



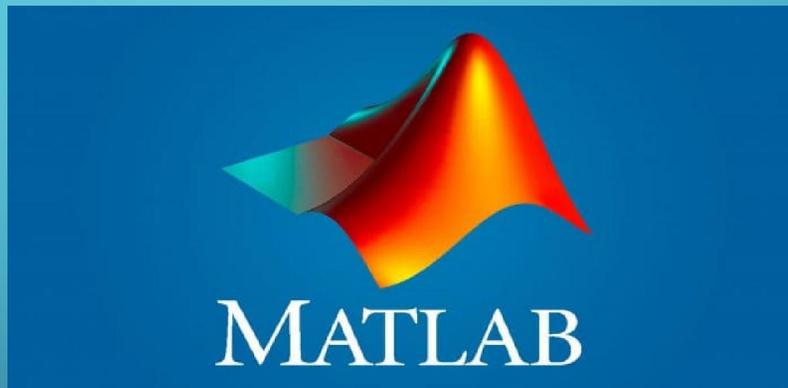
**INSTITUTO TECNOLÓGICO SUPERIOR DE  
SAN ANDRÉS TUXTLA**



**ITSSAT  
INGENIERIA MECATRONICA  
SEPTIMO SEMESTRE  
GRUPO:711 A  
INSTRUMENTACIÓN VIRTUAL  
APUNTES**

**DOCENTE: DR. JOSE ANGEL NIEVES VAZQUEZ  
ALUMNO: ARANTZA GUADALUPE GARCIA ZAPOT**

# ADQUISICIÓN DE DATOS USANDO:



POR: ARANTZA GUADALUPE GARCIA ZAPOT

INSTITUTO TECNOLÓGICO SUPERIOR DE SAN ANDRÉS TUXTLA



INSTITUTO TECNOLÓGICO SUPERIOR DE  
SAN ANDRÉS TUXTLA

INGENIERÍA MECATRÓNICA

SÉPTIMO SEMESTRE GRUPO 711 A

INSTRUMENTACIÓN VIRTUAL

DR. JOSÉ ÁNGEL NIEVES VÁZQUEZ



## INTRODUCCIÓN

---

## ADQUISICIÓN DE DATOS

TIPOS DE SEÑALES

CONSIDERACIONES IMPORTANTES

DESAFIOS

## MATLAB

FUNCIONES Y HERRAMIENTAS

VENTAJAS

CODIGO PARA ADQUIRIR DATOS

VISUALIZACION Y ANALISIS DE DATOS

APLICACIONES

## CONCLUSIÓN

---

# INTRODUCCION

EN ESTA PRESENTACIÓN ABORDAREMOS EL PROCESO DE ADQUISICIÓN DE DATOS, EMPLEANDO UNO DE LOS SOFTWARES MÁS CONOCIDOS:

MATLAB, como muchos de ustedes sabrán, es una potente herramienta de software utilizada en ingeniería, ciencia y otras áreas técnicas. Entre sus numerosas capacidades, una de las más destacadas es su capacidad para la adquisición y análisis de datos. En esta presentación, exploraremos cómo MATLAB facilita este proceso crucial, desde la captura de datos hasta su visualización y análisis.





# ADQUISICIÓN DE DATOS

# ADQUISICIÓN DE DATOS

- Como ya sabemos, el proceso de adquisición de datos es el proceso de recolección y almacenamiento de información proveniente de sensores, dispositivos o sistemas.

# ADQUISICIÓN DE DATOS



# TIPOS DE SEÑALES

```
graph TD; A[TIPOS DE SEÑALES] --> B[ANALOGICA]; A --> C[DIGITALES]; B --- D[\"Son señales que varían de forma continua en el tiempo y pueden tomar cualquier valor dentro de un rango determinado.\"]; C --- E[\"Estas señales son discretas en el tiempo y en el nivel, tomando valores definidos en puntos específicos. Son típicas en sistemas digitales y se representan utilizando niveles lógicos, como 0 y 1. Ejemplos incluyen señales binarias, pulsos digitales y datos digitales transmitidos por buses.\"];
```

## ANALOGICA

Son señales que varían de forma continua en el tiempo y pueden tomar cualquier valor dentro de un rango determinado.

## DIGITALES

Estas señales son discretas en el tiempo y en el nivel, tomando valores definidos en puntos específicos.

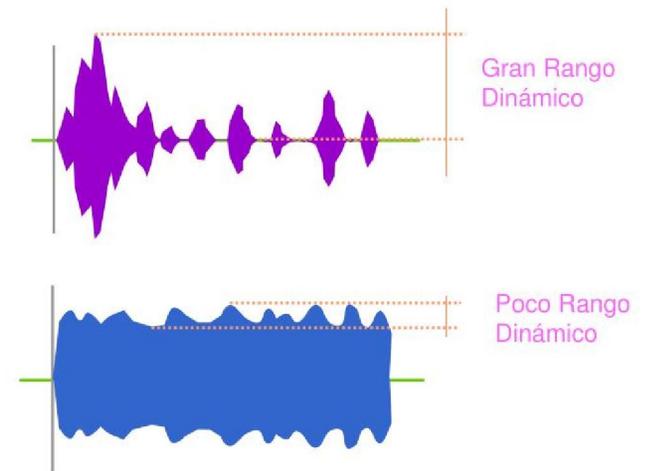
Son típicas en sistemas digitales y se representan utilizando niveles lógicos, como 0 y 1. Ejemplos incluyen señales binarias, pulsos digitales y datos digitales transmitidos por buses.

# CONSIDERACIONES IMPORTANTES EN LA ADQUISICIÓN DE DATOS

## RANGO DINAMICO

- El rango dinámico se refiere a la diferencia entre el valor más bajo y el valor más alto que puede medir un sistema de adquisición de datos antes de que se produzca la saturación.
- Un rango dinámico amplio es importante para garantizar que el sistema pueda capturar y representar con precisión una amplia gama de valores de señal.
- Se expresa típicamente en decibelios (dB) y se calcula como la relación entre la señal máxima y el ruido de fondo.

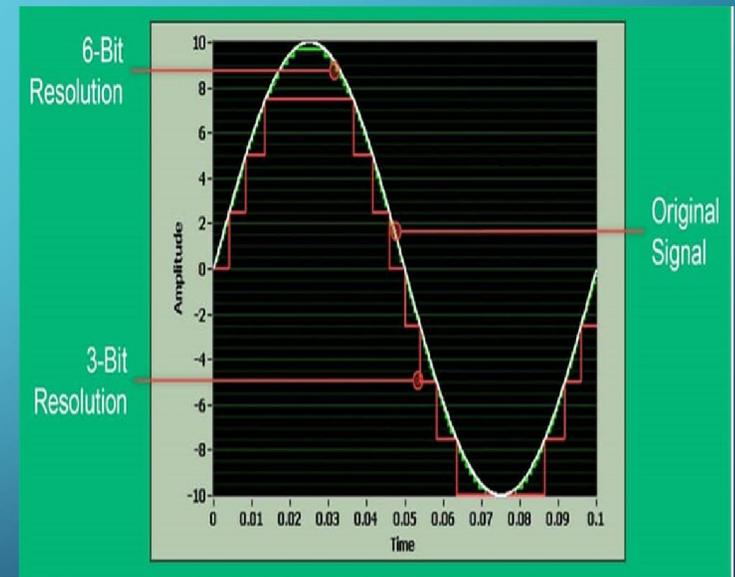
## RANGO DINAMICO



# CONSIDERACIONES IMPORTANTES EN LA ADQUISICIÓN DE DATOS

## RESOLUCIÓN

- La resolución se refiere a la capacidad del sistema para distinguir entre valores muy pequeños o cercanos entre sí.
- En el contexto de la adquisición de datos, la resolución se refiere al número de bits utilizados para representar la señal analógica en su versión digitalizada.
- Por ejemplo, una resolución de 16 bits permite representar la señal en  $2^{16}$  niveles discretos, lo que proporciona una mayor precisión que una resolución de 8 bits.



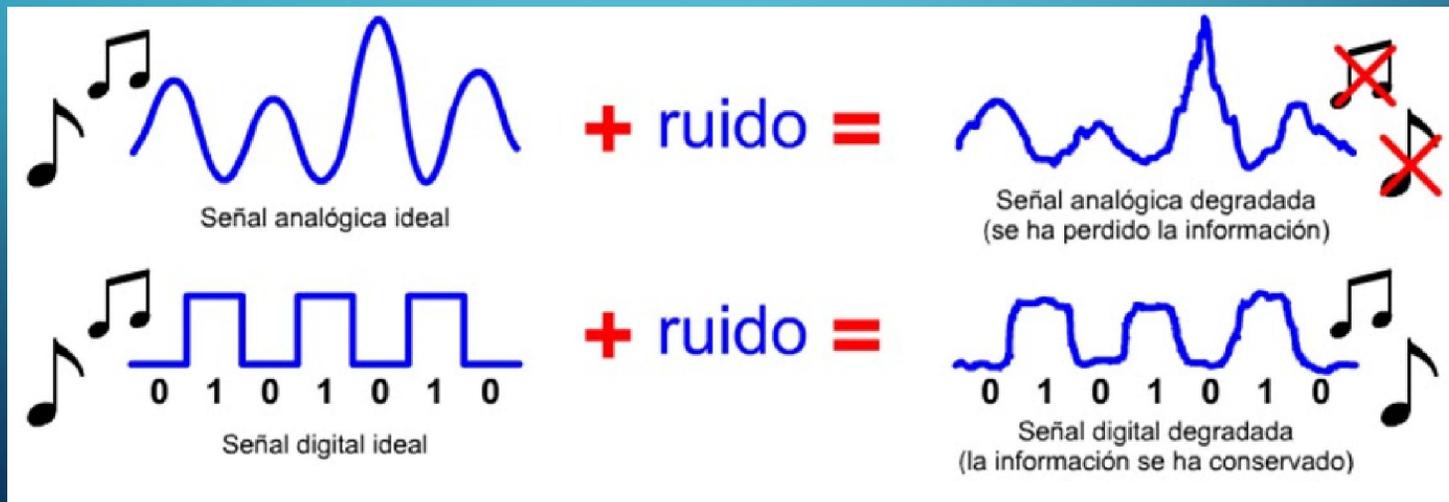
# CONSIDERACIONES IMPORTANTES EN LA ADQUISICIÓN DE DATOS

## TASA DE MUESTREO

- La tasa de muestreo, también conocida como frecuencia de muestreo, es la frecuencia a la que se toman muestras de una señal analógica para convertirla en una señal digital.
- Se mide en muestras por segundo (SPS o Hz) y determina la cantidad de información capturada por unidad de tiempo.
- La tasa de muestreo debe ser lo suficientemente alta como para evitar el aliasing, que ocurre cuando las frecuencias de la señal son mayores que la mitad de la tasa de muestreo (frecuencia de Nyquist)

# DESAFIOS

- Variedad de formatos y fuentes de datos.
- Ruido, errores y perturbaciones.
- Necesidad de sincronización y muestreo preciso.
- Manejo de grandes volúmenes de datos.





