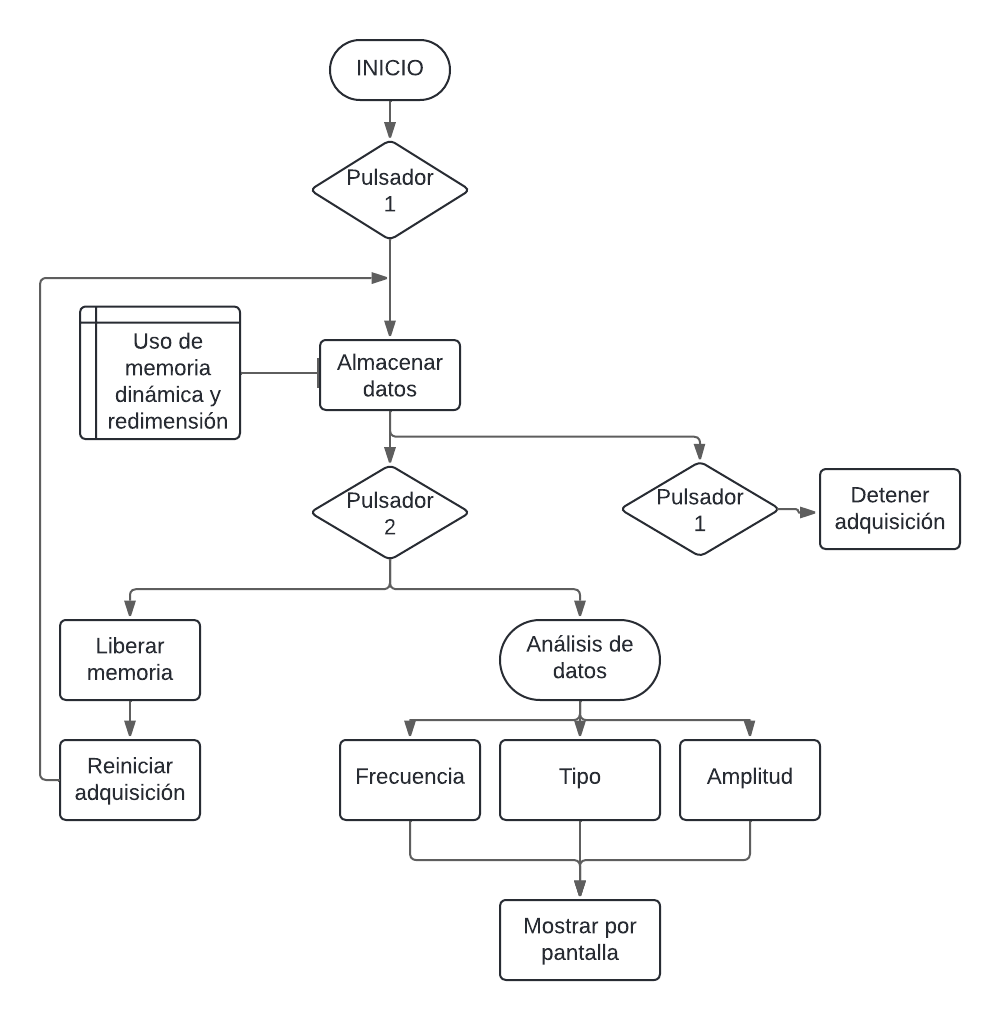
**Contexto del problema**

Se nos pide crear un programa que reciba datos discretos de una onda y devuelve la frecuencia, la amplitud y el tipo de onda. Cuando se enciende el dispositivo, aún no se realiza la adquisición sino hasta que presione un pulsador (1); al presionar (1) se debe iniciar la adquisición de los datos hasta que se presione un segundo pulsador (2); cuando éste sea presionado los datos capturados se deben analizar e imprimir las características requeridas.

Ahora bien, la adquisición de los datos no se detendrá hasta que presionemos nuevamente (1); por eso, al presionar (2), se debe, además de realizar el análisis, liberar inmediatamente la memoria utilizada y comenzar un nuevo periodo de adquisición de datos.

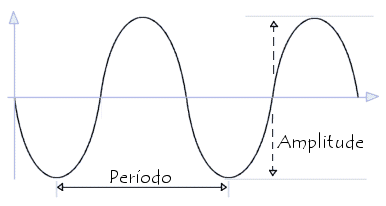
He aquí el diagrama de flujo del programa:



**Análisis del problema**

La principal limitación que se nos presenta es el carácter discreto de los datos y la velocidad del muestreo.

Como partimos del hecho que no conocemos la señal que estamos recibiendo no podemos asumir que para cierto periodo de tiempo la señal se mantendrá igual, aunque sea el escenario ideal. Como hay que cerciorarnos que cada cierto tiempo la onda cumple con unas características, vamos a definir el periodo de una onda como la unidad básica de análisis:



Es decir, durante un periodo completo, los datos deben cumplir con las características de uno de los tipos de onda, si se combina o se cambia alguna de ellas en menos de un periodo, este pedazo de onda será tratada como desconocida.

Ahora empecemos definiendo los diferentes casos de análisis que tenemos.

Caso 1: La señal que leemos se mantiene constante en forma, frecuencia y amplitud

Caso 2: La señal que leemos mantiene su misma forma y variamos su frecuencia y amplitud

Caso 3: La señal que leemos se mantiene contante en su frecuencia y amplitud y cambiamos su forma.

Caso 4: La señal varía todos sus parámetros.

Observaciones:

- Cuando cambiamos una de las variables la señal tarda aproximadamente un periodo en estabilizarse y representar la señal de forma correcta.

- Como estamos limitados por la velocidad del muestreo, la fidelidad de la información que leemos disminuye a medida que aumentamos la frecuencia de nuestra onda.

Declaremos la estructura de datos que usaremos para la recolección de los datos, como estamos usando un ciclo como unidad de medida usaremos un arreglo que almacene los datos de cada ciclo, es decir, una lista de listas. Por otro lado, conocer los intervalos de tiempo a medida de fluctúa la onda es importante para calcular la frecuencia e identificar los ciclos, por ello, usaremos otro arreglo que guarde los tiempos de duración de cada ciclo y estará relacionado uno a uno con las posiciones del arreglo de ciclos. Recordemos que no conocemos la cantidad de datos que vamos a recibir por tanto es necesario usar memoria dinámica para la manipulación de los arreglos.

Sigamos declarando las funciones de necesitamos para el análisis de nuestra onda: como la cantidad de datos es considerable y se deben recorrer todos los datos, mientras menos veces recorramos los datos será mejor.

*redimensionar():* esta función se debe encargar de agrandar nuestro arreglo dinámico a medida que se complete su capacidad inicial

*frecuencia():* esta función debe calcular la frecuencia de la onda, considero usar la función *millis()* para calcular el tiempo en que se completa un ciclo

*amplitud():* esta función debe calcular la amplitud de la onda, considero identificar los máximos y mínimos y dividirlos entre dos

*tipo():* esta función calculará la forma de la onda, como conocemos los tres tipos de ondas que vamos a analizar, un patrón para identificar es la pendiente entre dos puntos consecutivos: para la función cuadrada la pendiente es cero o no existe, para la función triangular la pendiente es igual y para la función sinusoidal la pendiente varía.

Sugerencia: si es posible crear una sola función que analice las tres características, así solo se deberá recorrer los arreglos única vez.