismas funciones de la librería SDRSCANFM excepto la función setup, esto debido a que las longitudes de los arreglos para almacenar las muestras leídas difieren entre sí, debido a que existen canales con un mayor ancho de banda y por ende un mayor número de muestras. El cambio de dimensiones se observa en la longitud del arreglo de psd\_total, en donde se multiplica el valor de las filas por un entero para no tener el error de las dimensiones. Este proceso es repetido en las 5 iteraciones que se realiza para analizar el espectro radioeléctrico de televisión, teniendo 5 valores diferentes para cada iteración.



Figura 3.2.13 Ejemplo función setup para primera iteración de TV

**Procesamiento de los canales TV en el algoritmo**

De la misma manera que el espectro radioeléctrico de FM, el espectro de TV se ha dividido en 5 iteraciones de 10 canales cada una. Este proceso es necesario ya que al igual que en FM esta división permite disminuir la complejidad computacional además que en el espectro radioeléctrico de TV, existen saltos de frecuencia considerables entre los espectros radioeléctricos de VHF y UHF, y emplear recursos computacionales en rangos de frecuencias que no son utilizados en el servicio de TV, provocaría que el dispositivo RTL-SDR tarde tiempos considerables para la lectura del espectro. Cabe recalcar que en el servicio de TV se usan dos portadoras, una para el video y otra para el audio, con una distancia de frecuencia considerables entre sí, dejando un espacio libre en el canal. De esta manera las frecuencias a analizar en el algoritmo son asignadas de acuerdo si no están en el rango de las portadoras de audio y video de un canal de TV. Las siguientes tablas muestran con detalle los rangos de frecuencia que el algoritmo analiza y la disponibilidad de estos canales.

**Primera iteración:**

Frecuencia Mínima: 54 MHz

Frecuencia Máxima: 88 MHz

|  |  |  |  |
| --- | --- | --- | --- |
| canal 2 - Libre 1 | 54 | 54.249 | libre |
| canal 2 - Video | 54.25 | 56.25 | usado |
| canal 2 - Libre 2 | 56.251 | 58.49 | libre |
| canal 2 - Armonicos | 58.5 | 59.25 | usado |
| canal 2 - Libre 3 | 59.251 | 59.68 | libre |
| canal 2 - Audio | 59.69 | 59.81 | usado |
| canal 2 - Libre 4 | 59.82 | 60 | libre |
| canal 3 - Libre 1 | 60.001 | 61.149 | libre |
| canal 3 - Armonicos | 61.15 | 61.5 | usado |
| canal 3 - Libre 2 | 61.51 | 66 | libre |
| canal 4 - Libre 1 | 66.001 | 66.749 | libre |
| canal 4 - Video | 66.75 | 68.75 | usado |
| canal 4 - Libre 2 | 68.751 | 70.749 | libre |
| canal 4 - Armonicos | 70.75 | 71.01 | usado |
| canal 4 - Libre 3 | 71.02 | 71.7 | libre |
| canal 4 - Audio | 71.71 | 71.77 | usado |
| canal 4 - Libre 4 | 71.78 | 72 | libre |
| canal 5 - Libre 1 | 76 | 76.249 | libre |
| canal 5 - Video | 76.25 | 78.75 | usado |
| canal 5 - Libre 2 | 78.751 | 80.249 | libre |
| canal 5 - Armonicos | 80.25 | 81.25 | usado |
| canal 5 - Libre 3 | 81.251 | 81.68 | libre |
| canal 5 - Audio | 81.69 | 81.81 | usado |
| canal 5 - Libre 4 | 81.82 | 82 | libre |
| canal 6 - Libre 1 | 82.01 | 85.39 | libre |
| canal 6 - Armonicos | 85.4 | 85.6 | usado |
| canal 6 - Libre 2 | 85.7 | 88 | libre |

**Segunda iteración:**

Frecuencia Mínima: 174 MHz

Frecuencia Máxima: 216 MHz

|  |  |  |  |
| --- | --- | --- | --- |
| canal 7 - Libre 1 | 174 | 180 | usado |
| canal 8 - Libre 1 | 180.01 | 180.49 | libre |
| canal 8 - Video | 180.5 | 182.25 | usado |
| canal 8 - Libre 2 | 182.26 | 184.49 | libre |
| canal 8 - Armonicos | 184.5 | 185.25 | usado |
| canal 8 - Libre 3 | 185.26 | 185.68 | libre |
| canal 8 - Audio | 185.69 | 185.81 | usado |
| canal 8 - Libre 4 | 185.82 | 186 | libre |
| canal 9 - Libre 1 | 186.01 | 186.99 | libre |
| canal 9 - Video | 187 | 187.5 | usado |
| canal 9 - Libre 2 | 187.6 | 190.5 | libre |
| canal 9 - Armonicos | 190.6 | 191 | usado |
| canal 9 - Libre 3 | 191.01 | 191.68 | libre |
| canal 9 - Audio | 191.69 | 191.81 | usado |
| canal 9 - Libre 4 | 191.82 | 192 | libre |
| canal 10 - Libre 1 | 192.01 | 198 | libre |
| canal 11 - Libre 1 | 198.01 | 198.99 | libre |
| canal 11 - Video | 199 | 199.75 | usado |
| canal 11 - Libre 2 | 199.76 | 202.74 | libre |
| canal 11 - Armonicos | 202.75 | 203 | usado |
| canal 11 - Libre 3 | 203.01 | 203.68 | libre |
| canal 11 - Audio | 203.69 | 203.81 | usado |
| canal 11 - Libre 4 | 203.82 | 204 | libre |
| canal 12 - Libre 1 | 204.01 | 210 | libre |
| canal 13 - Libre 1 | 210.01 | 210.749 | usado |
| canal 13 - Video | 210.75 | 212 | usado |
| canal 13 - Libre 2 | 212.01 | 214.6 | usado |
| canal 13 - Armonicos | 214.7 | 215.1 | usado |
| canal 13 - Libre 3 | 215.2 | 215.67 | usado |
| canal 13 - Audio | 215.68 | 215.81 | usado |
| canal 13 - Libre 4 | 215.82 | 216 | usado |