MATLAB<sup>®</sup>

# MATLAB

MATLAB es un entorno de software de alto nivel y un lenguaje de programación especializado ampliamente utilizado en ingeniería, ciencia y otras disciplinas técnicas. Proporciona un conjunto poderoso de herramientas para el análisis numérico, la visualización de datos, el modelado matemático y la implementación de algoritmos.



**MATLAB<sup>®</sup>**

# MATLAB

## UN POCO DE HISTORIA

1970  
Desarrollo inicial por Cleve Moler

1984  
Fundación de MathWorks por Jack Little, Cleve Moler y Steve Bangert para comercializar MATLAB.

1992  
Introducción de GUI (Interfaz Gráfica de Usuario) en la versión 4.0 facilitando su uso para una audiencia más amplia.

2006  
Transición a la arquitectura de 64 bits para manejar conjuntos de datos más grandes y mejorar el rendimiento.

2019  
Lanzamiento de MATLAB Online permitiendo el acceso a MATLAB a través de un navegador web.



MATLAB®

# MATLAB

## FUNCIONES Y HERRAMIENTAS



### Análisis de los datos

Explorar, modelar y analizar datos



### Gráficos

Visualice y explore datos



### Programación

Crear scripts, funciones y clases.



### Creación de aplicaciones

Crear aplicaciones web y de escritorio



### Interfaces de idiomas externos

Utilice MATLAB con Python, C/C++, Fortran, Java y otros lenguajes



### Hardware

Conecte MATLAB al hardware



### Computación paralela

Realice cálculos a gran escala y paralelice simulaciones utilizando escritorios multinúcleo, GPU, clústeres y nubes.



### Implementación web y de escritorio

Comparte tus programas MATLAB



### MATLAB en la nube

Ejecute en entornos de nube desde MathWorks Cloud hasta nubes públicas, incluidas AWS y Azure.



# MATLAB®

# MATLAB

## VENTAJAS DE USAR MATLAB

- **Amplia gama de herramientas y funciones**
- **Facilidad de uso**
- **Eficiencia computacional**
- **Visualización de datos**
- **Flexibilidad y personalización**
- **Interoperabilidad con otros lenguajes y herramientas**



# MATLAB

## CODIGO PARA ADQUIRIR DATOS

```
% Crear una sesión de adquisición de datos
s = daq.createSession('ni');

% Agregar canales de entrada (por ejemplo, un canal analógico)
addAnalogInputChannel(s,'Dev1',0,'Voltage');

% Configurar la tasa de muestreo y la duración de la adquisición
s.Rate = 1000; % Tasa de muestreo en muestras por segundo
duration = 5; % Duración de la adquisición en segundos

% Adquirir datos
[data, timestamps] = s.startForeground();

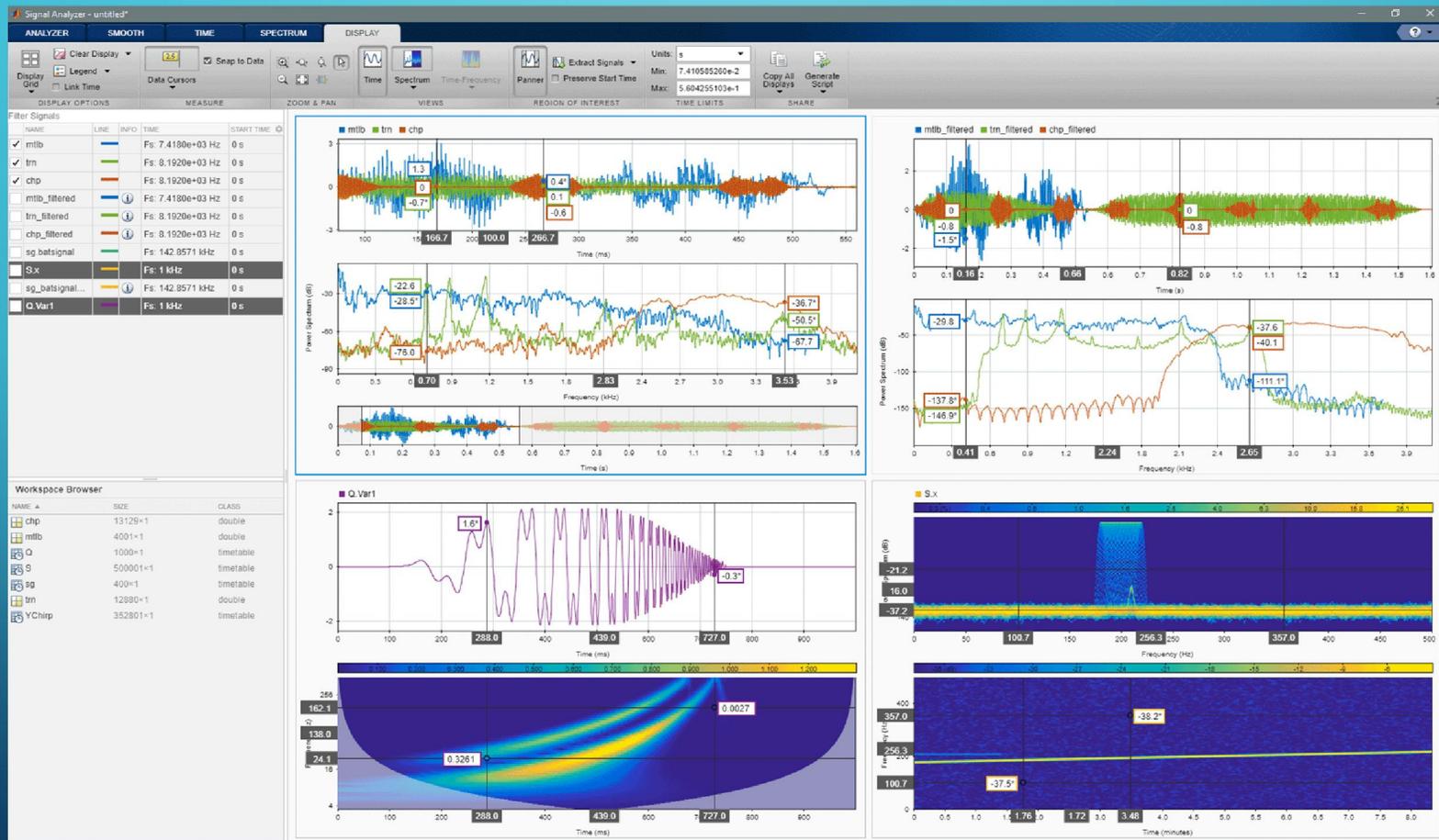
% Graficar los datos adquiridos
plot(timestamps, data);
xlabel('Tiempo (s)');
ylabel('Voltaje (V)');
title('Datos Adquiridos');

% Limpieza de la sesión
delete(s);
```



**MATLAB®**

# MATLAB



VISUALIZACION Y ANALISIS DE DATOS

# MATLAB

## APLICACIONES



### Inteligencia artificial (IA)

Transforme la ingeniería y las ciencias con IA



### Sistemas de conducción autónoma

Diseñe, simule y pruebe sistemas de conducción autónoma



### Biología computacional

Analice, visualice y modele sistemas y datos biológicos



### Sistemas de control

Diseñe, pruebe e implemente sistemas de control



### Data Science

Explore datos, cree modelos de Machine Learning y realice análisis predictivo



### Deep Learning

Prepare datos, diseñe, simule y despliegue en redes neuronales profundas



### Electrificación

Desarrolle tecnología eléctrica desde componentes hasta sistemas



### Sistemas integrados

Diseñe, codifique y verifique sistemas integrados



### Sistemas informáticos y empresariales

Utilice MATLAB con sus sistemas informáticos



### Desarrollo de FPGA, ASIC y SoC

Automatice el flujo de trabajo desde el desarrollo de algoritmos hasta el diseño y la verificación de hardware



### Procesamiento de imágenes y visión artificial

Capture, procese y analice imágenes y videos para el desarrollo de algoritmos y el diseño de sistemas



### Internet of Things

Conecte dispositivos integrados a Internet y obtenga información a partir de los datos



### Machine Learning

Entrene modelos, ajuste parámetros y despliegue en producción o en dispositivos edge



### Mecatrónica

Diseñe, optimice y verifique sistemas mecatrónicos



### Sistemas de señal mixta

Analice, diseñe y verifique sistemas de señal analógica y mixta



### Mantenimiento predictivo

Desarrolle y despliegue software de supervisión de condiciones y mantenimiento predictivo



### Sistemas de radar

Diseñe, simule, pruebe y despliegue sistemas de radar multifunción



### Robótica

Diseñe, simule y verifique sistemas robóticos y autónomos



### Procesamiento de señales

Analice señales y datos de series temporales. Modele, diseñe y simule sistemas de procesamiento de señales



### Prueba y medición

Capture, analice y explore datos y automatice pruebas



### Comunicaciones inalámbricas

Cree, diseñe, pruebe y verifique sistemas de comunicaciones inalámbricas

# CONCLUSIÓN

- En conclusión, MATLAB es una herramienta extremadamente poderosa y versátil para la adquisición, procesamiento y análisis de datos en una amplia gama de aplicaciones en ingeniería, ciencia y otros campos técnicos. A lo largo de esta presentación, hemos explorado algunas de las ventajas clave de utilizar MATLAB en el contexto de la instrumentación virtual y la adquisición de datos.
- Hemos visto cómo MATLAB ofrece una amplia variedad de funciones y herramientas integradas para realizar tareas como el procesamiento numérico, la visualización de datos, el modelado y la simulación de sistemas, el análisis estadístico, el control de procesos y mucho más. Su sintaxis simple y legible, combinada con su interfaz gráfica de usuario intuitiva, facilita la escritura, ejecución y depuración de código, incluso para aquellos que no tienen una experiencia extensa en programación.

