

# **INSTITUTO TECNOLÓGICO SUPERIOR DE SAN ANDRÉS TUXTLA**

**CARRERA  
INGENIERIA MECATRONICA**

**ASIGNATURA  
INSTRUMENTACION VIRTUAL**

**DOCENTE  
JOSE ANGEL NIEVES VAQUES**

**UNIDAD: 2**

**GRUPO: 711-A**

**FECHA: 19/10/2022**

## **INTEGRANTES**

- ✓ **ACOSTA GUILLEN ÁNGEL DANIEL- 191U0419**
- ✓ **CONDE ROQUE JONATHAN- 191U0470**
- ✓ **MARTÍNEZ HERNÁNDEZ JUAN ALBERTO- 191U0453**
- ✓ **PELAYO XOLO LUIS MIGUEL-191U0461**
- ✓ **POLANCO POLITO ESTEBAN BERNABÉ- 191U0464**



## INTRODUCCIÓN

En un proceso industrial existirán varios sensores que suministran información, convenientemente acondicionada, al elemento controlador del sistema. El elemento controlador, que estará basado en algún microprocesador, recibirá la información de los sensores directamente o mediante un proceso de comunicación. Además de la presentación de dicha información en la forma deseada (generalmente gráfica) el elemento controlador dará las órdenes oportunas a los actuadores para mantener el proceso funcionando dentro de los márgenes previstos.

Para la instrumentación virtual o los procesos de laboratorio, la información puede venir dada no sólo por sensores, sino también por otros sistemas de medida (osciloscopios, multímetros, etc.) con capacidad de comunicación. Partiendo de la información recogida podemos cambiar las condiciones de la prueba, modificando parámetros de los aparatos (generadores de funciones, fuentes de alimentación).

En un entorno como el descrito, la tendencia actual es que sea un software especializado quien se encargue del control del sistema, coordinando el funcionamiento de los distintos elementos. Uno de estos programas software es LabView de la multinacional National Instruments. Labview permite recoger, analizar y monitorizar los datos dentro de un entorno de programación gráfico en el que se ensamblan objetos llamados instrumentos virtuales (Vis) para formar el programa de aplicación con el que interactuará el usuario y que se denomina instrumento virtual.

Además de lo que es la propia representación de los datos en los paneles interactivos que funcionan como si se tratara de instrumentación real, permite múltiples opciones de manejo de datos, como su almacenamiento en disco y compartirlos en red o con otras aplicaciones. La interacción con otras aplicaciones se podrá realizar mediante llamadas a librerías de enlace dinámico (DLL: Dynamic Link Library) e intercambio dinámico de datos (DDE: Dynamic Data Exchange) en modo local o mediante TCP/IP en conexiones remotas. Siempre buscando independencia de la plataforma en la que hayamos realizado nuestra aplicación.



## ❖ ¿POR QUÉ USAR LABVIEW?

- ✓ LabVIEW es un entorno de desarrollo gráfico con funciones integradas para realizar adquisición de datos, control de instrumentos, análisis de medidas y presentaciones de datos.
- ✓ LabVIEW es un lenguaje potente en un ambiente de programación gráfico, pero mucho más sencillo que los entornos tradicionales.
- ✓ Lenguaje Desarrollado para Medición, Control y Automatización A diferencia de los lenguajes de propósito general, LabVIEW tiene funciones específicas para acelerar el desarrollo de aplicaciones de medición, control y automatización.
- ✓ Fácil Integración con Instrumentos y Dispositivos de Medida LabVIEW.  
Se puede conectar de manera transparente con todo tipo de hardware incluyendo instrumentos, plaquetas adquisidoras, controladores lógicos programables (PLCs).
- ✓ LabVIEW para Investigación y Análisis Puede utilizarse LabVIEW  
Para analizar y registrar resultados reales para aplicaciones en amplios sectores orientados a ingeniería mecánica, eléctrica, electrónica, biomédica, etc.
- ✓ LabVIEW para Control de Procesos y Automatización en Fábricas  
Puede utilizarse LabVIEW para numerosas aplicaciones de control de procesos y automatización, realizar medidas y control de alta velocidad y con muchos canales.

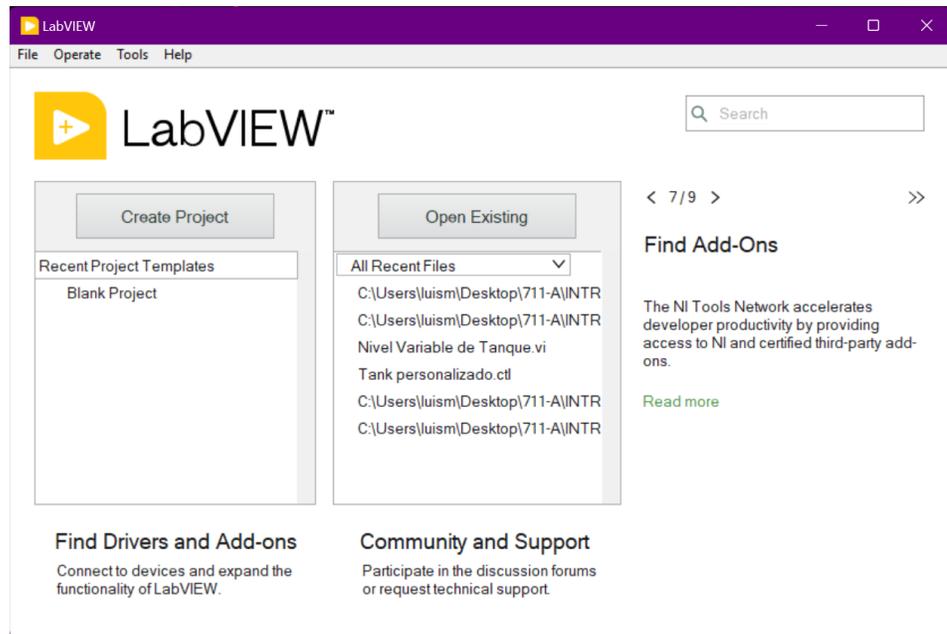
Basándose en los puntos anteriores, es fácil saber la razón por la cual este tipo de software es muy utilizado para el desarrollo de proyectos enfocados a la instrumentación virtual. Teniendo en cuenta esto, se ha procedido a efectuar este programa para la programación y monitoreo de la variable **nivel** como si se llevara a cabo en un proceso industrial real.

