**a Análisis del problema y consideraciones para la alternativa de solución propuesta**.

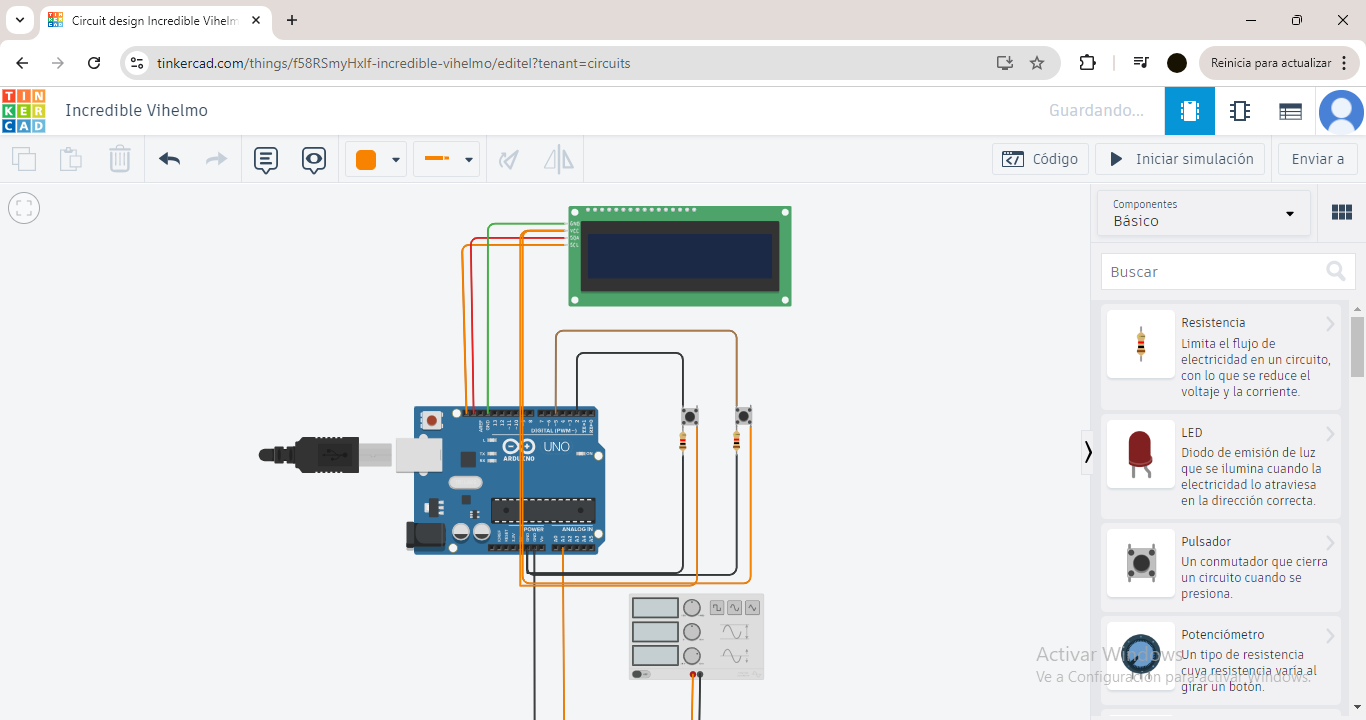
El problema central del desafío es implementar un sistema que pueda capturar una señal analógica, identificar las características frecuencia, amplitud, forma de onda y visualizarlas en una pantalla LCD en arduino. El sistema debe ser controlado a través de un pulsador para iniciar y pausar la adquisición de datos. Se implementarán funciones para capturar la señal, analizar los datos, calcular la frecuencia y amplitud e identificar la forma de onda. El uso de arreglos dinámicos y punteros es crucial para almacenar los datos capturados de manera eficiente.

Características de las ondas

Senoidal: la forma suave y continua o sea que la onda sube y baja de manera gradual y repetitiva, sin cambios bruscos, también que es simetría ya que la mitad positiva de la onda es simétrica a la mitad negativa. Ciclos periódicos por que los valores máximos y mínimos se repiten a intervalos regulares y por ultimo cambio gradual de pendiente ya que la curva varía suavemente, pasando de valores positivos a negativos y viceversa.

Triangular: Subida y bajada lineal: Los valores de la señal aumentan y disminuyen de manera constante (es decir, la pendiente es constante).Simetría: La onda tiene una subida y una bajada simétricas. La pendiente positiva es igual en magnitud a la pendiente negativa.Cambio gradual: A diferencia de la onda cuadrada, la transición entre los picos no es abrupta, sino gradual, y el valor cambia de manera lineal entre el máximo y el mínimo.

**b. Esquema donde describa las tareas que usted definió en el desarrollo de los algoritmos**



**c. Algoritmos implementados.**

Para capturar la señal se usa la función analogRead() para leer la señal analógica y esta se almacena en un arreglo dinamico

int\* data = nullptr; // Puntero a los datos

int dataSize = 0; // Tamaño del arreglo

void agregarDato(int valor) {

int\* nuevoArray = new int[dataSize + 1];

for (int i = 0; i < dataSize; i++) {

nuevoArray[i] = data[i];

}

nuevoArray[dataSize] = valor;

delete[] data;

data = nuevoArray; dataSize++;

}

Calculamos la amplitud calculando el valor max y min

float calcularAmplitud(int\* data, int dataSize) {

int maxVal = data[0];

int minVal = data[0];

for (int i = 1; i < dataSize; i++) {

if (data[i] > maxVal) maxVal = data[i];

if (data[i] < minVal) minVal = data[i];

}

return (maxVal - minVal) \* (5.0 / 1023); // Conversión de valores a voltios

}

**d. Problemas de desarrollo que afrontó.**

**e. Evolución de la solución y consideraciones para tener en cuenta en la implementación.**

**Consideraciones**

Optimizar el uso de memoria para evitar cuellos de botella en el procesamiento de señales largas Asegurarse de que la visualización en la pantalla LCD sea clara y fácil de entender, mostrando solo los datos relevantes.

.