The image features a teal-to-blue gradient background. In the four corners, there are decorative white line-art elements resembling circuit traces or neural network connections, with small circles at the end of the lines.

MUCHAS GRACIAS POR SU ATENCIÓN J

# GUIA DE OBSERVACIÓN EXPOSICION INSTRUMENTACIÓN VIRTUAL.



Nombre del estudiante: Arantza Guadalupe García Zapot.

Tema: Adquisición de datos usando Matlab.

Explicación	10	10
Dominio del tema	10	10
Presentación en tiempo y forma	10	10
Total	30 %	30 %



INSTITUTO TECNOLÓGICO SUPERIOR DE  
SAN ANDRÉS TUXTLA



INSTITUTO TECNOLÓGICO SUPERIOR DE SAN ANDRÉS TUXTLA

INGENIERÍA MECATRÓNICA

SEPTIMO SEMESTRE

GRUPO: 711 A

INSTRUMENTACIÓN VIRTUAL

PRACTICA UNIDAD: 3 (Elementos de adquisición de datos)

DOCENTE: DR. JOSE ANGEL NIEVES VAZQUEZ

ALUMNO: ARANTZA GUADALUPE GARCIA ZAPOT

## INDICE

INDICE .....	1
INTRODUCCIÓN .....	2
DESARROLLO DE LA PRACTICA.....	3

## INTRODUCCIÓN

El presente informe detalla los procedimientos, resultados y conclusiones obtenidas en la práctica realizada con el software LabVIEW, donde se simuló el llenado de un tanque virtual. Durante esta actividad, se emplearon sensores de nivel para monitorear el nivel de llenado del tanque, los cuales a su vez activaron indicadores LED y generaron una representación gráfica en tiempo real.

La simulación del llenado de un tanque es un ejercicio relevante en el contexto de la instrumentación y el control de procesos, permitiendo entender cómo se comportan los sistemas de monitoreo y control en un entorno práctico. El uso de LabVIEW como herramienta principal para este propósito ofrece una experiencia significativa en el desarrollo de aplicaciones de adquisición de datos y visualización.

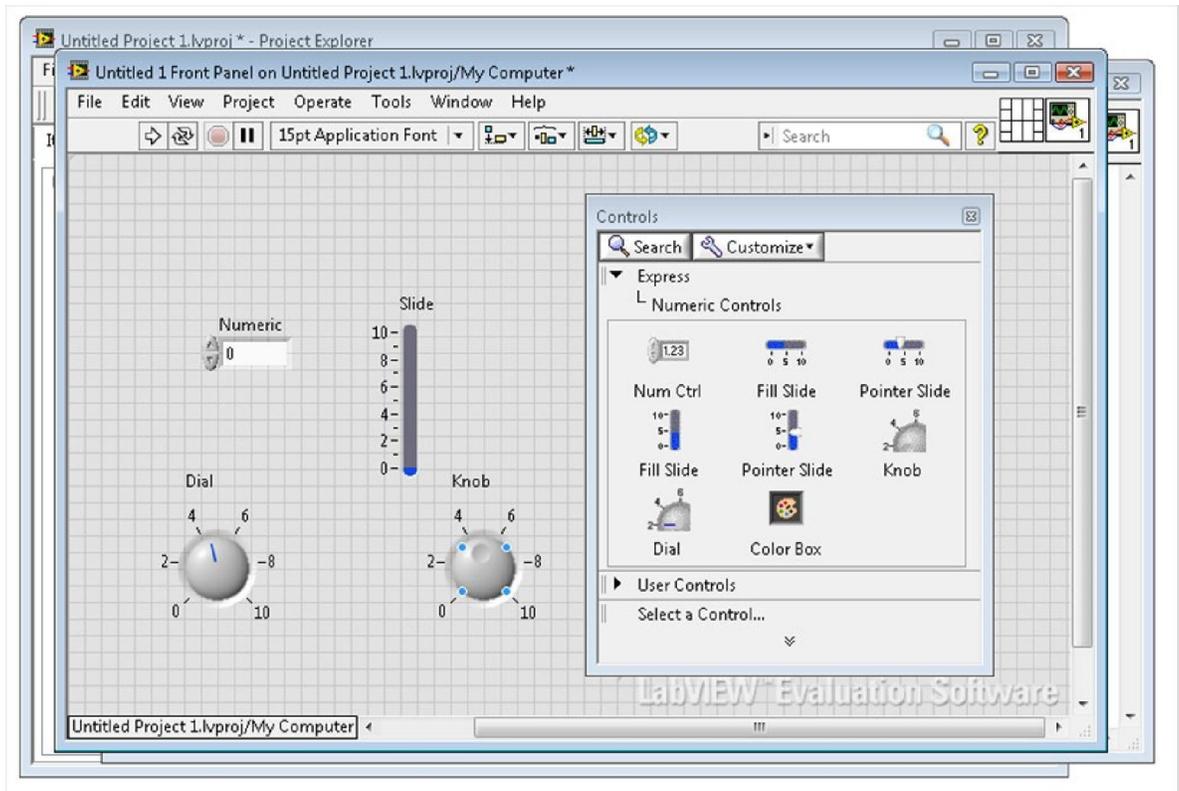
## DESARROLLO DE LA PRACTICA

Para la realización de esta practica fue necesario contar con:

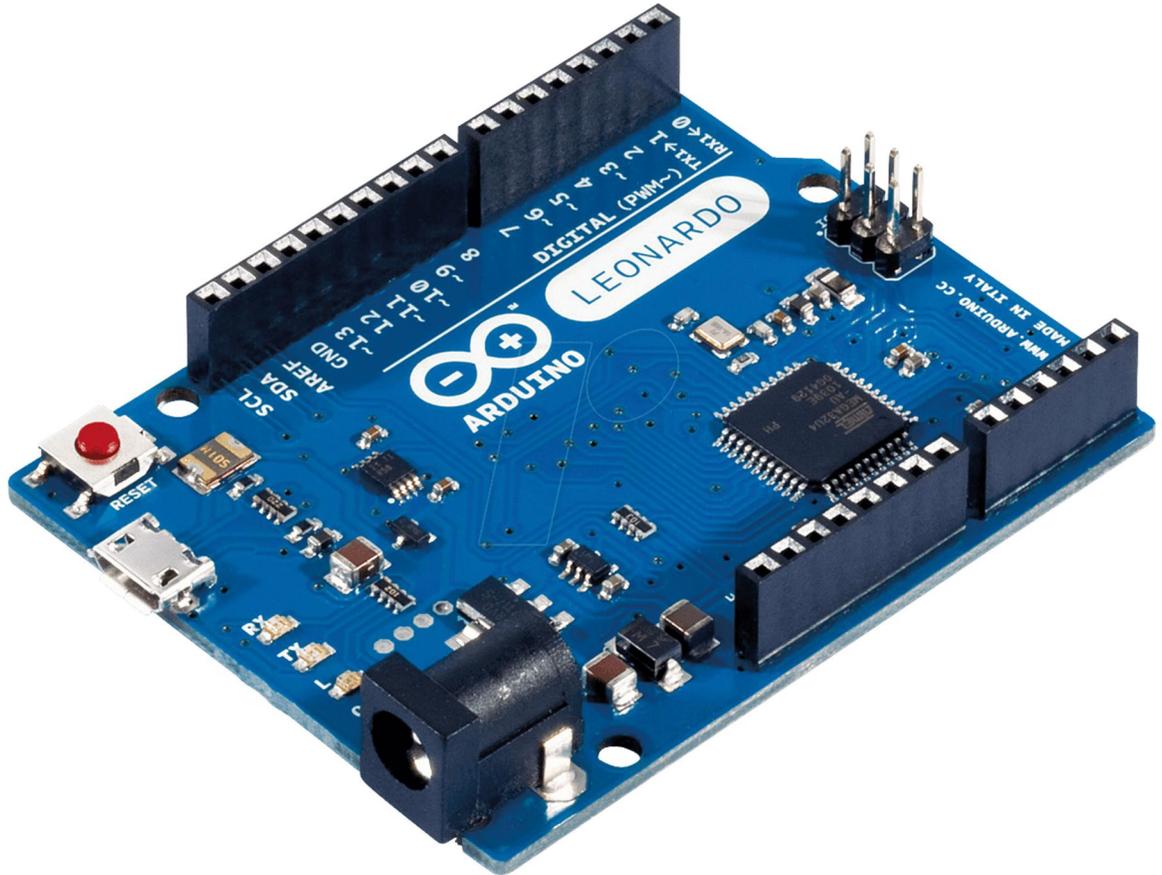
- ✓ Una pc



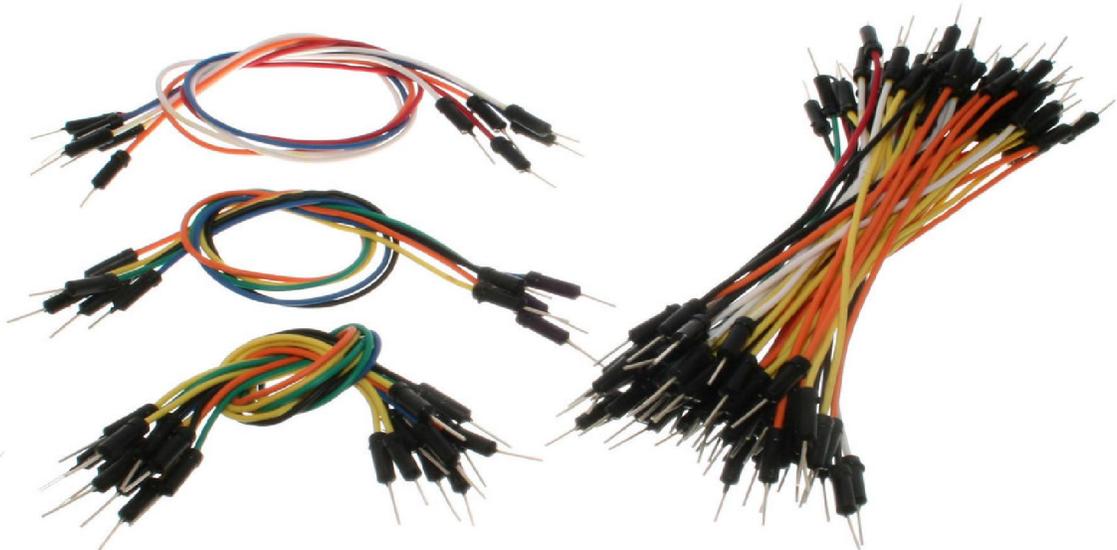
- ✓ Instalar el software LABVIEW



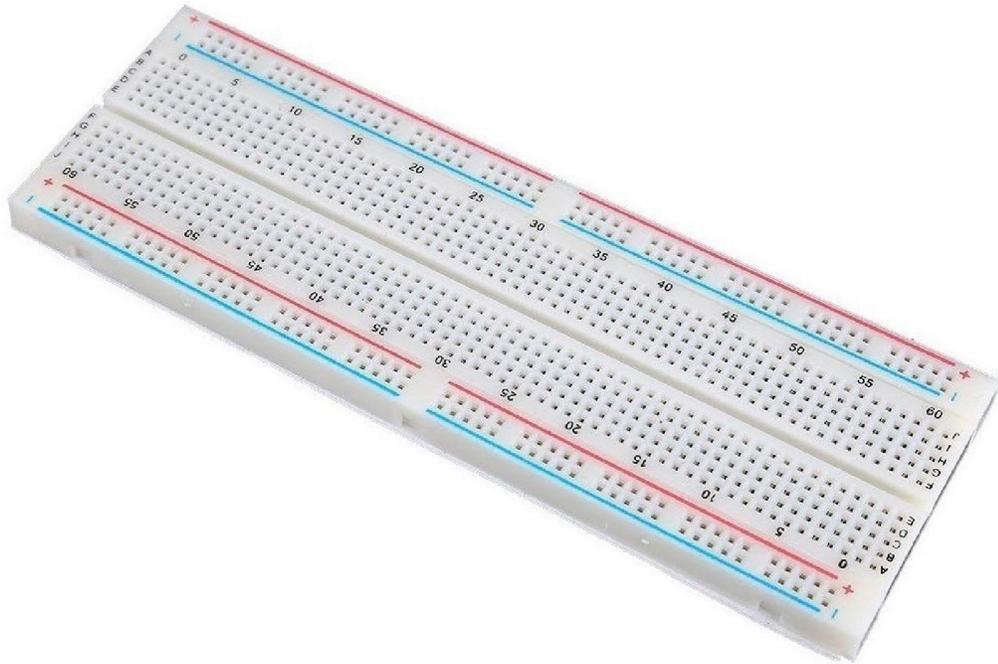
✓ Arduino uno



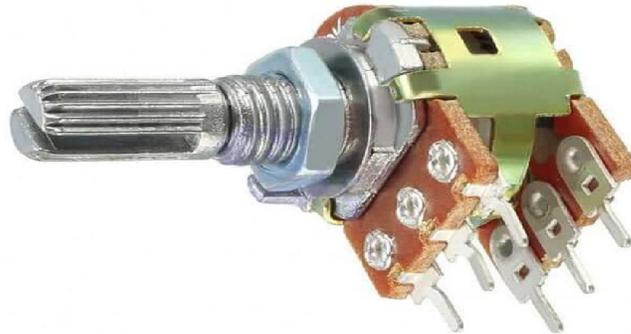
✓ Jumpers



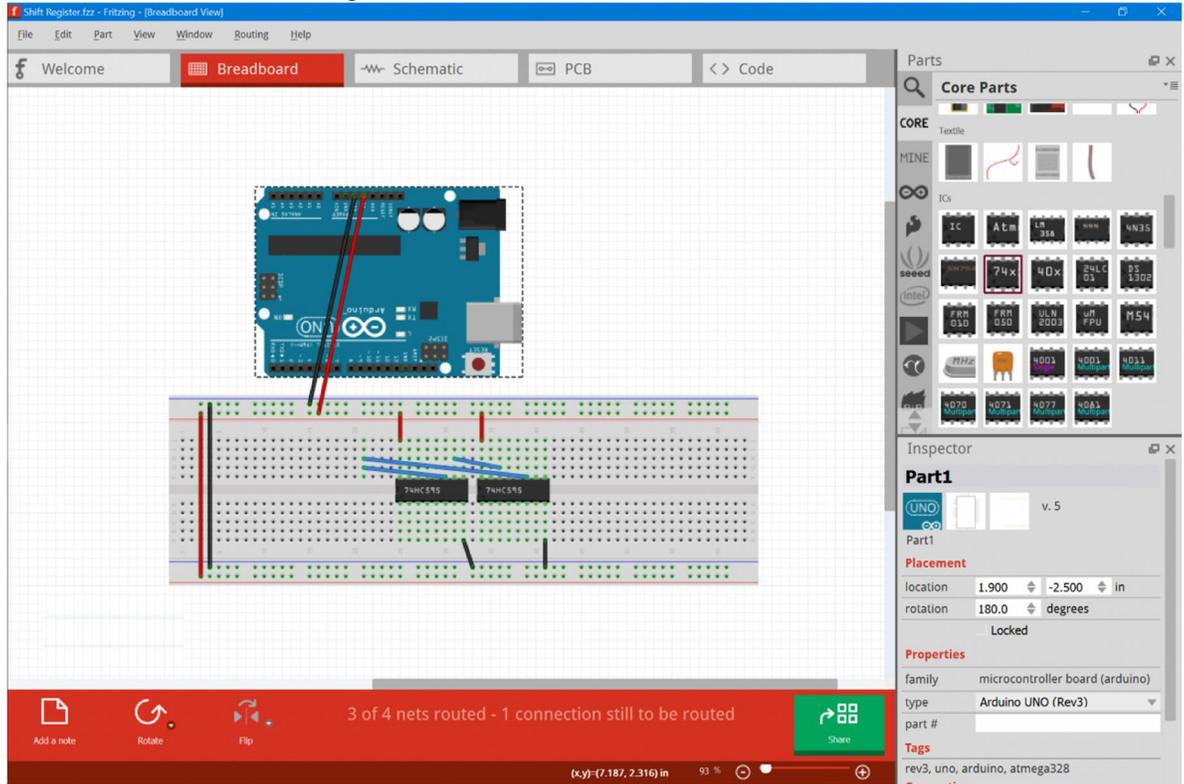
✓ Protoboard



✓ Potenciómetro

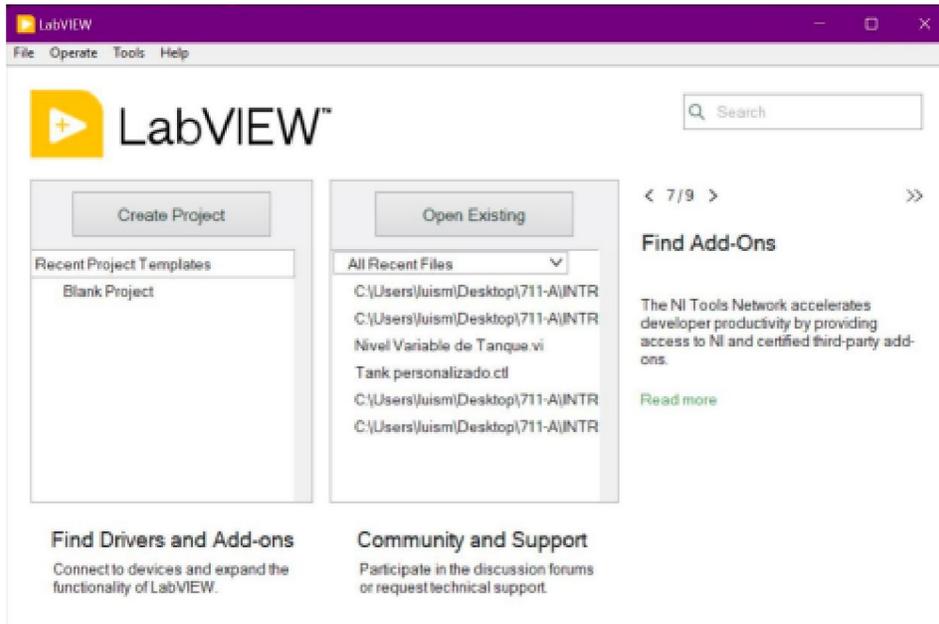


## ✓ Instalar el software Fritzing

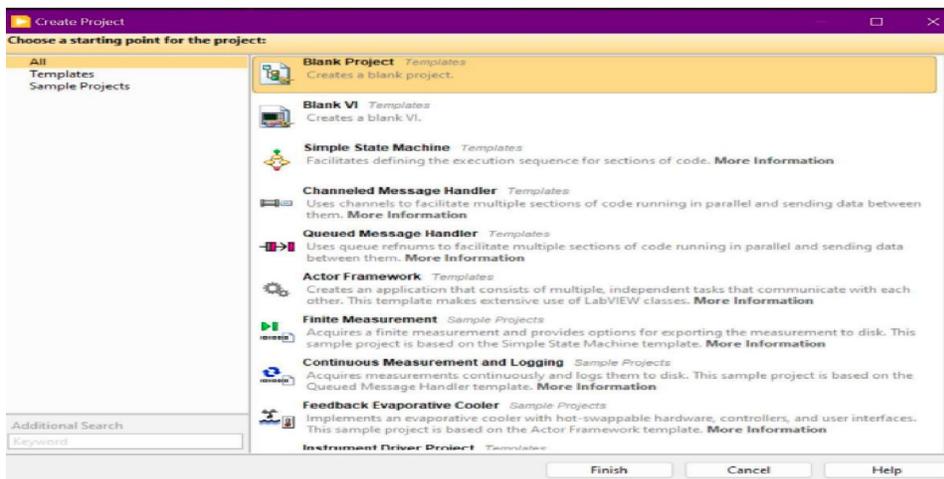


## CREACION DEL SISTEMA EN LABVIEW

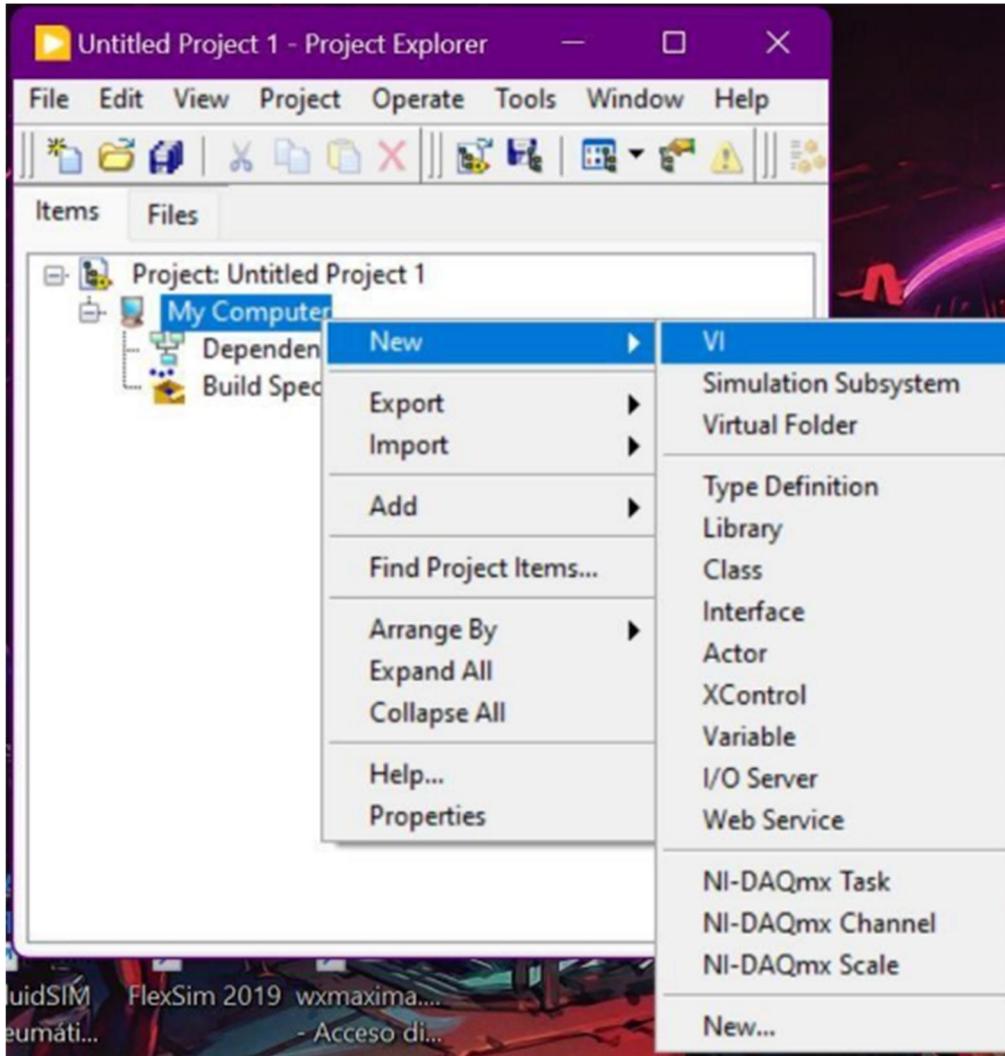
Para dar inicio con este proyecto, debemos abrir el programa anteriormente mencionado en nuestro equipo de cómputo donde seleccionaremos la opción de crear un nuevo proyecto (Create Project).



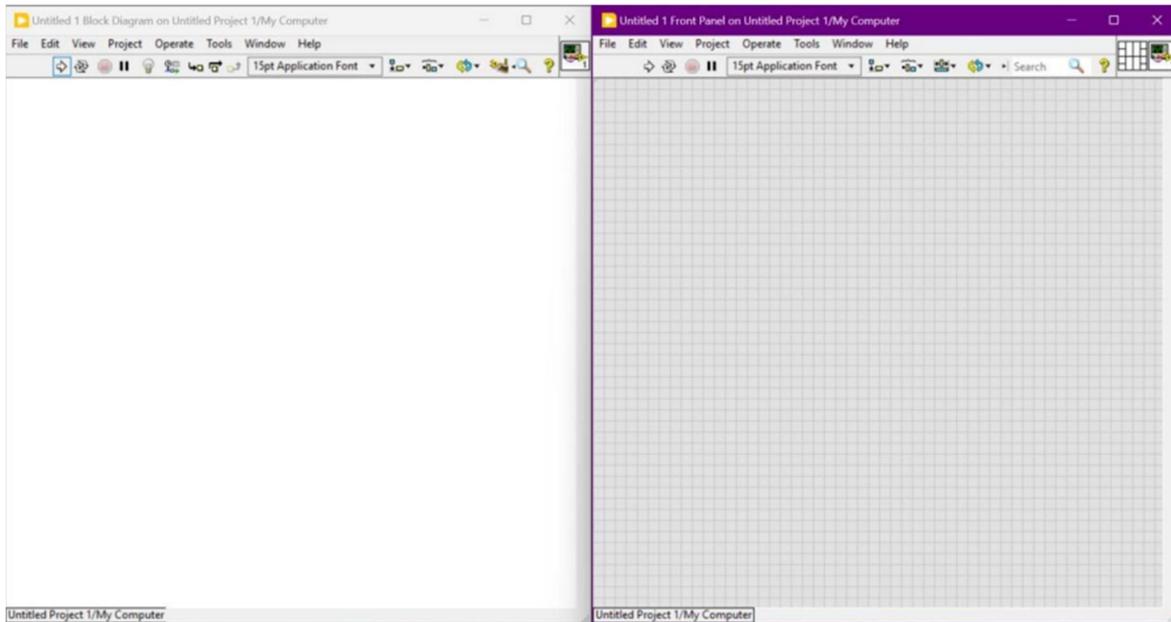
Sucesivamente de ello, pulsaremos dos veces en “Blank Project”



