**Desafío 1**

Samuel Gonzalez Hernandez 1027946121

Juan Sebastian Galeano 1006110753

**Análisis y Diseño de la Solución**

**A. Análisis del Problema**

La empresa Informa2 tiene la necesidad de implementar un sistema que permita adquirir y analizar una señal analógica, de manera que se puedan identificar sus características principales como la frecuencia, amplitud y forma de onda. Para esto, se utilizará una plataforma Arduino UNO en conjunto con un generador de señales y una pantalla LCD, que permitirá visualizar los resultados obtenidos en tiempo real.

El proyecto consiste en utilizar dos pulsadores, uno para iniciar y el otro para detener la adquisición de datos, mientras que el procesamiento de la señal permitirá calcular su frecuencia y amplitud, además de identificar la forma de onda (ya sea senoidal, cuadrada, triangular, etc.). La solución se debe implementar en C++, haciendo uso de punteros y arreglos para la gestión eficiente de la memoria.

**Consideraciones para la Solución**

Al momento de plantear la solución, es importante tener en cuenta varios aspectos:

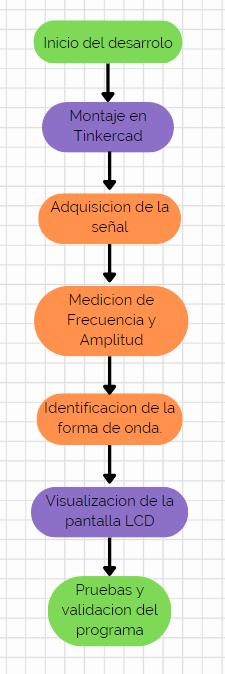
* Simulación del sistema en Tinkercad: Utilizaremos Tinkercad como herramienta de simulación para construir virtualmente el circuito que incluye el Arduino UNO, un generador de señales y una pantalla LCD.

El pulsador será utilizado como medio para controlar la adquisición de datos, permitiendo iniciar y detener el proceso, de ser necesario, se usarán dos pulsadores para facilitar la obtención de datos, dándole utilidad a cada uno de ellos por separado y así evitando confusiones o mal funcionamiento.

* Medición de la frecuencia y amplitud: La frecuencia de la señal se calculará midiendo el tiempo entre dos picos consecutivos. Para esto, se detectarán los cruces de la señal por un cierto umbral y se calculará el tiempo transcurrido entre ellos.

La amplitud se determinará midiendo los valores máximos y mínimos de la señal, calculando la diferencia entre ellos (amplitud pico a pico) y convirtiendo este valor a voltios.

* Identificación de la forma de onda: Los algoritmos deben ser capaces de comparar las características de la señal adquirida con patrones típicos de formas de onda (senoidal, cuadrada, triangular) para identificarlas correctamente. En caso de que no se corresponda con ninguna de las formas predefinidas, el sistema marcará la señal como desconocida.
* Uso de memoria dinámica: Para manejar de forma eficiente la cantidad de muestras de la señal, utilizaremos punteros y arreglos dinámicos, lo que nos permitirá optimizar el uso de la memoria disponible.
* Librerías y componentes adicionales: Se usará la librería Adafruit\_LiquidCrystal.h para gestionar de manera eficiente la pantalla LCD que mostrará las características de la señal procesada.

**B. Esquema de desarrollo de los Algoritmos**

**D. Desafíos y Problemas Potenciales**

Durante el desarrollo es posible que surjan algunos retos. Por ejemplo, es importante calibrar correctamente el umbral de detección de picos para calcular la frecuencia de manera precisa. También hay que considerar el manejo eficiente de la memoria cuando se almacenan muchas muestras de la señal, para evitar sobrecargar el sistema.

Algunos puntos a tener en cuenta para mejorar el desarrollo son:

Realizar pruebas de cada parte del sistema por separado para detectar y solucionar errores antes de integrarlo todo.

Optimizar la adquisición de datos para que el sistema sea capaz de capturar señales de distintas frecuencias sin perder información.

Asegurar que el código esté bien estructurado y comentado, lo que facilitará el mantenimiento y posibles mejoras.

**E. Evolución de la Solución**

Al principio, la idea era crear una solución modular para dividir el problema en partes más pequeñas y manejables. Sin embargo, debido a las limitaciones de Tinkercad, no es posible implementar un diseño modular en esta plataforma. Por eso, tuvimos que modificar la solución para que funcione de forma lineal y secuencial.

Ahora, en lugar de dividir las funciones en módulos separados, todo el código estará en un solo bloque. A pesar de esto, vamos a seguir usando buenas prácticas de programación para que el código sea claro y fácil de seguir, incluyendo comentarios y una buena organización de las variables.