## ❖ MATERIALES

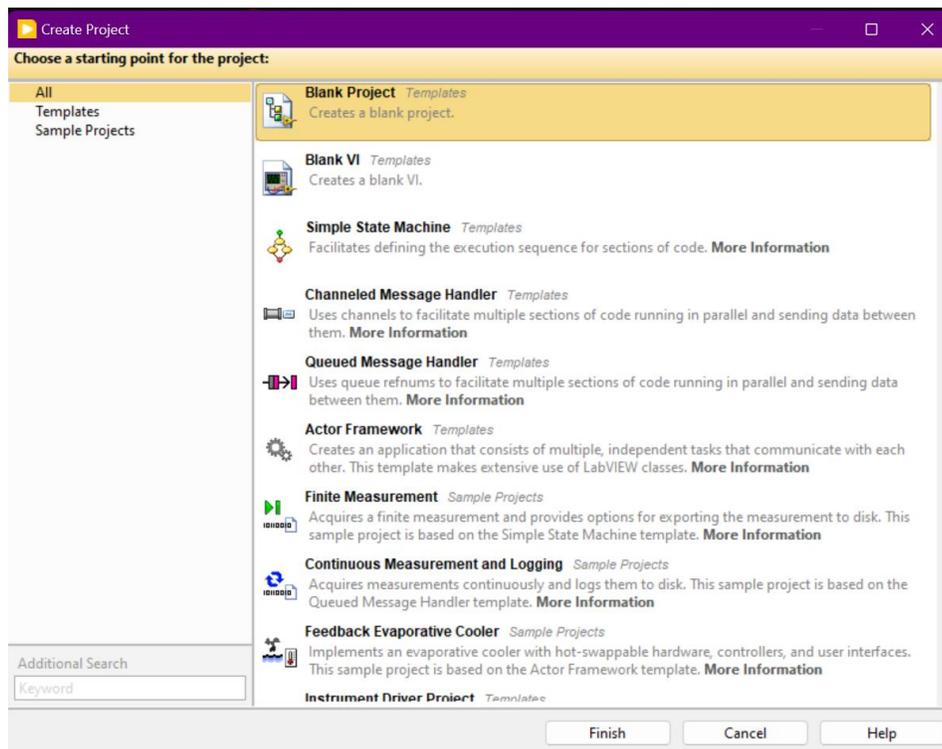
-  PC
-  Software LabVIEW

## ★ DESARROLLO

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).

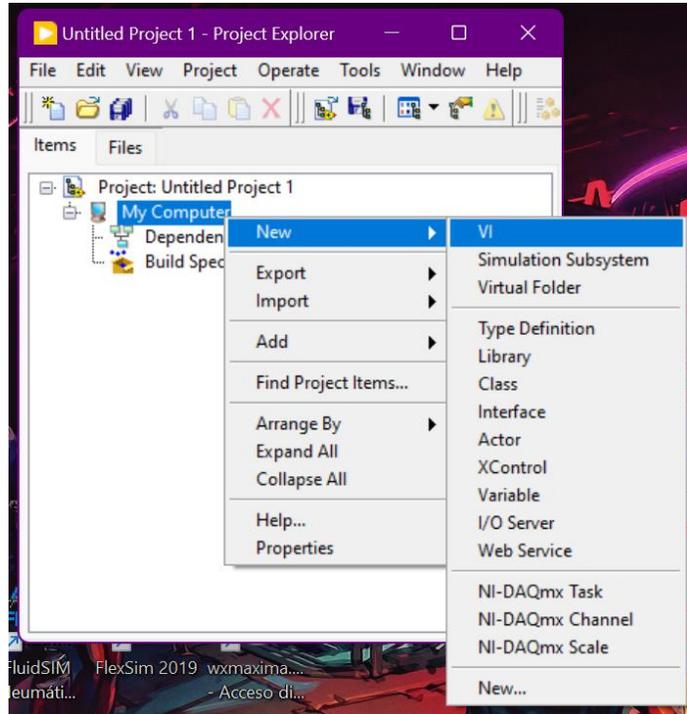


Sucesivamente de ello, pulsaremos dos veces en “Blank Project”