**Tercera iteración:**

Frecuencia Mínima: 500 MHz

Frecuencia Máxima: 572 MHz

|  |  |  |  |
| --- | --- | --- | --- |
| canal 19 - Libre 1 | 500 | 500.04 | libre |
| canal 19 - Armonicos | 500.05 | 500.25 | usado |
| canal 19 - Libre 2 | 500.26 | 506 | libre |
| canal 20 - Libre 1 | 506.01 | 512 | libre |
| canal 21 - Libre 1 | 512.01 | 512.49 | libre |
| canal 21 - Armonicos | 512.5 | 513.75 | usado |
| canal 21 - Libre 2 | 513.76 | 518 | libre |
| canal 22 - Libre 1 | 518.01 | 518.349 | libre |
| canal 22 - Armonicos | 518.35 | 518.45 | usado |
| canal 22 - Libre 2 | 518.46 | 524 | libre |
| canal 23 - Libre 1 | 524.01 | 524.49 | libre |
| canal 23 - Video | 524.5 | 526.5 | usado |
| canal 23 - Libre 2 | 526.6 | 528.49 | libre |
| canal 23 - Armonicos | 528.5 | 529.125 | usado |
| canal 23 - Libre 3 | 529.13 | 529.6 | libre |
| canal 23 - Audio | 529.625 | 529.85 | usado |
| canal 23 - Libre 4 | 529.86 | 530 | libre |
| canal 24 - Libre 1 | 530.01 | 532.774 | libre |
| canal 24 - Armonicos | 532.775 | 532.85 | usado |
| canal 24 - Libre 2 | 532.86 | 536 | libre |
| canal 25 - Libre 1 | 536.01 | 536.249 | libre |
| canal 25 - Video | 536.25 | 538.25 | usado |
| canal 25 - Libre 2 | 538.26 | 540.49 | libre |
| canal 25 - Armonicos | 540.5 | 541.25 | usado |
| canal 25 - Libre 3 | 541.26 | 541.64 | libre |
| canal 25 - Audio | 541.65 | 541.825 | usado |
| canal 25 - Libre 4 | 541.83 | 542 | libre |
| canal 26 - Libre 1 | 542.01 | 547.174 | libre |
| canal 26 - Armonicos | 547.175 | 547.225 | usado |
| canal 26 - Libre 2 | 547.226 | 548 | libre |
| canal 27 - Libre 1 | 548.01 | 548.249 | libre |
| canal 27 - Video | 548.25 | 550.5 | usado |
| canal 27 - Libre 2 | 550.6 | 552.49 | libre |
| canal 27 - Armonicos | 552.5 | 553.25 | usado |
| canal 27 - Libre 3 | 553.26 | 553.64 | libre |
| canal 27 - Audio | 553.65 | 553.875 | usado |
| canal 27 - Libre 4 | 553.88 | 554 | libre |
| canal 28 - Libre 1 | 554.01 | 560 | libre |
| canal 29 - Libre 1 | 560.01 | 560.249 | libre |
| canal 29 - Video | 560.25 | 562 | usado |
| canal 29 - Libre 2 | 562.01 | 564.49 | libre |
| canal 29 - Armonicos | 564.5 | 565.05 | usado |
| canal 29 - Libre 3 | 565.06 | 565.67 | libre |
| canal 29 - Audio | 565.68 | 565.81 | usado |
| canal 29 - Libre 4 | 565.82 | 566 | libre |
| canal 30 - Libre 1 | 566.01 | 572 | libre |