Ahora bien, para comenzar a programar y diseñar nuestro instrumento virtual, debemos dar click derecho sobre “My Computer”, luego seleccionar “New” y finalmente “VI”. Después de los pasos anteriores, estamos listos para comenzar a editar nuestro proyecto.



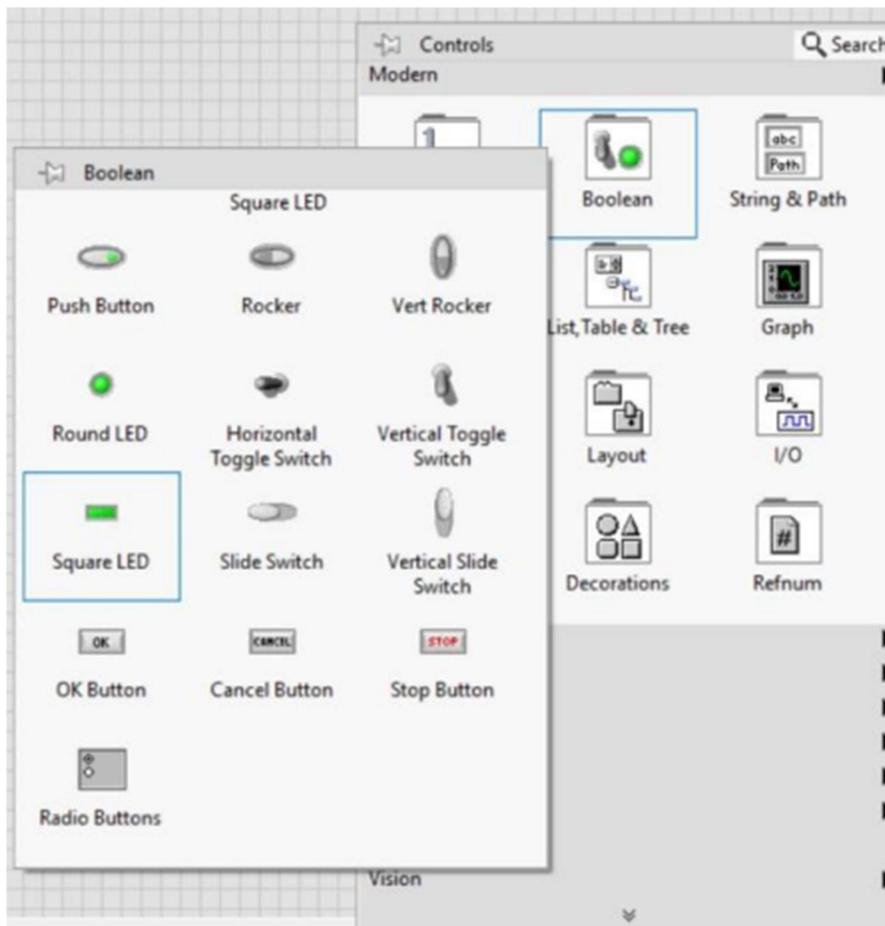
Aquí vamos a tener disponibles dos ventanas, una será para realizar la programación en bloques de nuestro sistema (lado izquierdo) y otra para poder agregar, visualizar y personalizar nuestros elementos a ocupar (lado derecho).



En nuestro sistema utilizaremos:

✓ 3 Square LED

Son tres debido a que cada uno servirá para indicar los tres niveles del tanque (bajo, medio y alto). Para agregarlos solo damos click derecho en cualquier parte de la ventana derecha, elegimos “Boolean” - “Square LED”. Este paso lo hacemos tres veces para así agregar los tres elementos que necesitamos.