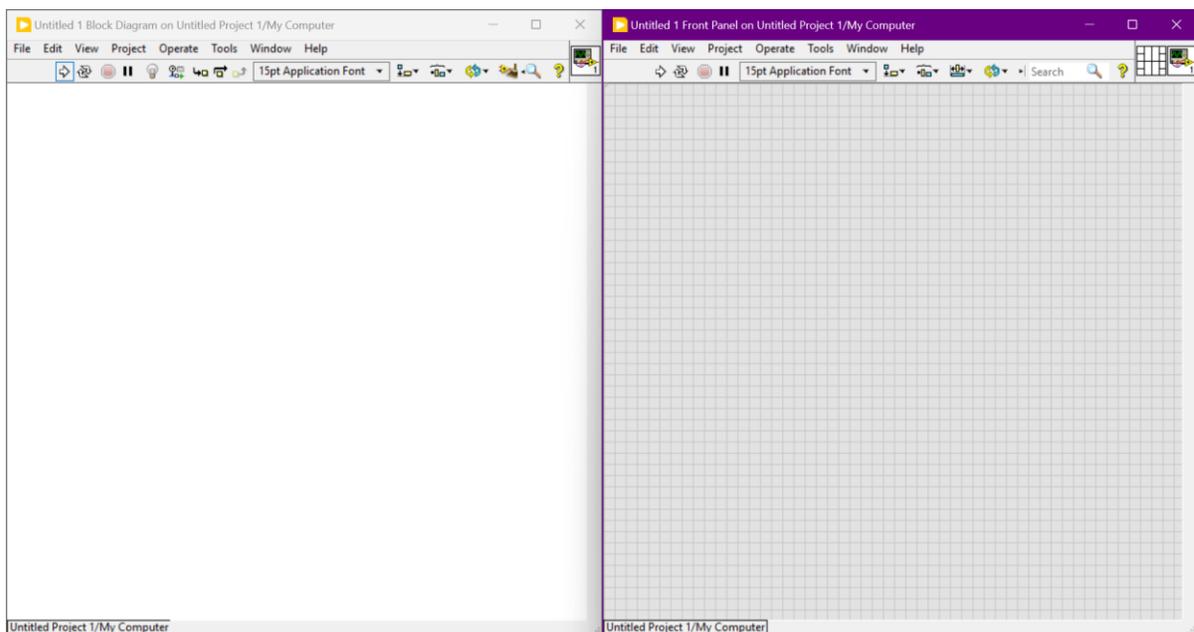


Ahora bien, para comenzar a programar y diseñar nuestro instrumento virtual, debemos dar click derecho sobre “My Computer”, luego seleccionar “New” y finalmente “VI”.

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).

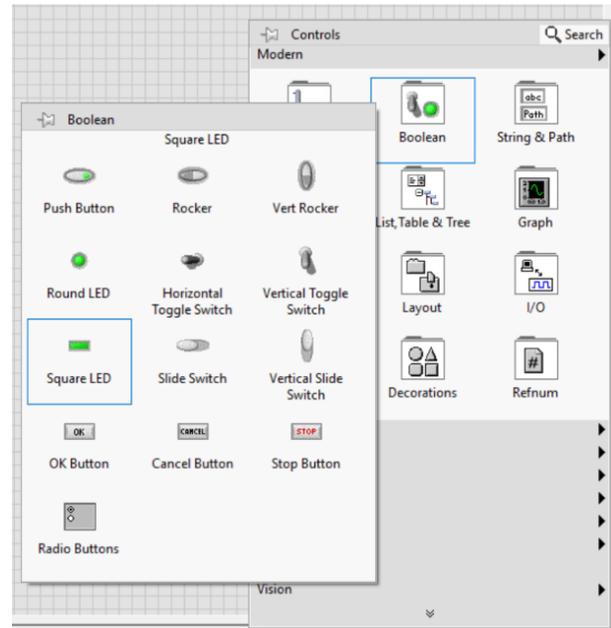


Los elementos que ocuparemos son los siguientes:

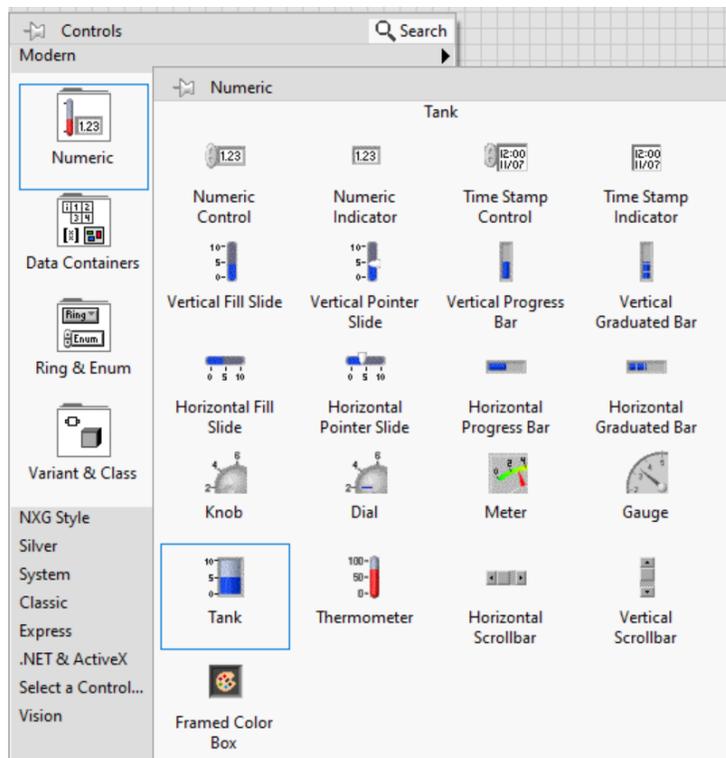
✓ 3 Square LED

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 Tank

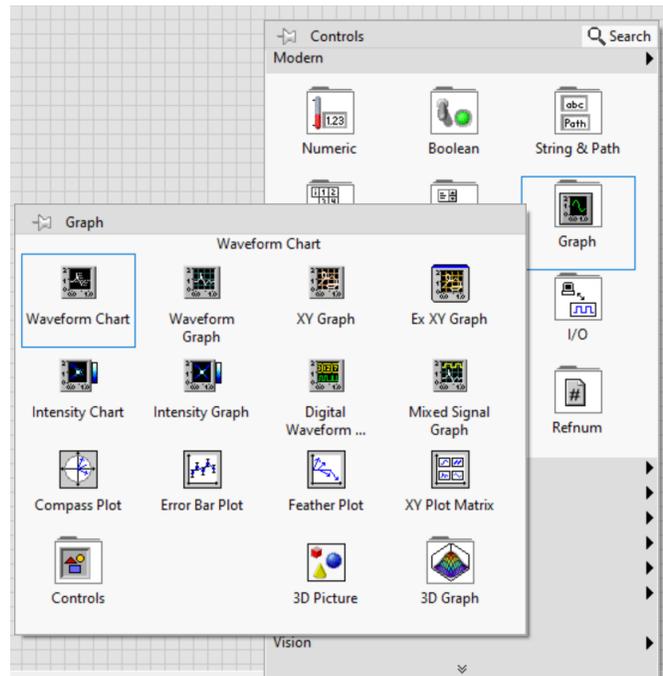


Para agregar este elemento, al igual que en el punto anterior, debemos dar click derecho en cualquier parte de la ventana “Front Panel”, “Numeric” – “Tank”.