**Cuarta iteración:**

Frecuencia Mínima: 572 MHz

Frecuencia Máxima: 624 MHz

|  |  |  |  |
| --- | --- | --- | --- |
| canal 31 - Libre 1 | 572.01 | 572.49 | libre |
| canal 31 - Video | 572.5 | 574 | usado |
| canal 31 - Libre 2 | 574.01 | 575.92 | libre |
| canal 31 - Armonicos | 575.925 | 577.25 | usado |
| canal 31 - Libre 3 | 575.3 | 577.69 | libre |
| canal 31 - Audio | 577.7 | 577.81 | usado |
| canal 31 - Libre 4 | 577.82 | 578 | libre |
| canal 32 - Libre 1 | 578.01 | 584 | libre |
| canal 33 - Libre 1 | 584.01 | 584.49 | libre |
| canal 33 - Video | 584.5 | 586 | usado |
| canal 33 - Libre 2 | 586.01 | 588.649 | libre |
| canal 33 - Armonicos | 588.65 | 589.1 | usado |
| canal 33 - Libre 3 | 589.2 | 589.68 | libre |
| canal 33 - Audio | 589.69 | 589.8875 | usado |
| canal 33 - Libre 4 | 589.9 | 590 | libre |
| canal 34 - Libre 1 | 590.01 | 590.34 | libre |
| canal 34 - Armonicos | 590.35 | 590.45 | usado |
| canal 34 - Libre 2 | 590.46 | 596 | libre |
| canal 35 - Libre 1 | 596.01 | 596.24 | libre |
| canal 35 - Video | 596.25 | 598.55 | usado |
| canal 35 - Libre 2 | 598.56 | 599.95 | libre |
| canal 35 - Armonicos | 599.96 | 601.1 | usado |
| canal 35 - Libre 3 | 601.2 | 601.68 | libre |
| canal 35 - Audio | 601.69 | 601.825 | usado |
| canal 35 - Libre 4 | 601.9 | 602 | libre |
| canal 36 - Libre 1 | 602.01 | 608 | libre |
| canal 37 - Libre 1 | 608.01 | 614 | libre |
| canal 38 - Libre 1 | 614.01 | 620 | libre |
| canal 39 - Libre 1 | 620.01 | 623.69 | libre |
| canal 39 - Armonicos | 623.7 | 623.9 | usado |
| canal 39 - Libre 2 | 624 | 624 | libre |

**Quinta iteración:**

Frecuencia Mínima: 626 MHz

Frecuencia Máxima: 686 MHz

|  |  |  |  |
| --- | --- | --- | --- |
| canal 40 - Libre 1 | 626.01 | 632 | libre |
| canal 41 - Libre 1 | 632.01 | 638 | libre |
| canal 42 - Libre 1 | 638.01 | 644 | libre |
| canal 43 - Libre 1 | 644.01 | 650 | libre |
| canal 44 - Libre 1 | 650.01 | 656 | libre |
| canal 45 - Libre 1 | 656.01 | 662 | libre |
| canal 46 - Libre 1 | 662.01 | 668 | libre |
| canal 47 - Libre 1 | 668.01 | 674 | libre |
| canal 48 - Libre 1 | 674.01 | 680 | libre |
| canal 49 - Libre 1 | 680.01 | 686 | libre |

**3.3 Bloque de infraestructura de la aplicación web**

**3.3.1 Backend de la aplicación**

Para el desarrollo de la interfaz web es necesario implementar un backend que se conecte al servidor de la Raspberry Pi y a la base de datos de Google. Para esto el framework Flask nos permite fácilmente crear un backend desde cero e implementar las funciones necesarias para la conexión tanto de la interfaz que se muestra al usuario como de las librerías de escaneo del espectro radioeléctrico.

En la siguiente figura se muestra todas las librerías necesarias para desplegar el backend de la aplicación, como se observa se importan funciones de Flask que permiten una fácil conexión del usuario a la base de datos, así como funciones que permiten la entrada de datos del usuario.

La librería app.firestore\_service permite la conexión del backend con la base de datos de Google, y las liberias anteriormente creadas sdrscanfm y sdrscantv son importadas para que obtener la información del espectro radioeléctrico analizado.

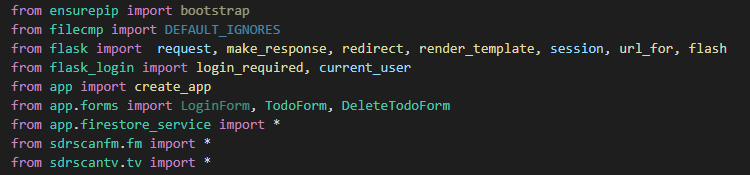


Figura 3.3.1 Librerías importadas para el backend

Flask emplea código de Python para poder ejecutar las funciones de conexión al servidor y a la base de datos, para esto las funciones implementadas se describen a continuación.

**Función index:** Esta función permite obtener la dirección IP del usuario y la almacena dentro de la variable user\_ip, que sirve para crear una sesión de usuario permitiendo navegar en la página con mayor seguridad, ya que la IP del usuario se oculta dentro de la variable session.

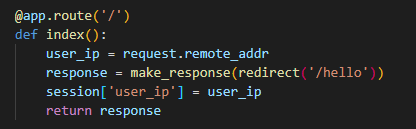
****

Figura 3.3.2 Función index

**Función hello:** Esta función permite la conexión con la página de inicio que se muestra en la aplicación web. En la siguiente figura se muestra que la función acepta los métodos POST y GET, además la función render template permite renderizar una página web creada con HTML, que en ese caso es hello.html, también se envía la variable contexto que contiene toda la información del usuario.

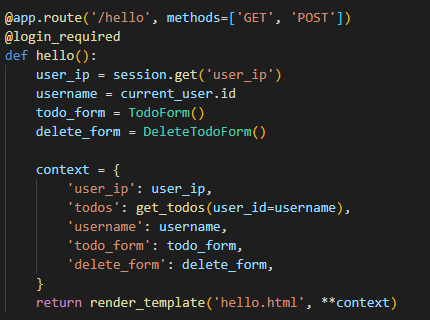
****

Figura 3.3.3 Función de página de inicio

**Función fm:** Esta función permite la renderización de la página que va a dar la orden del escaneo del espectro radioeléctrico FM, partiendo de la información del usuario, se ejecutan las funciones de la librería sdrscanfm en donde por partes se analiza todo el espectro FM y en caso de existir una frecuencia ilegal la almacena en una variable. La variable contexto contiene toda la información sobre el espectro analizado y envía a la página HTML que muestra al usuario si existe o no una frecuencia ilegal. Finalmente, como se puede se puede ver en la figura 3.3.5 en caso de existir frecuencias ilegales, se ejecuta la función put\_alarma que envía los datos de la transmisión ilegal a la base de datos.

