**Análisis**

**Planteamiento del problema:**

La empresa informa2 solicita la implementación de un sistema que permita identificación y clasificación de señales analógicas generadas a través de generador de función que permite generar tres tipos de señales principales, senoidal triangular y cuadrada. Además, debe incluir posibilidad de calcular la frecuencia en Hertz, la amplitud de la señal de entrada. Lo anterior mediante la herramienta Tinkercad en el cual se crea una simulación de un circuito electrónico el cual está compuesto por los siguientes elementos:

1. Arduino UNO: es una placa de prototipado de circuitos electrónicos, la cual hace el procesamiento de las entradas de lectura y salida de los componentes.

2. Generador de funciones: es el dispositivo que genera la entrada de la señal analógica hacia el Arduino que posteriormente será analizada de acuerdo a los requerimientos.

3. Pulsadores: componente electrónico que actúa como interruptor cuya función es permitir o interrumpir el flujo de corriente en un instante de tiempo. Se hace uso de dos pulsadores los cuales le van a permitir al usuario interactuar con el sistema de la siguiente forma:

el pulsador 1 tiene la función de indicar al sistema el inicio de la captura de la señal

el pulsador 2 tiene la función de indicar al sistema el fin de la captura de la señal

nota: los pulsadores incluyen una resistencia de 10 kΩ que tienen la función de controlar el flujo de la corriente.

4. Placa de prueba: elemento que permite y facilita la conexión entre los diferentes componentes del circuito.

5. Pantalla LCD 16x2: elemento a través del cual se mostrará al usuario los datos de salida requeridos.

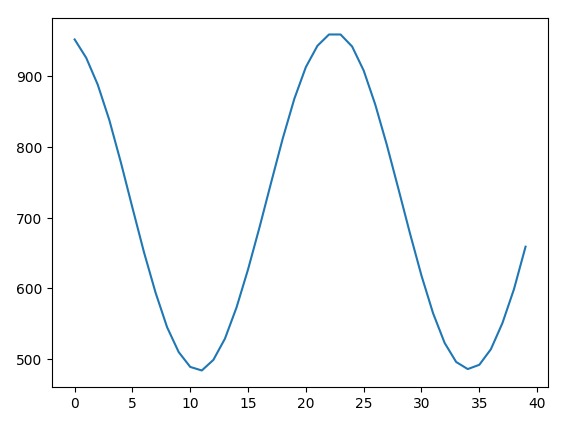
Por otro lado, en el componente del software se requiere el diseño e implementación de un algoritmo que permita identificar los tipos de señales, la frecuencia y la amplitud de señal y que permita la visualización de los resultados (tipo de señal, frecuencia en Hertz y amplitud en voltios).

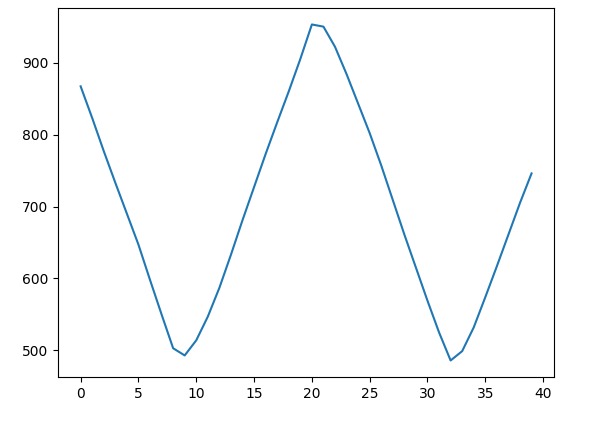
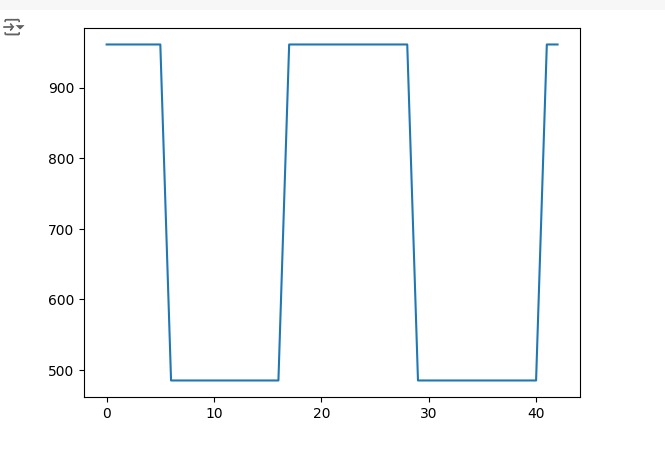
**Planteamiento de la solución:**

Para clasificar el tipo de señal analógica en los tres tipos mencionados en los requerimientos del cliente: senoidal, triangular, cuadrada y desconocida, se plantea la posible solución a partir de usar un enfoque basado en el análisis de los patrones de la señal.