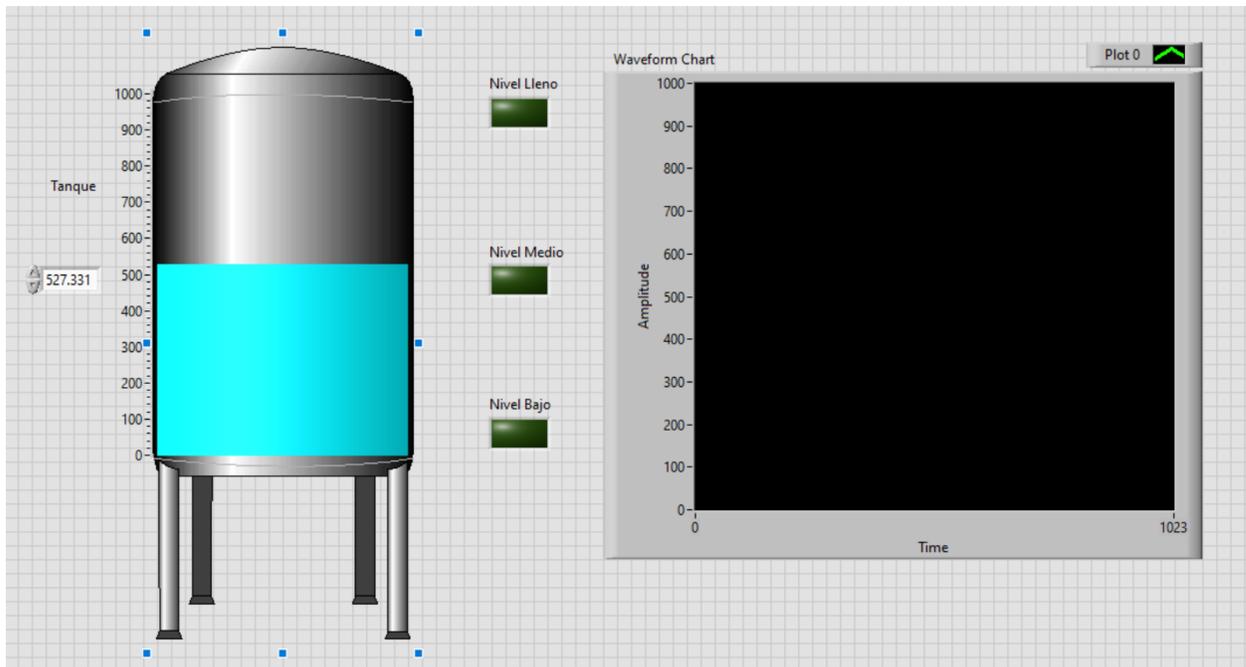
NOTA: El tanque que utilizamos varia en cuestiones de estilización ya que fue editado de manera personal pero el principio es el mismo.

✓ 1 Waveform Chart

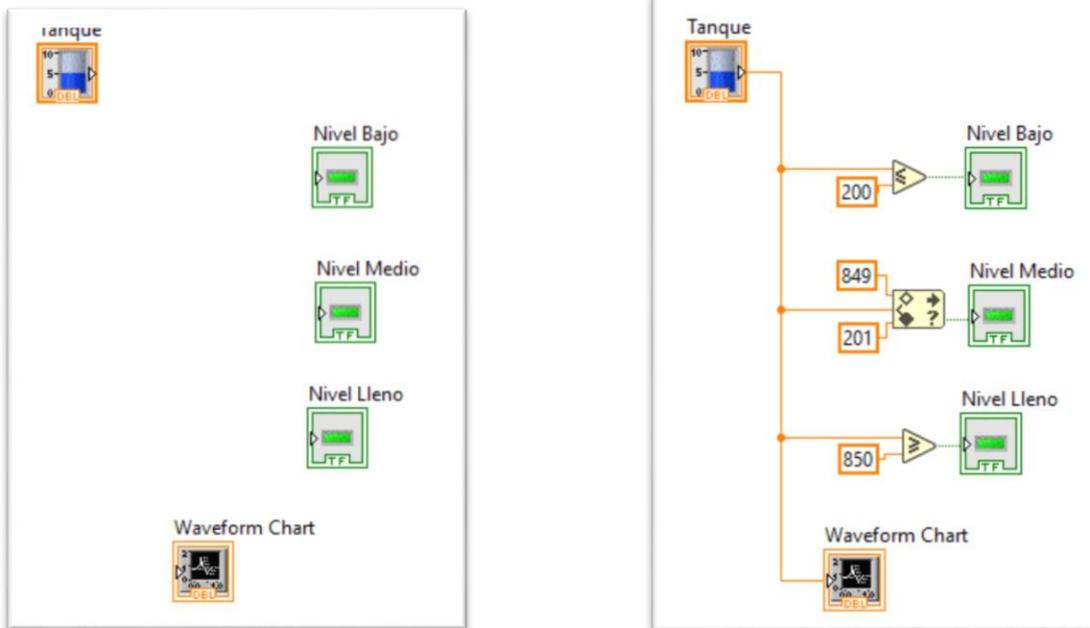
Volvemos a repetir los pasos anteriores pero con la diferencia que en vez de escoger “Numeric” – “Tank”, elegiremos “Graph” - “Waveform Chart”.



