El presente trabajo como evaluación final de la materia de Procesamiento de Señales tiene como objetivo aplicar un meta clasificador de señales a un conjunto de señales que se obtienen a partir de los contornos de diferentes hojas de las plantas, en este proceso se utilizaron algoritmos, métodos, pasos o etapas, estos últimos se constituyeron desde una perspectiva personal como si fuesen un método o función que se tenía que codificar para conseguir el objetivo principal. A continuación, se explica de manera puntual todo lo mencionado de manera general:

- Características de la base de datos: La base de datos que se utilizó en el meta clasificador tiene un tamaño de 100x2523, es decir, tiene un total de 100 filas, los cuales representan una señal obtenida del contorno de cada hoja que seleccionó para su procesamiento, por tanto, se ocuparon 100 hojas, que representa las 100 filas de la base de datos, en cuanto a las columnas de la base de datos en total son 2523 columnas, en donde la primera columna corresponde a la clase de esa señal, este dato o las clases se obtuvieron de un archivo (hoja de Excel) que ya estaba definido para esta implementación en particular y las otras 2522 columnas corresponden puramente a los datos de la señal obtenida de los contornos de las hojas seleccionadas para su procesamiento. Vale la pena mencionar también que el formato o tipo de archivo que se optó para salvarlo como archivo local para que el acceso a los datos fuera más práctico y 'óptimo', fue el formato ASCII este no necesariamente fue la única opción que se consideró si no también había otro formato como por ejemplo V-6 y binario, pero por cuestiones prácticas para el acceso a la información de manera directa se optó el formato ASCII en este caso para la base de datos.
- Pasos y etapas que se siguió en el proceso:
    - Se seleccionaron las imágenes de las hojas que se procesaron para obtener las señales a partir de sus contornos.
    - Se crearon los primeros scripts en Octave para poder obtener los archivos (imágenes de las hojas) mencionadas, esto para que se automatizara la lectura de estos archivos y poder manipularlos en memoria y así obtener los contornos de cada imagen.
    - Se crearon funciones como por ejemplo ObtenerArchivos, ObtenerSenial; esto para poder representar y obtener los contornos de cada imagen en forma de señales que formará el conjunto de señales de entrenamiento (TRAIN) que será procesada a través del meta clasificador.
    - Se crearon los scripts y funciones, por ejemplo; ConjuntoSeniales, PromedioLongitudSeniales, este último para poder obtener un promedio de todas las señales generadas; esto dado que cuando se generaba cada señal las longitudes de estos eran diferentes; por ello se estableció que el promedio obtenido de todas las señales fuera el valor de longitud para todas las señales (que las 100 señales) tuvieran una longitud de ese valor promedio.
    - Se formó el conjunto de datos o la base datos de entrenamiento (TRAIN), esto ya agregando su clase correspondiente a cada señal.
    - Se agregaron todos los archivos correspondientes del meta clasificador para poder hacer los experimentos y/o procesar la base de datos generado en el meta clasificador.

- Se implementó la validación cruzada dejando uno fuera para poder obtener el error de clasificación, es decir, se utilizó el algoritmo KNN (K vecinos más cercanos) pero embebido en la función de validación cruzada.
- Se empezaron a buscar las configuraciones 'idóneas' o aceptables para colaborar y/o revisar que el meta clasificador no lanzara algún error al ejecutar o procesar la base de datos generado. Dado que se observó que con las configuraciones planteadas la ejecución era muy costoso o que el tiempo para procesar la base datos generado y mostrar los resultados, era sumamente largo el tiempo de espera o ejecución del meta clasificador para mostrar resultados.
- Una vez verificado que el meta clasificador procesa el conjunto de datos formado (base de datos TRAIN) sin detalle alguno; se procedió a realizar los experimentos con las configuraciones especificados para la evaluación.

- Métodos que se utilizaron:
  - Se implementó la función LeerArchivos que como su nombre lo indica, se utilizó para obtener el directorio y el nombre del archivo (imagen) a leer y cargarse en memoria para obtener su contorno y obtener su señal.
  - Se implementaron las funciones ObtenerArchivos y ObtenerSenial, que contribuyeron a la función LeerArchivos para lograr automatizar y obtener el archivo que se le indica específicamente dentro de las imágenes seleccionadas al inicio para generar u obtener las señales.
  - Se implementaron las funciones PromedioLongitudSeniales y ConjuntoSeniales; los cuales se utilizaron para ir generando el conjunto de señales y así obtener la base de datos (BD_TRAIN).
  - Se implementaron las funciones ObtenerSenialLongitudFinal y JuntarClasesDatos; este último se utilizó para formar el conjunto de datos de entrenamiento (BD_TRAIN), es decir se agregó su clase correspondiente a cada señal obtenido del archivo especificado inicialmente; mientras que la primera función se utilizó para obtener el valor de longitud para todas las señales del conjunto de datos (para las 100 señales); también vale la pena mencionar que en este punto se cargaron en memoria las clases correspondiente a cada una de las señales, esto con ayuda de un archivo .txt.
  -

No. Ejecución	Aptitud	Vector solución
1.	1	[57,1,1]
2.		
3.		
4.		
5.		
6.		
7.		
8.		
9.		
10.		
11.		
12.		
13.		
14.		
15.		
16.		
17.		
18.		
19.		
20.		
21.		
22.		
23.		
24.		
25.		
26.		
27.		
28.		
29.		
30.		