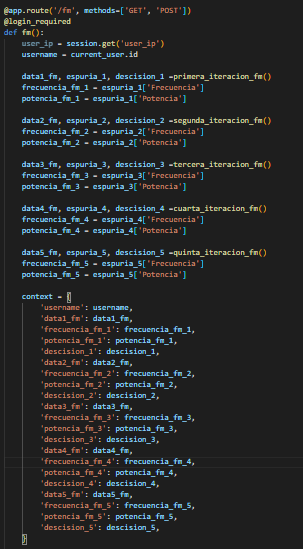
****

Figura 3.3.4 Función de página de escaneo FM

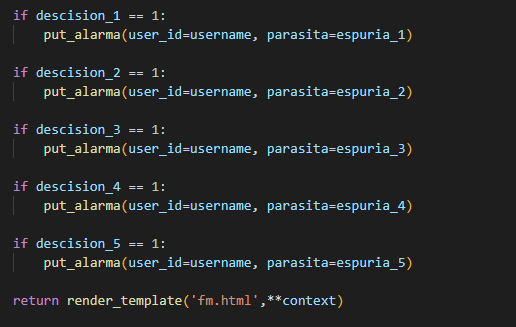


Figura 3.3.5 Asignación de alarmas dentro de la página FM

**Función tv\_vhf:** Esta función permite la renderización de la página que va a dar la orden del escaneo del espectro radioeléctrico TV VHF, usando la librería sdrscantv se escanea el espectro TV VHF en dos iteraciones, en caso de existir una transmisión ilegal los datos son almacenados en la variable contexto y enviados a la página HTML de TV VHF. Al igual que la función de fm, en caso de existir una transmisión ilegal la función put\_alarma envía la información de la transmisión ilegal a la base de datos.

****

Figura 3.3.6 Función tv\_vhf

**Función tv\_uhf:** Como la función anteriormente descrita para tv uhf, esta función permite la renderización de la página que escanea el espectro radioeléctrico de TV UHF, las funciones ejecutadas permiten el análisis del espectro y en caso de existir una transmisión ilegal, almacena los datos y los envía a la base de datos de Google.

****

Figura 3.3.6 Función tv\_uhf

**3.4 Frontend de la aplicación**

Para el desarrollo de la interfaz del usuario se emplea lenguaje HTML y CSS. El proyecto está dividido en varios archivos html que modularizan el código y permiten un fácil despliegue de la aplicación web. A continuación, se analizan los archivos html que despliegan la interfaz de usuario.

**Archivo base.html:** Este archivo es replicado en toda la interfaz de usuario, y contiene la información del encabezado y mensajes para el usuario. Este archivo se ha creado con el fin de tener menos líneas de código para la interfaz ya que se repite en todas las demás paginas html creadas.

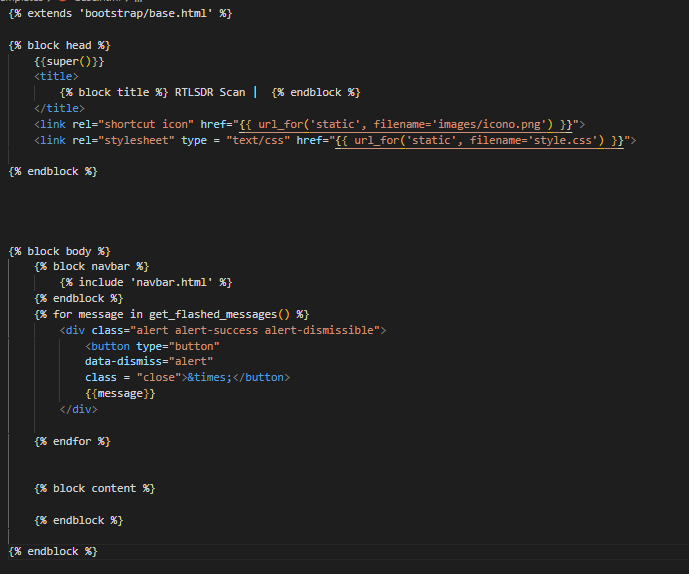
****

Figura 3.4.1 Archivo base.html

**Archivo signup.html:** Este archivo html muestra la página de registro del usuario dentro de la aplicación, como se ve en la siguiente figura, el archivo extiende o trae el archivo base.html y crea un formulario en donde la información del nuevo usuario es enviada a la base de datos, para posteriores inicios de sesión del mismo usuario.

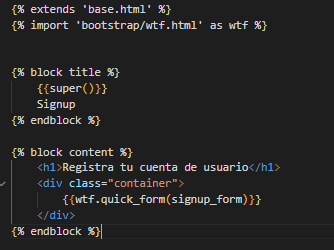
****

Figura 3.4.2 Archivo signup.html

**Archivo login.html:** Este archivo es el encargado de mostrar la página de inicio de sesión del usuario. Para esto se implementa un formulario en donde el usuario digita su nombre de usuario y la contraseña y posteriormente se compara estos valores con los valores de la base de datos, en caso que el usuario no haya digitado correctamente sus credenciales la página retornara un mensaje de error de inicio de sesión. Tal como en el archivo signup.html se trae el archivo base.html y se importa la librería que crea los formularios para que el usuario digite sus credenciales.