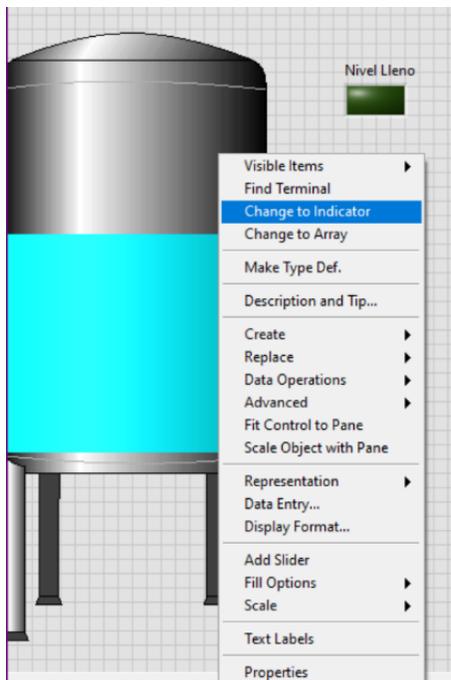
Finalmente ordenamos a nuestro gusto y obtenemos algo como lo siguiente:



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.

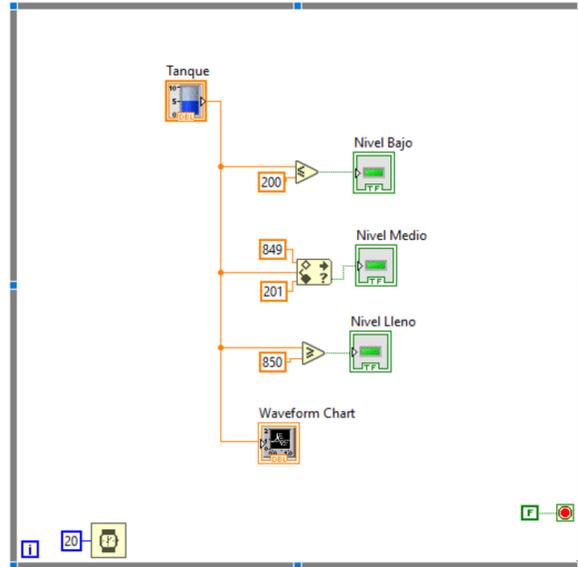
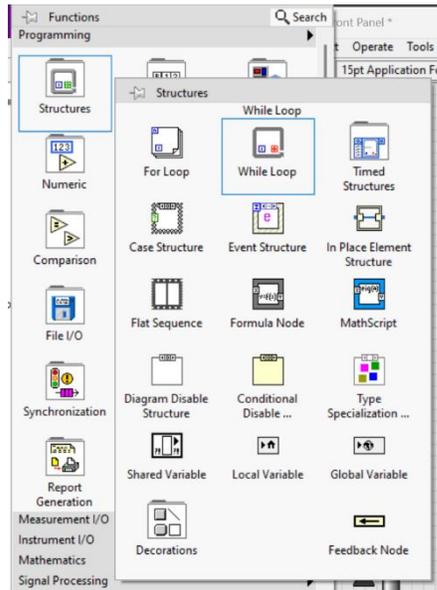


Como se puede observar, en la imagen de lado derecho solo se agregan unos comparadores para poder delimitar los rangos que corresponderán a los límites de los niveles especificados.

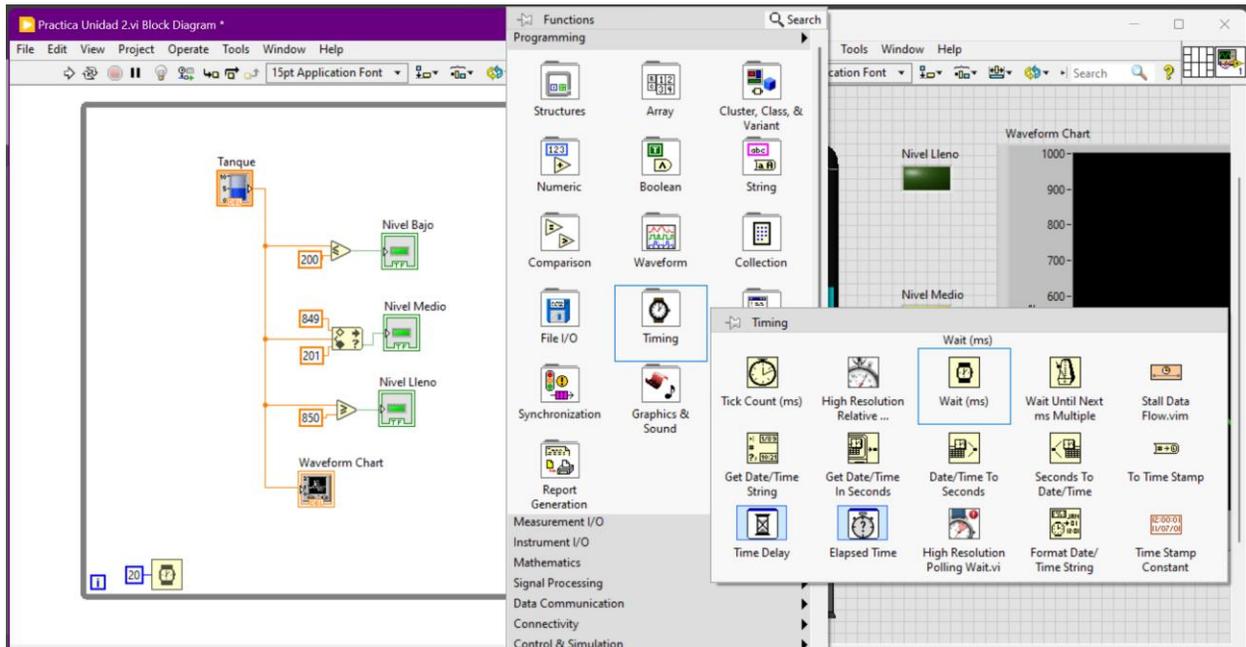


Aquí se realiza una configuración ya que el tanque se nos muestra como indicador, y nosotros lo que queremos es que sea un controlador. Para cambiar esto, debemos dar click derecho sobre el tanque y seleccionar “Change to Indicator”