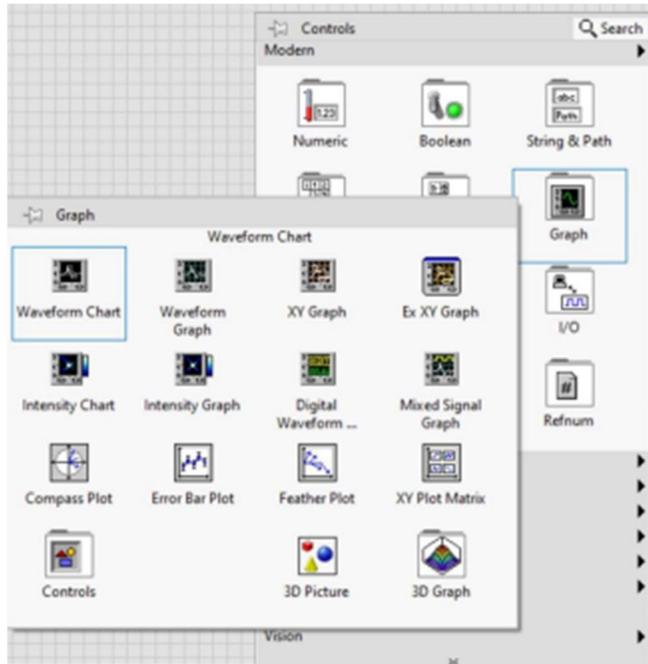
✓ 1 Termometro

Para agregar este elemento, al igual que en el punto anterior, debemos dar click derecho en cualquier parte de la ventana “Front Panel”, “Numeric” – “Tank”.

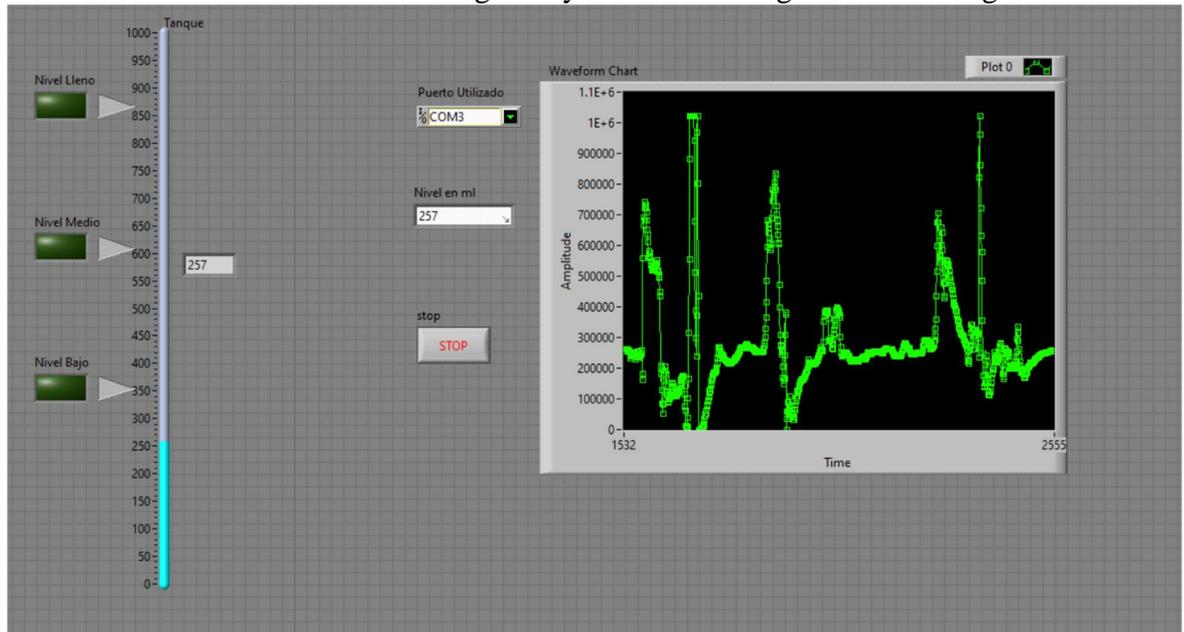
NOTA: El tanque que utilizamos varia en cuestiones de estilización ya que fue editado de manera personal pero el principio es el mismo.

✓ 1 Wavefrom Chart

Volvemos a repetir los pasos anteriores, pero con la diferencia que en vez de escoger “Numeric” – “Tank”, elegiremos “Graph” - “Wavefrom Chart”.

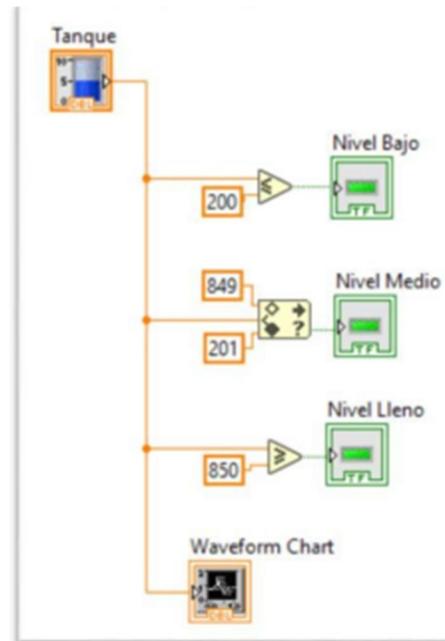
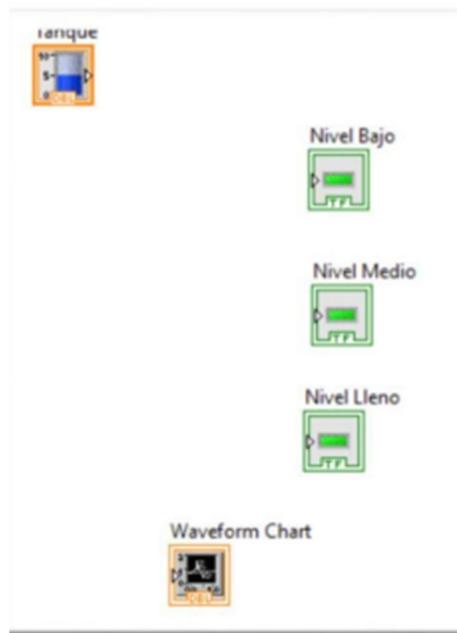


Finalmente ordenamos a nuestro gusto y obtenemos algo como lo siguiente:

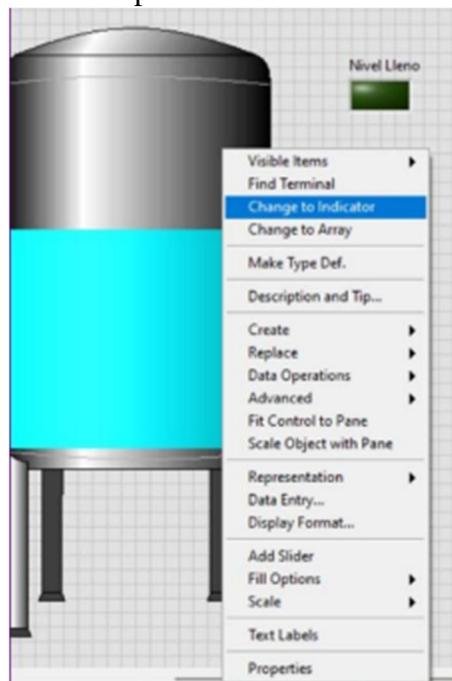


Empezaremos la programación

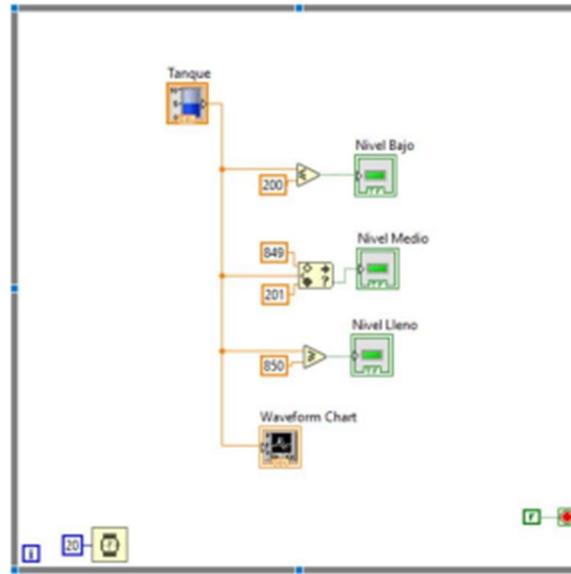
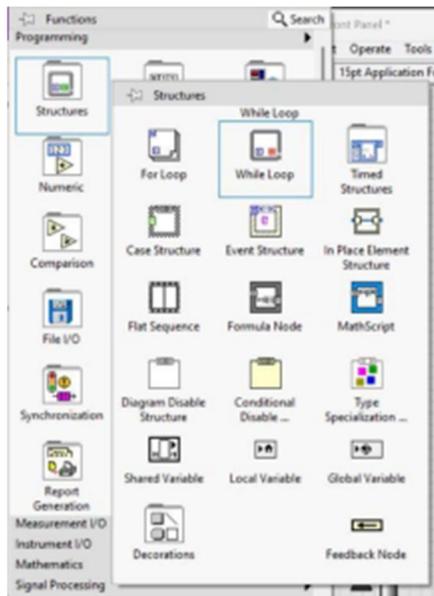
Ahora nos enfocaremos en la ventana “Block Diagram”, donde básicamente nos encontraremos con los bloques de los elementos agregados para así configurarlos y realizar las conexiones pertinentes.



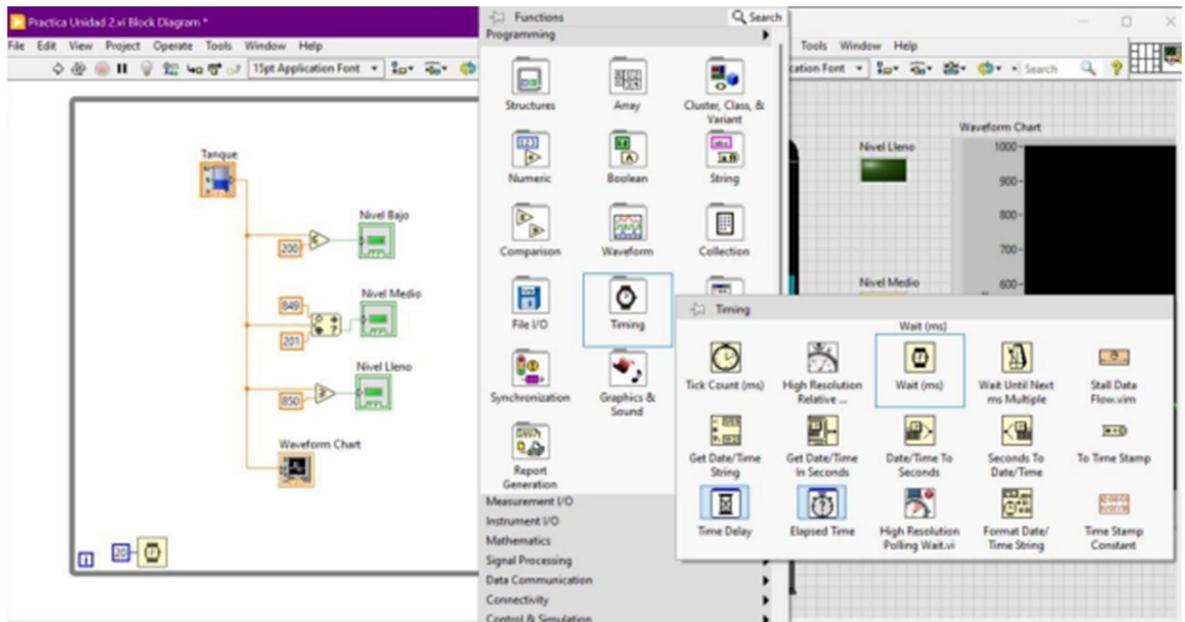
Como se puede observar, en la imagen de lado derecho solo se agregan unos comparadores para poder delimitar los rangos que corresponderán a los límites de los niveles especificados.