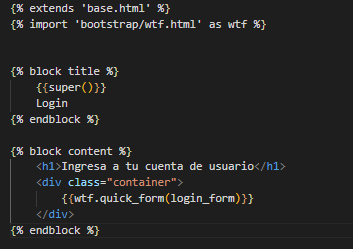
****

Figura 3.4.3 Archivo login.html

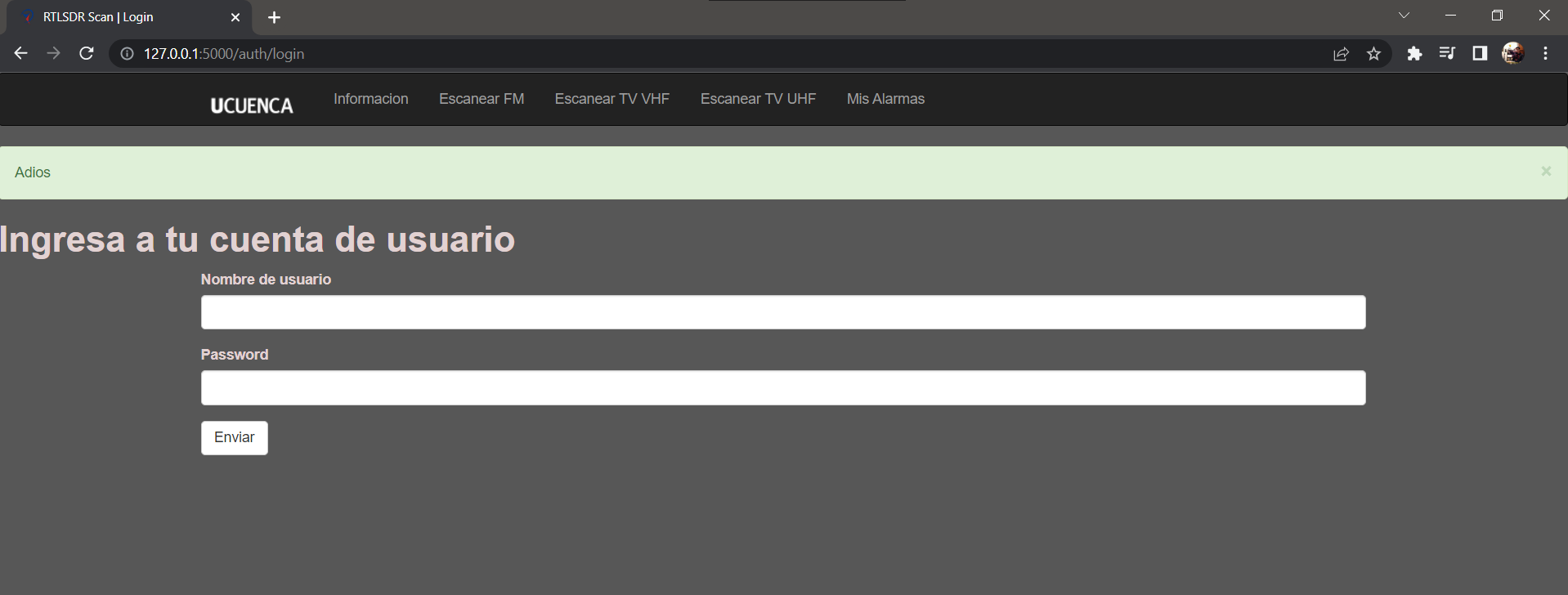


Figura 3.4.3.1 Página login.html

**Archivo hello.html:** Este archivo muestra la página de inicio de la aplicación, en donde el usuario encontrara la información necesaria para utilizar el escaneo del espectro tanto de FM como de TV.