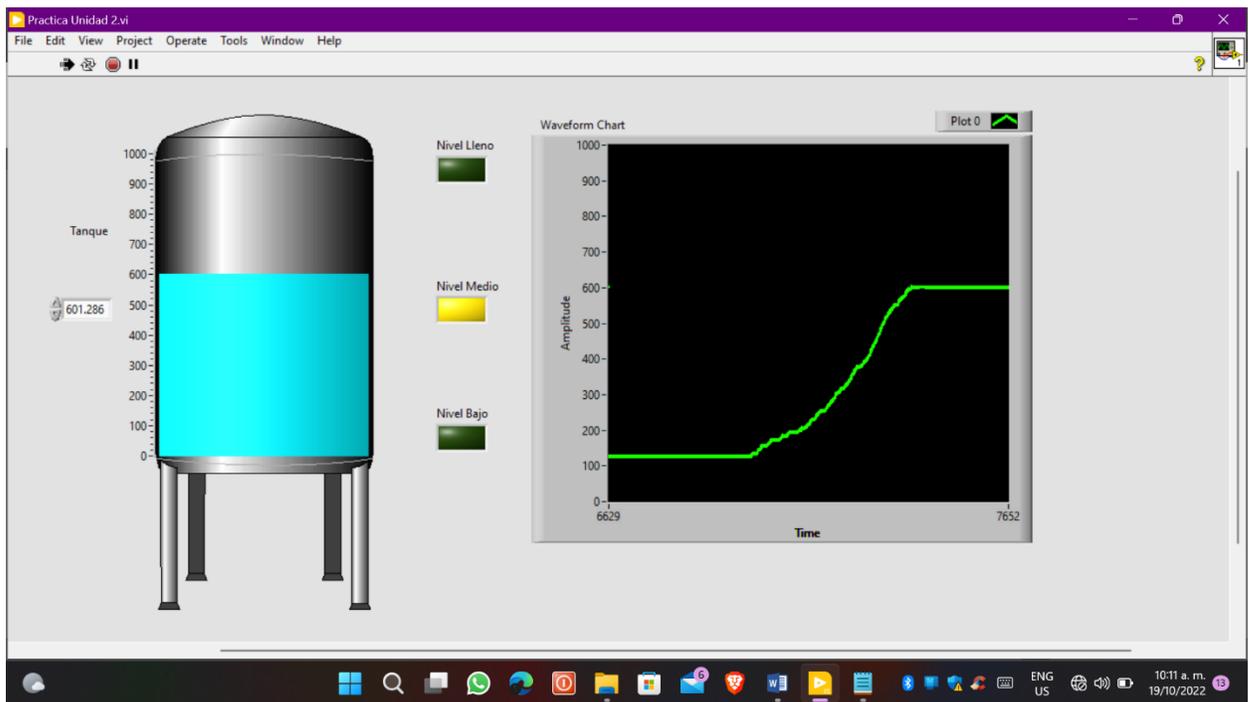
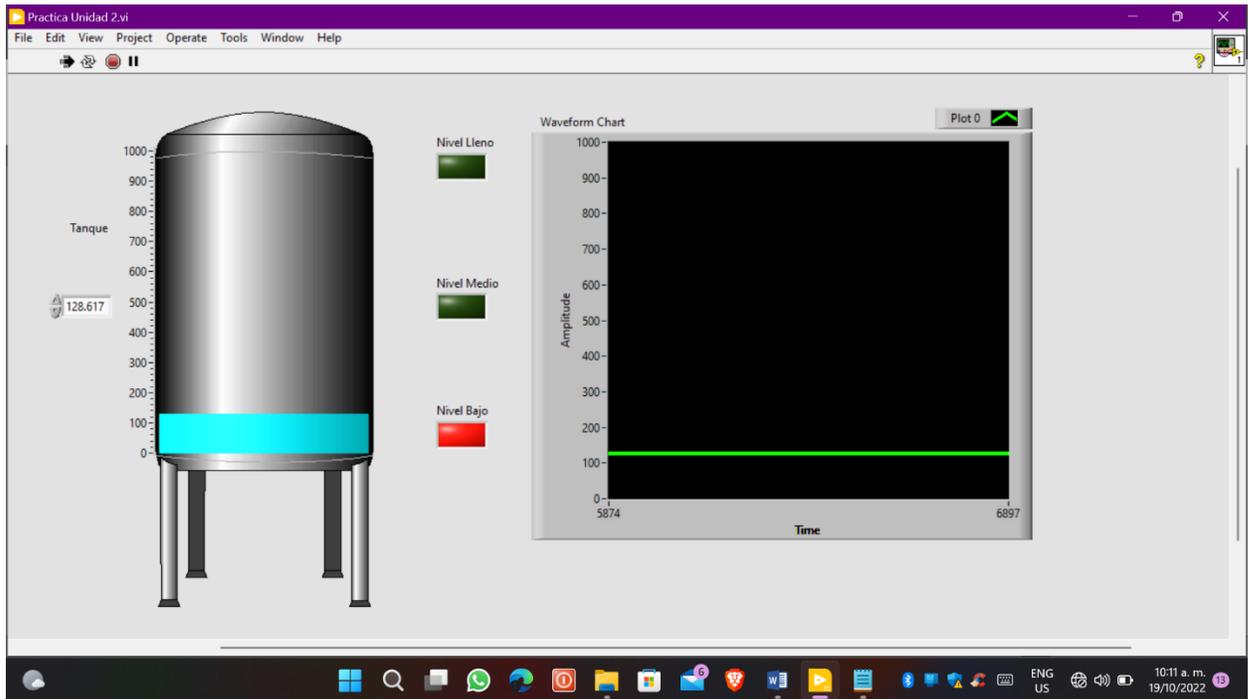
Finalmente encerramos este conjunto de diagramas dentro de una “Structure” “while Loop”