A grandes rasgos, para distinguir estos tipos de señales:

* **Señal senoidal**: En una señal senoidal, los ascensos y descensos no son perfectamente lineales, y las diferencias consecutivas de los valores de la señal (derivadas aproximadas) suelen ser menores y más suaves. Podemos calcular la variación media y usar un umbral bajo para identificar la senoidal.



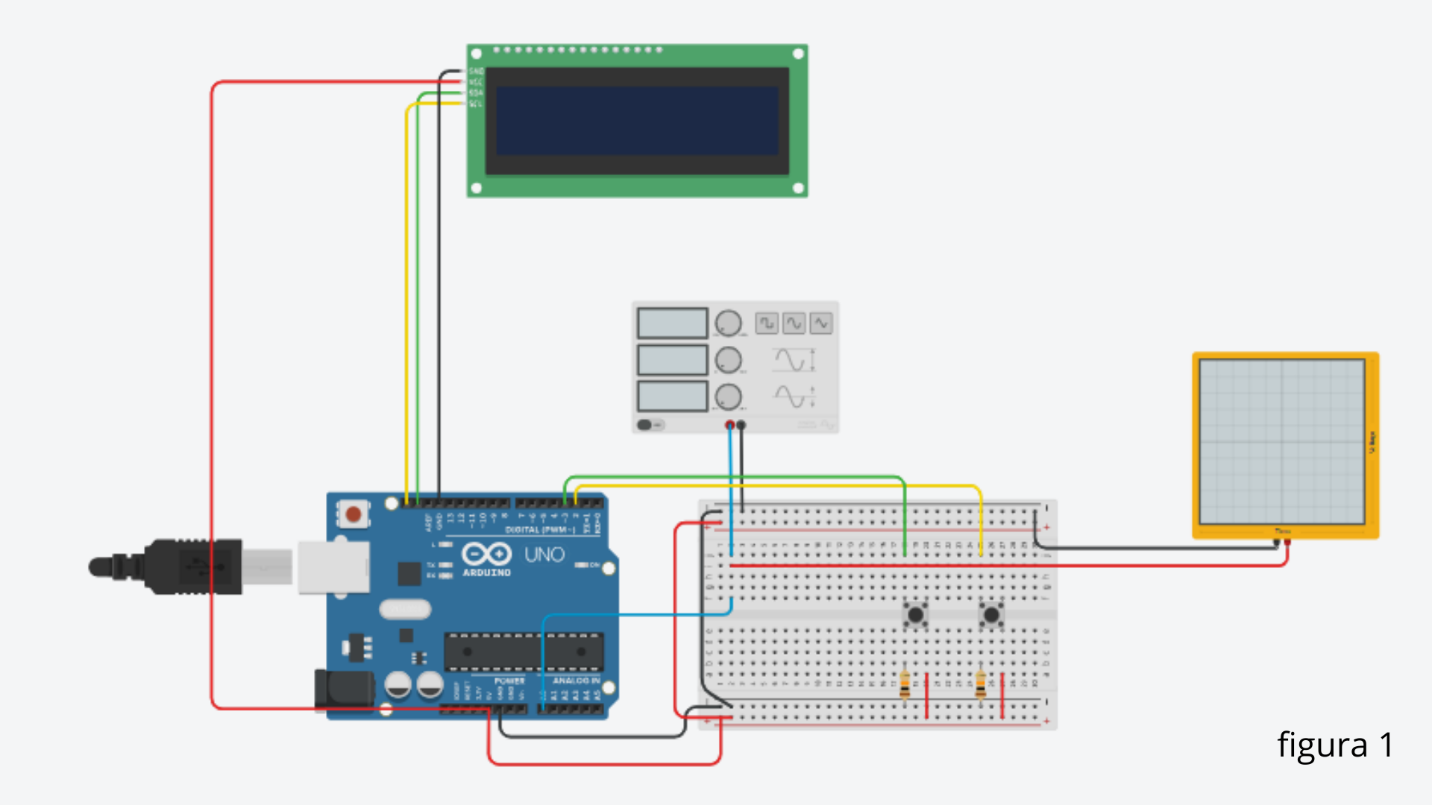
* **Señal triangular**: En la señal triangular, los ascensos y descensos son lineales, por lo que las diferencias consecutivas entre puntos son casi constantes. Si detectamos cambios abruptos que no correspondan a un salto cuadrado, la señal será triangular.
* **Señal cuadrada**: Tiene transiciones abruptas entre valores altos y bajos (una forma de onda con saltos rectangulares).

**Algoritmo en C++:**

1. **Representación de la señal**: La señal analógica se almacenará en un arreglo dinámico para usar la memoria dinámica (requerimiento).
2. **Clasificación de la señal**: El algoritmo va a analizar la pendiente y la forma de la señal para clasificarla.