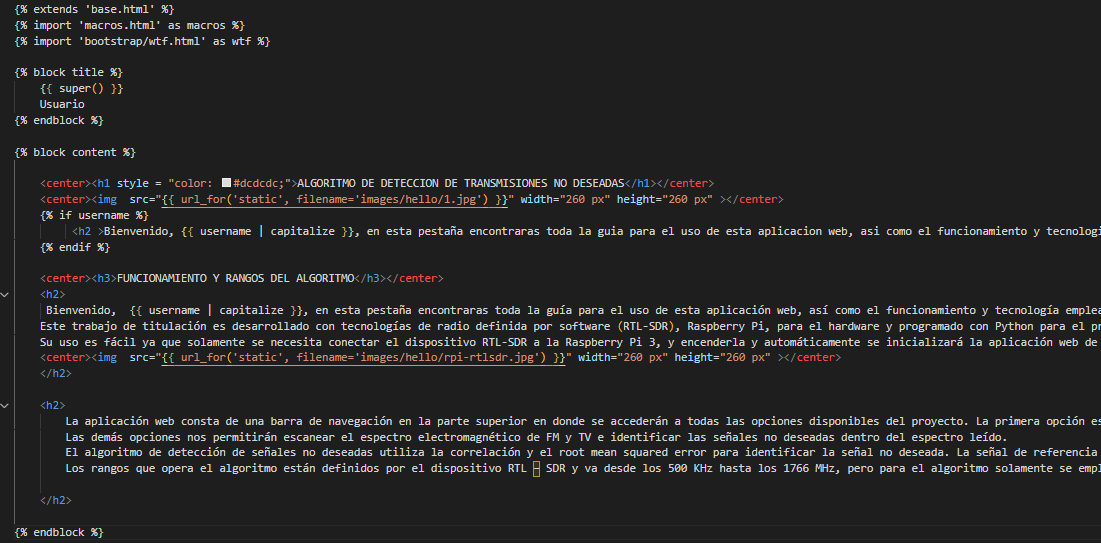
****

Figura 3.4.4 Archivo hello.html



Figura 3.4.4.1 Página hello.html

**Archivo fm.html:** Este archivo muestra al usuario toda la información necesaria del escaneo FM, como se puede ver en la figura siguiente existe un condicional en donde en caso de existir una transmisión ilegal se mostrará la gráfica del espectro analizado y también se muestra una alerta que se ha detectado la transmisión ilegal.