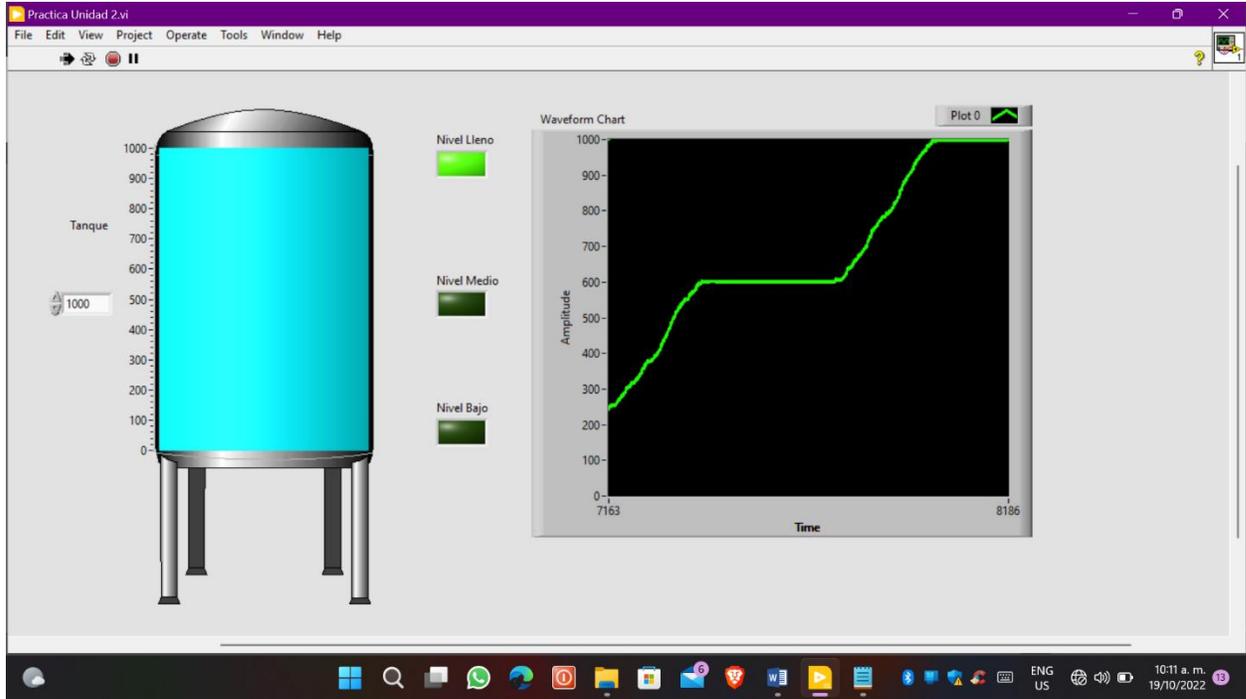


En este último paso, agregamos un “Wait (ms)” en el apartado “Timing”.



## ★ SIMULACION





## ❖ CONCLUSIÓN

Se puede decir que hoy en día la tecnología es una herramienta sumamente importante en las actividades que el ser humano realiza día a día.

Esto debido a que antes de que esta apareciera las formas de trabajar eran un poco más pesadas y difíciles y una forma clara eran las formas de mediciones que estas utilizaban como por ejemplo son: los codos, las brazadas, la vara, el plomo, etc.

En los últimos años han aparecido diferentes softwares que nos han facilitado aprender y así llevar a cabo actividades que parecían difíciles e imposibles, unos de estos ha sido LabVIEW el cual es un software que nos proporciona un ambiente gráfico sumamente importante y en el cual podemos realizar distintos tipos de actividades y en cual aprendimos a diseñar y a medir lo que fue un tanque de agua y así mismo interactuar con el nivel del agua.



**EDUCACIÓN**  
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO  
NACIONAL DE MÉXICO



INSTITUTO TECNOLÓGICO SUPERIOR DE  
SAN ANDRÉS TUXTLA



VERACRUZ  
GOBIERNO  
DEL ESTADO



SEV  
Secretaría  
de Educación

SEMSyS  
Subsecretaría de Educación  
Media Superior y Superior



DET  
Departamento de Educación  
Tecnológica del Estado

## ★ BIBLIOGRAFÍA

<https://youtu.be/ejC3pLn8V68>

# **INSTITUTO TECNOLÓGICO SUPERIOR DE SAN ANDRÉS TUXTLA**

**CARRERA  
INGENIERIA MECATRONICA**

**ASIGNATURA  
INSTRUMENTACION VIRTUAL**

**DOCENTE  
JOSE ANGEL NIEVES VAQUES**

**UNIDAD: 2**

**GRUPO: 711-A**

**FECHA: 19/10/2022**

## **INTEGRANTES**

- ✓ **ACOSTA GUILLEN ÁNGEL DANIEL- 191U0419**
- ✓ **CONDE ROQUE JONATHAN- 191U0470**
- ✓ **MARTÍNEZ HERNÁNDEZ JUAN ALBERTO- 191U0453**
- ✓ **PELAYO XOLO LUIS MIGUEL-191U0461**
- ✓ **POLANCO POLITO ESTEBAN BERNABÉ- 191U0464**



## INTRODUCCIÓN

En un proceso industrial existirán varios sensores que suministran información, convenientemente acondicionada, al elemento controlador del sistema. El elemento controlador, que estará basado en algún microprocesador, recibirá la información de los sensores directamente o mediante un proceso de comunicación. Además de la presentación de dicha información en la forma deseada (generalmente gráfica) el elemento controlador dará las órdenes oportunas a los actuadores para mantener el proceso funcionando dentro de los márgenes previstos.