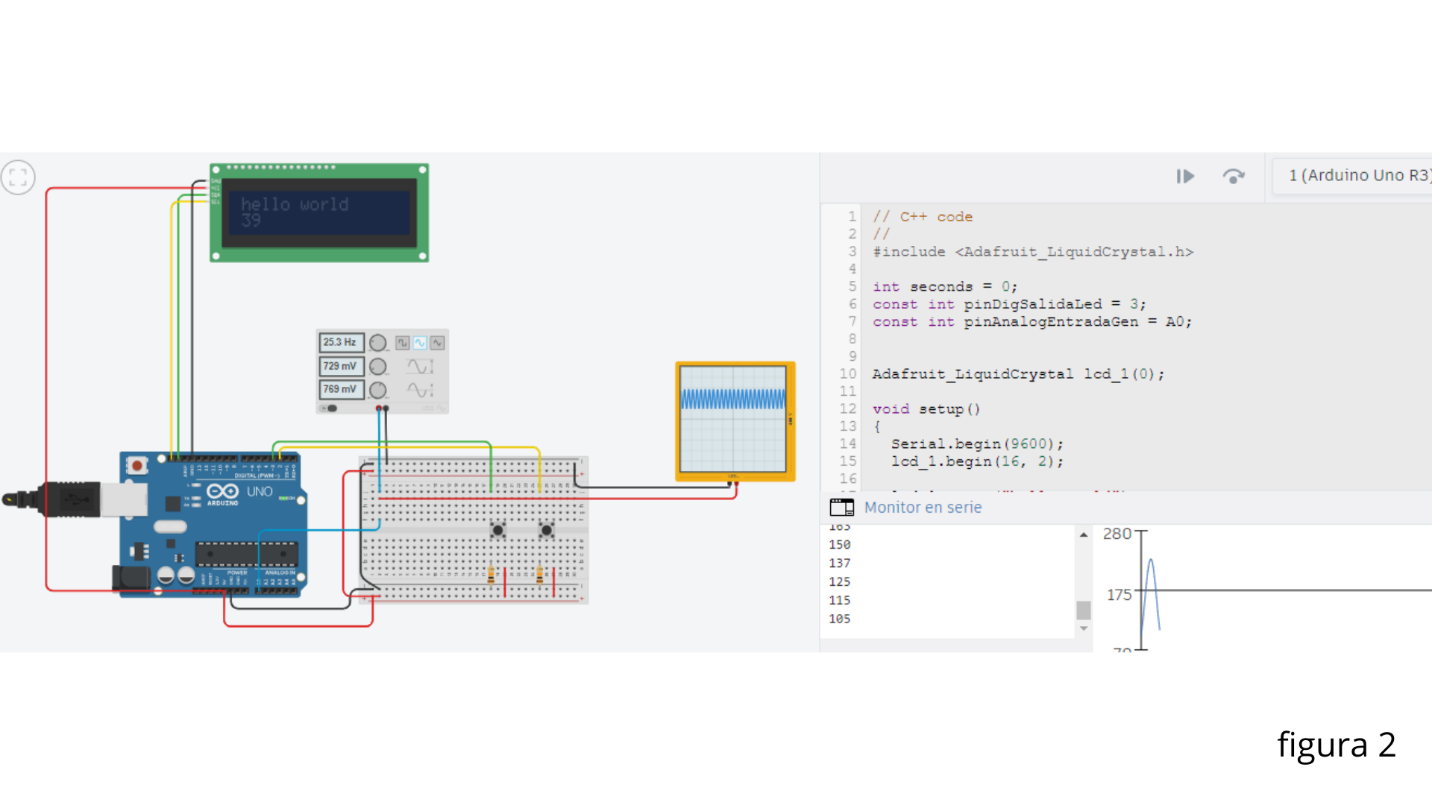
**Circuito en Tinkercad:**

**1. Circuito compuesto (figura 1):** el sistema propuesto está basado en componentes requeridos y opcionales:

* **Arduino LCD de dos cables:** placa de microcontrolador.
* **Generador de función:** genera diversas formas de onda de voltaje.
* **Osciloscopio(opcional):** para visualizar el comportamiento grafico de las señales.
* **Pulsador:** conmutador momentáneo.
* **Resistencia:** limita el flujo de corriente a través de un circuito.

**2. Interacción con el simulador (figura 2):**

* **Panel de texto:** instruye las acciones de los componentes.
* **Monitor en serie:** verifica o sigue que hace el código.
* **Acciones de los componentes:** visualizar el entorno real de la operación de los componentes.

****

**3**. **código en panel de texto (figura 3):** el programa Arduino tiene limitado el uso de memoria y aunque este se compone con algunas variaciones de sintaxis tiene el esquema de programación basado en c++. Con la diferencia de que este código de texto en Arduino está compuesto de dos funciones principales de las cuales depende para funcionar y cumplir las órdenes, estas funciones son:

**void setup():** funciona para inicializar los pines de entrada o salida, configurar velocidad, frecuencia y compatibilidad con el monitor serial.

**Propuesta de solución**