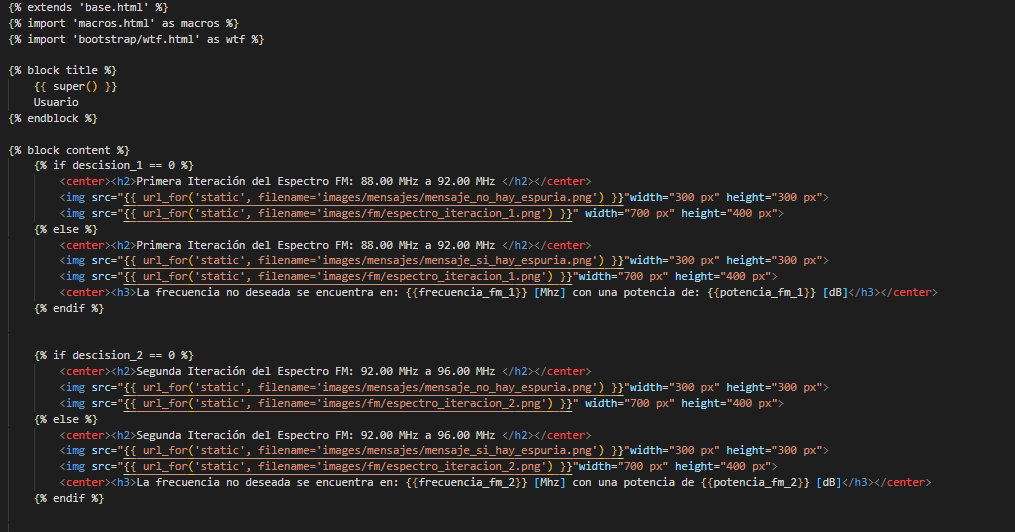
****

Figura 3.4.5 Archivo fm.html

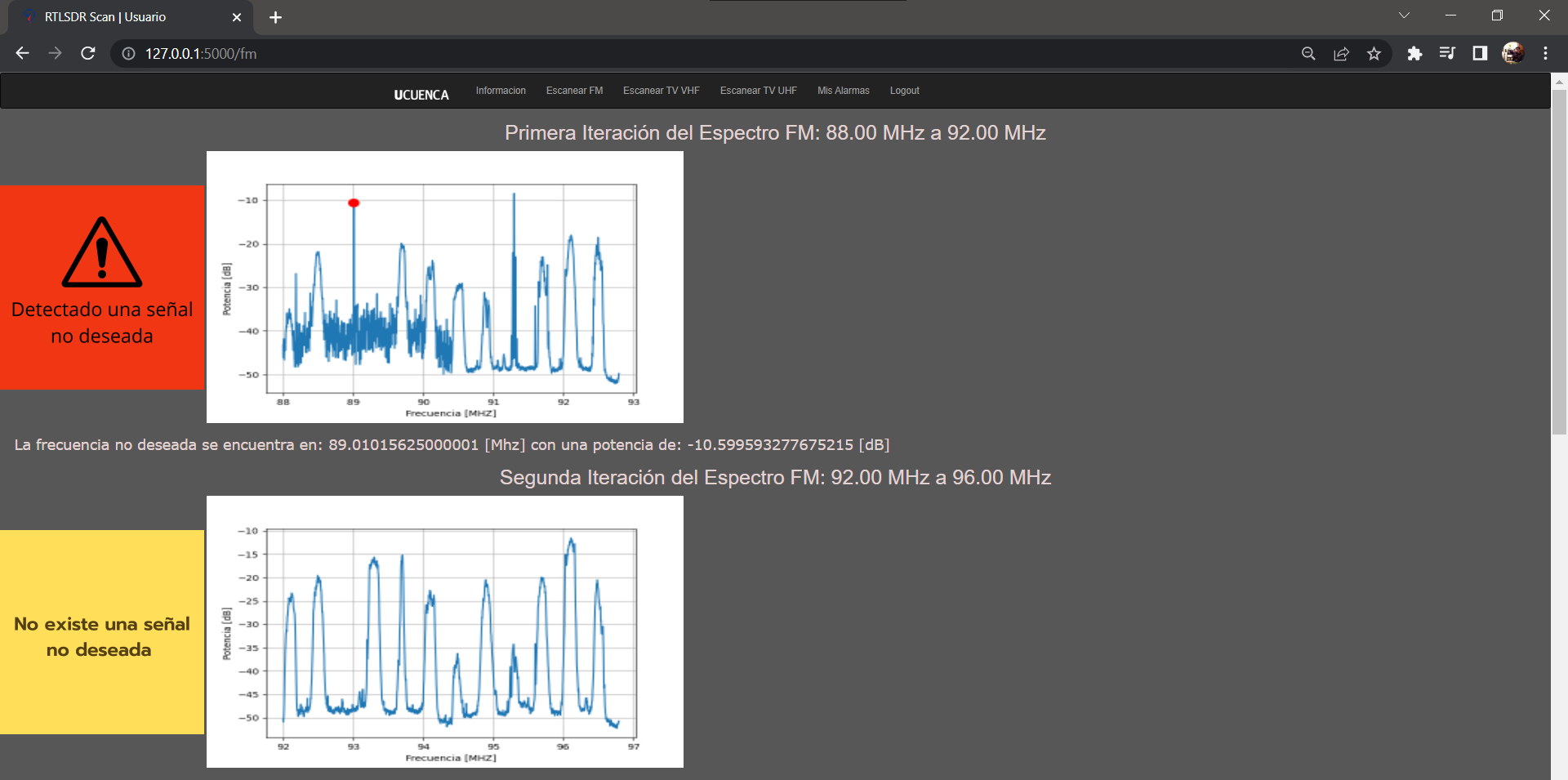


Figura 3.4.5.1 Página fm.html

**Archivo tv.html:** Este archivo muestra la información del escaneo en el espectro radioeléctrico de TV, este archivo tiene la misma estructura que el archivo fm.html, donde se divide en condicionales que permiten mostrar una imagen de alerta en caso de existir una transmisión no deseada.

****

Figura 3.4.6 Archivo tv.html

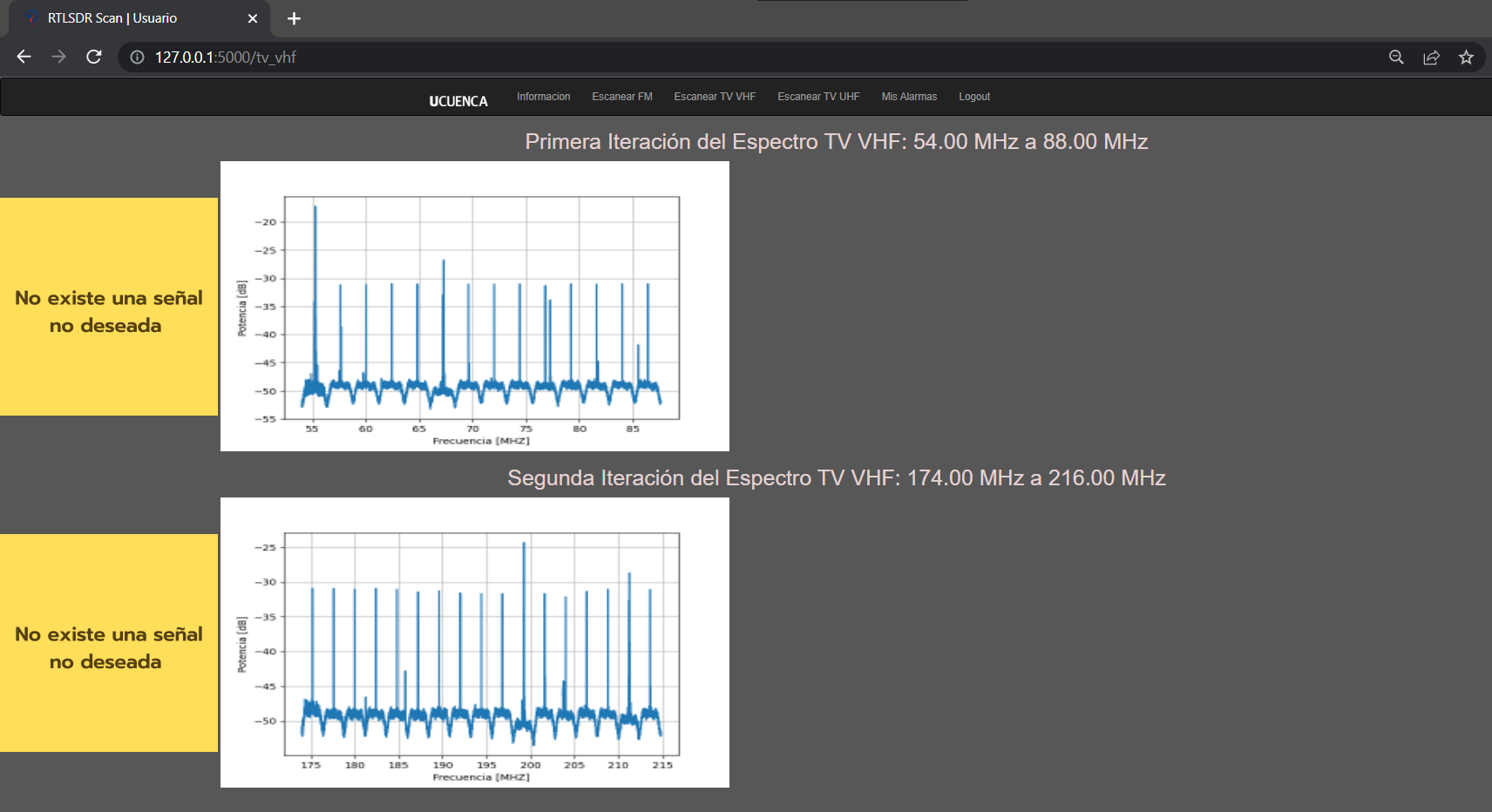


Figura 3.4.6.1 Página tv.html

**Archivo alarmas.html:** Este archivo muestra las alarmas del usuario que se han generado por transmisiones no deseadas. Como se muestra en la figura este archivo accede la variable alarmas y crea una tabla en donde se visualiza la información de la alarma, esta información incluye la frecuencia, potencia, fecha y a qué servicio del espectro radioeléctrico pertenece.