Aquí se realiza una configuración ya que el tanque se nos muestra como indicador, y nosotros lo que queremos es que sea un controlador. Para cambiar esto, debemos dar click derecho sobre el tanque y seleccionar “Change to Indicator” Finalmente encerramos este conjunto de diagramas dentro de una “Structure” “while Loop”

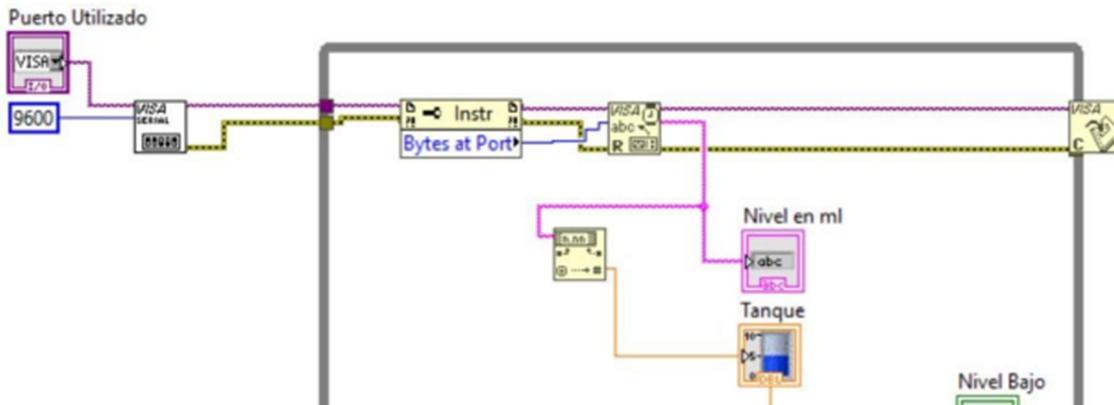
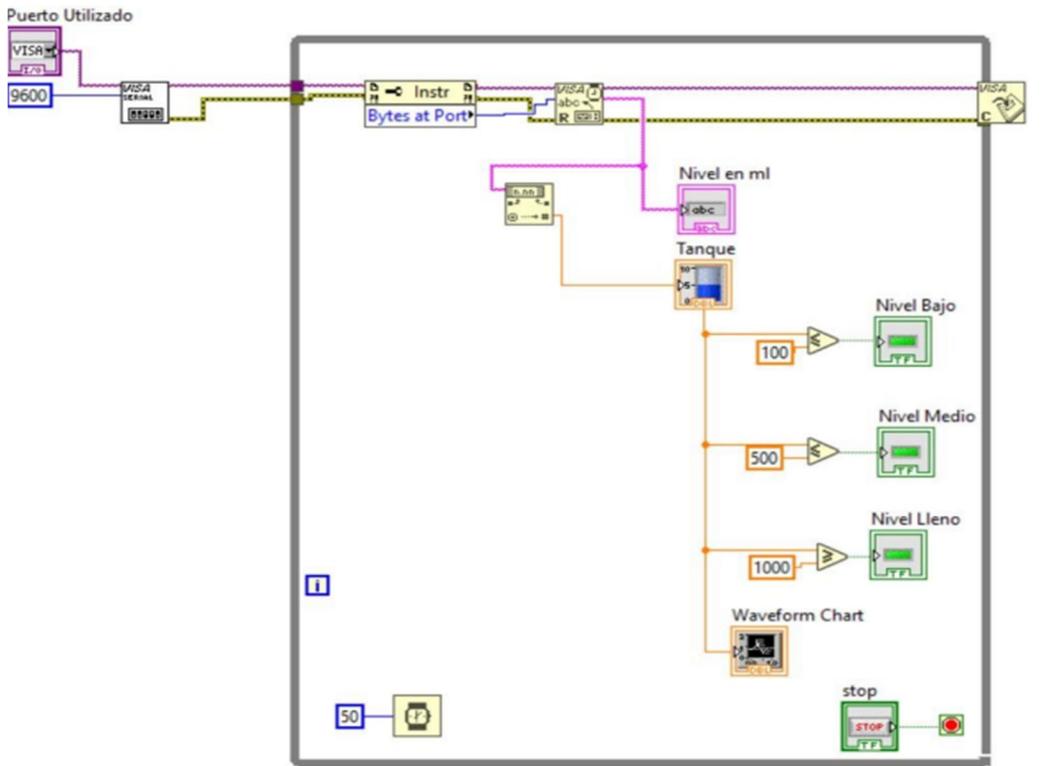


En este último paso, agregamos un “Wait (ms)” en el apartado “Timing”.



## RESULTADOS

A continuación, como una variación del sistema utilizaremos un potenciómetro para que actúe como el llenado del tanque, para ello realizaremos unos pequeños cambios a la programación a bloques, apoyándonos de la plataforma de arduino.



Como se puede apreciar en el diagrama de bloques, la única parte que se agregó fue para poder enlazar Arduino con el programa “LabVIEW”. En pocas palabras lo que se hizo fue agregar un bloque que permitiera la comunicación entre las plataformas antes mencionadas, el problema aquí es que Arduino envía los datos en forma no numéricos (String), entonces para corregir esto, se agregó otro bloque que permitiera esa conversión y así poder visualizar bien los datos inferidos directo a nuestro Tanque.

El código utilizado en Arduino fue el siguiente:

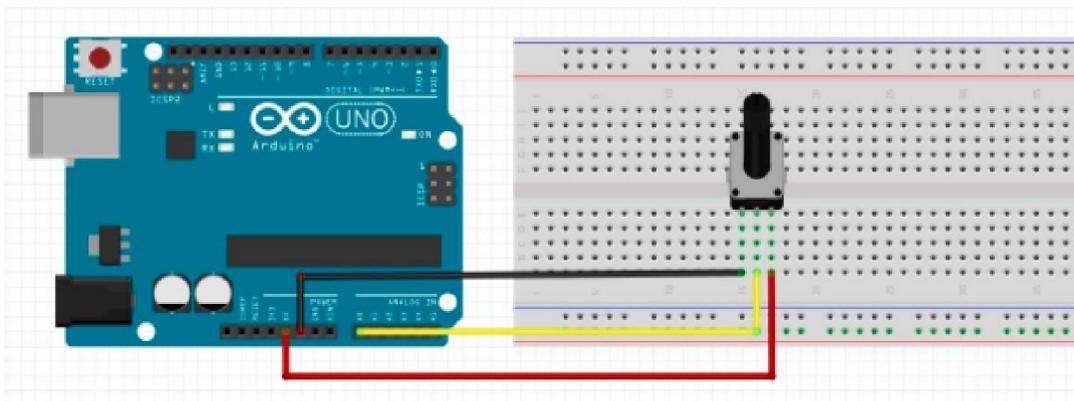
```
int valorA0;

void setup() {
  Serial.begin(9600);
  pinMode(A0, INPUT);
}

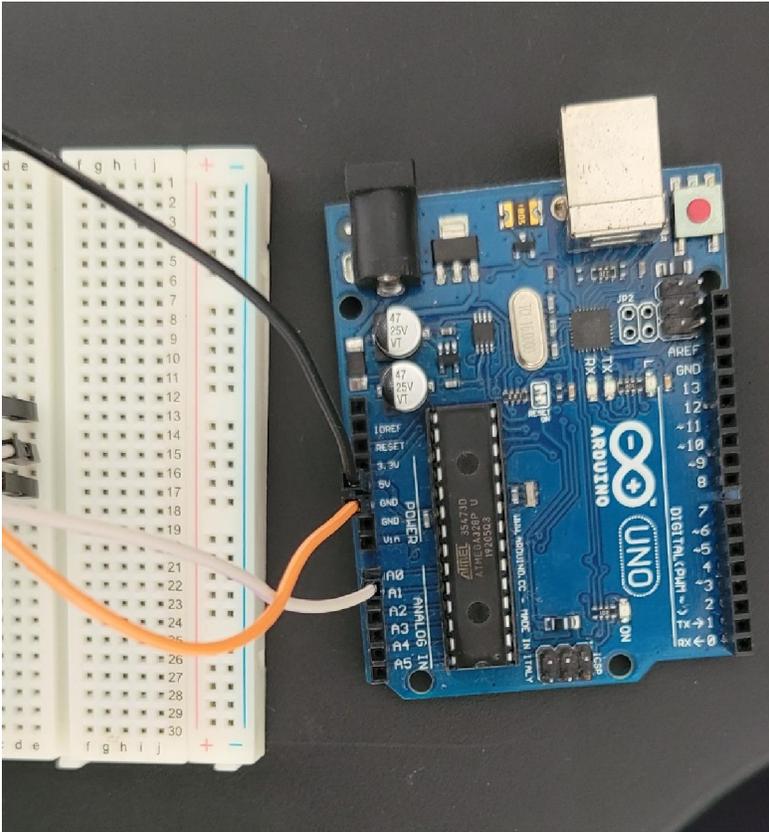
void loop() {
  valorA0 = analogRead(A0);
  Serial.println(valorA0);
  delay(50);
}
```

