# Informe desarrollo de proyecto

### Desafio 1 2024

##### Emanuel Guerra Urrea

##### Sebastián Molina Ciro

El problema que se presenta, es:  
  
Desarrollar un montaje en thinkercad, el cual debe de permitir realizar la simulación del

sistema de adquisición y visualización de la información de la señal capturada.

Implementar un algoritmo que permita medir la frecuencia en Hertz y la

amplitud, en Voltios, de la señal de entrada

Implementar un algoritmo que permita identificar la forma de onda de la señal

de entrada, siendo estas las ondas incluidas en el generador de señales de thinkercad.

La solución que proponemos es la siguiente:

**se necesitan 7 funciones** , las cuales son:

Iniciar adquisición

Detener adquisición

Analizar la señal

Calcular la frecuencia

Calcular la amplitud

Identificar la forma de onda

Mostrar los resultados

##### Iniciar adquisición

void iniciar\_adquisicion() {

adquisicion\_activa = true;

indice\_muestra = 0;

lcd.clear();

lcd.print("Adquiriendo...");

}

##### Detener adquisición

void detener\_adquisicion() {

adquisicion\_activa = false;

}

##### **Analizar la señal**

void analizar\_senal() {

float frecuencia = calcular\_frecuencia();

float amplitud = calcular\_amplitud();

String forma = identificar\_forma\_onda();

mostrar\_resultados(frecuencia, amplitud, forma);

}

##### Calcular la frecuencia

float calcular\_frecuencia() {

// Implementar algoritmo de detección de cruces por cero

int cruces\_cero = 0;

float umbral = 2.5; // Asumiendo una señal centrada en 2.5V

for (int i = 1; i < TAMANO\_BUFFER; i++) {

if ((muestras[i-1] < umbral && muestras[i] >= umbral) ||

(muestras[i-1] > umbral && muestras[i] <= umbral)) {

cruces\_cero++;

}

}

float tiempo\_total = TAMANO\_BUFFER / 1000.0; // Asumiendo muestreo a 1kHz

return cruces\_cero / (2 \* tiempo\_total);

}

##### Calcular la amplitud

float calcular\_amplitud() {

float max\_valor = muestras[0];

float min\_valor = muestras[0];

for (int i = 1; i < TAMANO\_BUFFER; i++) {

if (muestras[i] > max\_valor) max\_valor = muestras[i];

if (muestras[i] < min\_valor) min\_valor = muestras[i];

}

return (max\_valor - min\_valor) / 2.0;

}

##### Identificar la forma de onda

##### Mostrar los resultados

void mostrar\_resultados(float frecuencia, float amplitud, String forma) {

lcd.clear();

lcd.print("F:");

lcd.print(frecuencia, 2);

lcd.print("Hz A:");

lcd.print(amplitud, 2);

lcd.print("V");

lcd.setCursor(0, 1);

lcd.print("Forma: ");

lcd.print(forma);

}

String identificar\_forma\_onda() {

// Implementación simplificada - se necesita un algoritmo más robusto

float suma\_derivadas = 0;

for (int i = 1; i < TAMANO\_BUFFER; i++) {

suma\_derivadas += abs(muestras[i] - muestras[i-1]);

}

float promedio\_derivadas = suma\_derivadas / TAMANO\_BUFFER;

if (promedio\_derivadas < 0.1) return "Cuadrada";

else if (promedio\_derivadas < 0.5) return "Triangular";

else return "Senoidal";

}

# Problemas en el desarrollo.

Uno de los problemas en el desarrollo, fue la implementación de la función para identificar la forma de onda, ya que esta sin importar que, siempre devolvía, que la onda era senoidal.

Aún falta mas trabajo para poder corregir este error.