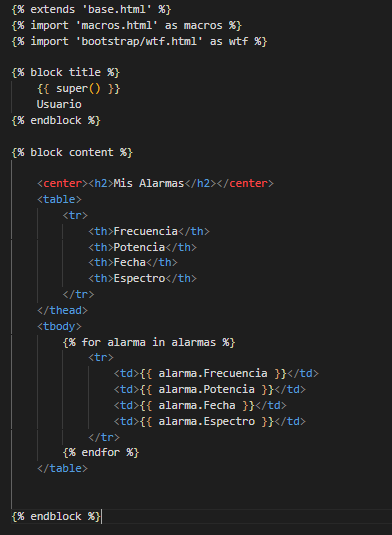
****

Figura 3.4.7 Archivo alarmas.html

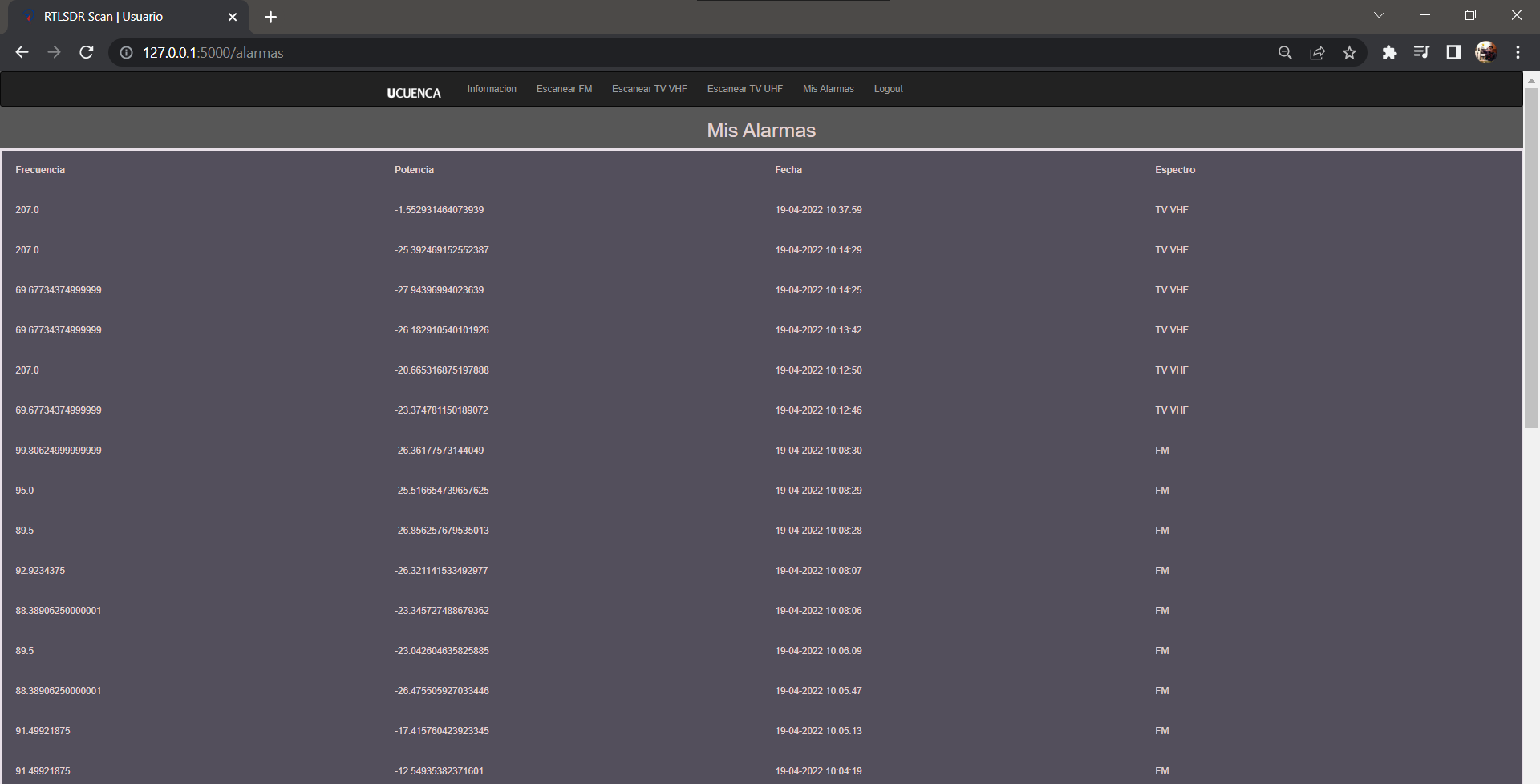
****

Figura 3.4.7.1 Página alarmas.html