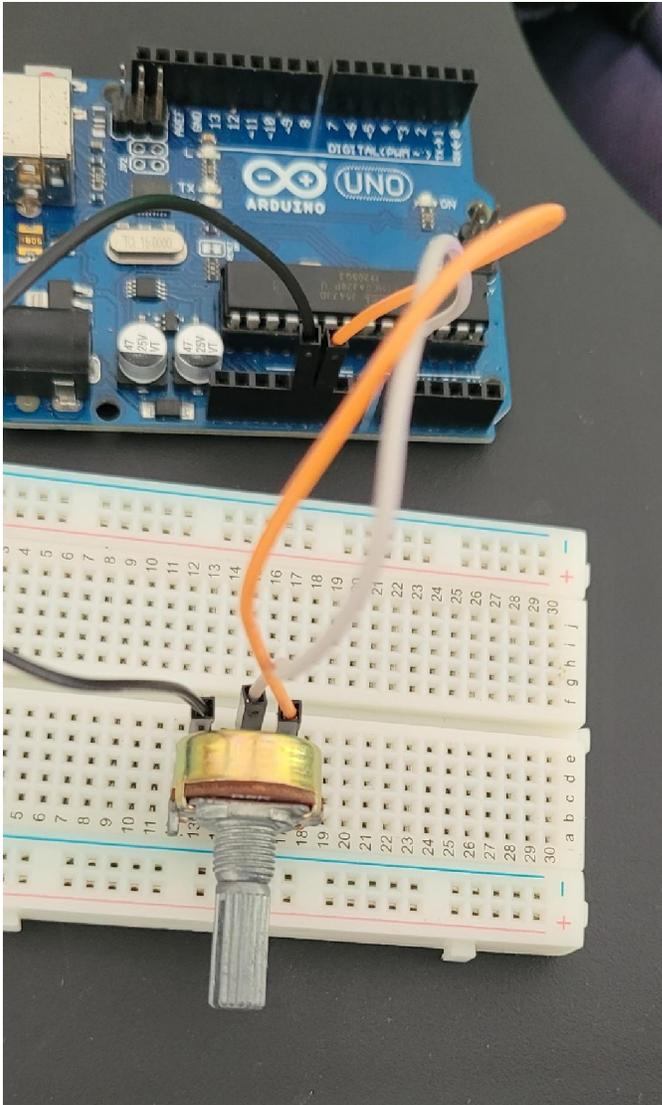
Como se puede apreciar, el código es sumamente sencillo ya que lo que hace el programa es básicamente inferir y leer los datos generados por el potenciómetro conectado al pin A0 de nuestra tarjeta “Arduino Uno”. Como datos adicionales podemos resaltar que la velocidad de transmisión de datos es de 9600 con un retardo de 50 milisegundos.

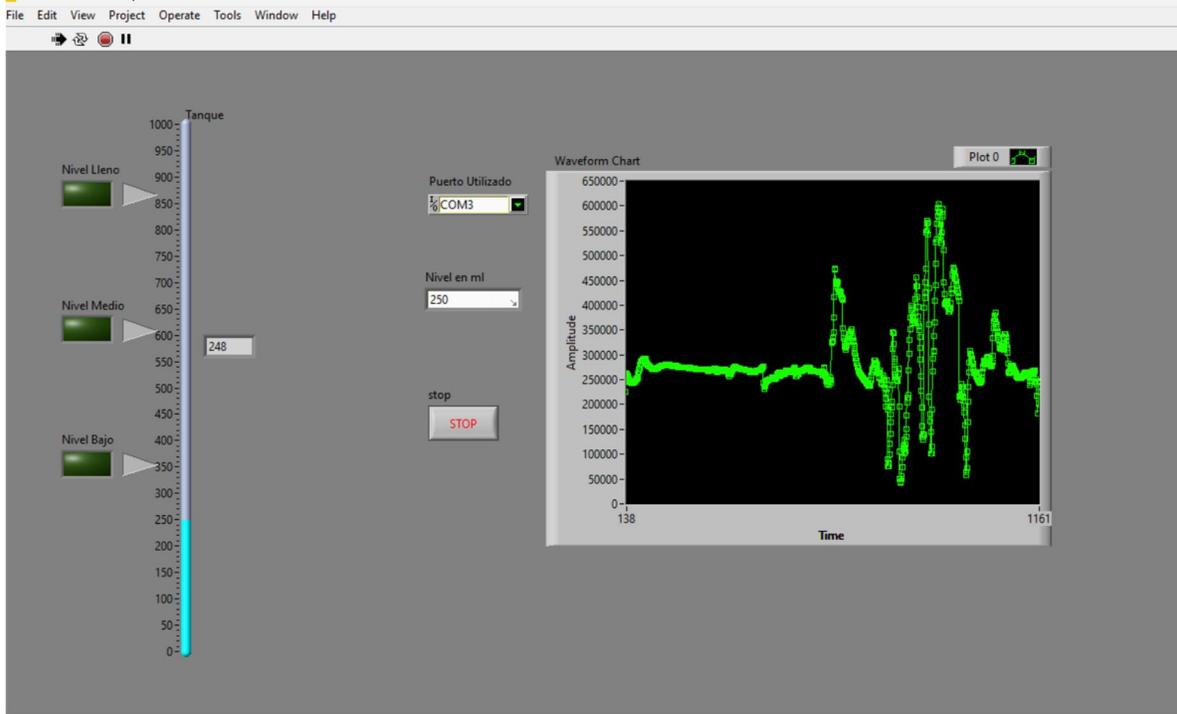
Ahora continuaremos con el circuito realizado:



Para realizar el circuito, se hizo uso del programa “Fritzing” y cómo podemos observar también es algo sencillo puesto que solo ocupamos un potenciómetro y este solo se conecta al pin A0, así también como a tierra y a una fuente de alimentación de 5V provenientes de la misma placa







# LISTA DE COTEJO INVESTIGACION

## INSTRUMENTACIÓN VIRTUAL.

Nombre del estudiante: Arantza Guadalupe García Zapot.

Tema: Elementos de adquisición de datos.

Desarrollo de temas	15	10
Entrega en tiempo y forma	5	5
Claridad en la información	10	7
Total	30 %	22 %

# LISTA DE COTEJO DE PRÁCTICAS

## INSTRUMENTACIÓN VIRTUAL.

### PRÁCTICA NÚMERO 3.

Nombre del estudiante: Arantza Guadalupe García Zapot.

Tema: Elementos de adquisición de datos.

Portada	2 %	2 %
Introducción	5 %	5 %
Desarrollo	20 %	10 %
Conclusiones	5 %	5 %
Referencias	3 %	3 %
Entrega en tiempo y forma	5 %	5 %
Total	40 %	30 %

27/12/24

### \* 3.1 Plataformas de software \*

Juegan un papel crucial en este proceso al proporcionar herramientas para el diseño, simulación y despliegue de instrumentación virtual.

Estas plataformas ofrecen una amplia gama de funcionalidades, desde la creación de interfaces de usuario personalizadas hasta la adquisición y procesamiento de datos en tiempo real.

Permiten a los usuarios diseñar sistemas de instrumentación adaptados a sus necesidades específicas, sin depender de hardware costoso o limitado.

Entre las más populares se encuentran LabView, Matlab, Simulink, Python con bibliotecas como Numpy y National Instrument Measurement Studio.

Estas herramientas proporcionan entornos de desarrollo intuitivos y potentes que permiten a los ingenieros crear, depurar y desplegar sistemas de instrumentación

### 3.1.1

## Lab View

6/3/21

Lab View es un entorno de programación gráfica que proporciona aceleradores de productividad únicos para el desarrollo de sistemas de pruebas, como un enfoque intuitivo para programación, conectividad con cualquier instrumento e interfaces de usuario completamente integradas.

LabView tiene lo que usted necesita para construir rápidamente sistemas de pruebas automatizados.

- \* Puede conectarse a cualquier instrumento, independientemente del proveedor.
- \* LabView tiene una interfaz de usuario nativa para monitorear y controlar los pruebas.
- \* Lab View tiene miles de funciones de análisis de ingeniería.
- \* Lab View funciona con lenguajes populares como Python, C y .NET

### 3.1.2 Matlab

1/2  
3/4

Matlab es un entorno de programación que combina un entorno de escritorio perfeccionado para el análisis interactivo y los procesos de diseño con un lenguaje de programación que expresa las matemáticas de matrices y arrays directamente.

Las apps de MATLAB permiten ver como funcionan diferentes algoritmos con sus datos.

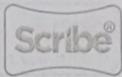
Realiza interacciones hasta obtener los resultados deseados y despues genere automaticamente un programa de MATLAB para reproducir o automatizar el trabajo realizado.

Es un lenguaje que se considera facil de usar.

El MATLAB es muy utilizado en el analisis de datos, graficas, programación, etc.

### 3.1.3 Python

1/2  
1/3/24



Python es un entorno de programación muy utilizado por todas las ventajas que nos ofrece.

Los sistemas de adquisición de datos son indispensables en el modelo matemático de sistemas de control, pues permiten obtener la curva característica por métodos experimentales.

Con el objetivo de desarrollar un sistema que permita las mediciones de voltaje y la posterior obtención de dichos modelos, se utiliza una aplicación visual en Python y se diseñó un protocolo basada en comandos a través del puerto serie con una placa de Arduino.

Python es un lenguaje de programación multiplataforma de código libre, poderoso y fácil de aprender.

Cuenta con estructuras de datos eficientes y de alto nivel, así como un enfoque simple pero efectivo en la programación orientada a objetos.

6/13/24

### \* 3.2 Diseño Hombre - Máquina \*

Una interfaz Hombre - Máquina consiste en algo tan sencillo como un canal de comunicación entre el usuario y el ordenador. Este puede ser unidireccional (unicamente un sentido de comunicación) o bidireccional (envío y recepción de datos en ambos sentidos).

El aumento de la capacidad de procesamiento y memoria de los dispositivos, ha permitido interfaces gráficas mucho más parecidas a la realidad y también la incorporación de nuevos canales de entrada, como sensores, habla, video, táctiles...

Los campos más destacados que utilizando interfaces hombre-máquina actualmente son:

- \* Multimodalidad
- \* Realidad aumentada
- \* Interfaces adaptativas
- \* Aspectos emocionales.