Para la instrumentación virtual o los procesos de laboratorio, la información puede venir dada no sólo por sensores, sino también por otros sistemas de medida (osciloscopios, multímetros, etc.) con capacidad de comunicación. Partiendo de la información recogida podemos cambiar las condiciones de la prueba, modificando parámetros de los aparatos (generadores de funciones, fuentes de alimentación).

En un entorno como el descrito, la tendencia actual es que sea un software especializado quien se encargue del control del sistema, coordinando el funcionamiento de los distintos elementos. Uno de estos programas software es LabView de la multinacional National Instruments. Labview permite recoger, analizar y monitorizar los datos dentro de un entorno de programación gráfico en el que se ensamblan objetos llamados instrumentos virtuales (Vis) para formar el programa de aplicación con el que interactuará el usuario y que se denomina instrumento virtual.

Además de lo que es la propia representación de los datos en los paneles interactivos que funcionan como si se tratara de instrumentación real, permite múltiples opciones de manejo de datos, como su almacenamiento en disco y compartirlos en red o con otras aplicaciones. La interacción con otras aplicaciones se podrá realizar mediante llamadas a librerías de enlace dinámico (DLL: Dynamic Link Library) e intercambio dinámico de datos (DDE: Dynamic Data Exchange) en modo local o mediante TCP/IP en conexiones remotas. Siempre buscando independencia de la plataforma en la que hayamos realizado nuestra aplicación.



## ❖ ¿POR QUÉ USAR LABVIEW?

- ✓ LabVIEW es un entorno de desarrollo gráfico con funciones integradas para realizar adquisición de datos, control de instrumentos, análisis de medidas y presentaciones de datos.
- ✓ LabVIEW es un lenguaje potente en un ambiente de programación gráfico, pero mucho más sencillo que los entornos tradicionales.
- ✓ Lenguaje Desarrollado para Medición, Control y Automatización A diferencia de los lenguajes de propósito general, LabVIEW tiene funciones específicas para acelerar el desarrollo de aplicaciones de medición, control y automatización.
- ✓ Fácil Integración con Instrumentos y Dispositivos de Medida LabVIEW.  
Se puede conectar de manera transparente con todo tipo de hardware incluyendo instrumentos, plaquetas adquisidoras, controladores lógicos programables (PLCs).
- ✓ LabVIEW para Investigación y Análisis Puede utilizarse LabVIEW  
Para analizar y registrar resultados reales para aplicaciones en amplios sectores orientados a ingeniería mecánica, eléctrica, electrónica, biomédica, etc.
- ✓ LabVIEW para Control de Procesos y Automatización en Fábricas  
Puede utilizarse LabVIEW para numerosas aplicaciones de control de procesos y automatización, realizar medidas y control de alta velocidad y con muchos canales.

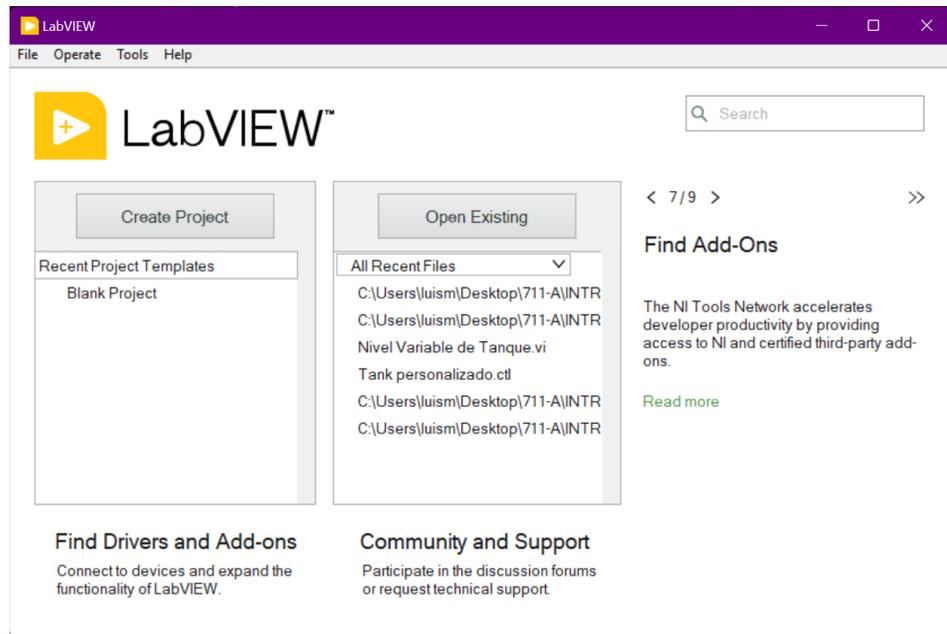
Basándose en los puntos anteriores, es fácil saber la razón por la cual este tipo de software es muy utilizado para el desarrollo de proyectos enfocados a la instrumentación virtual. Teniendo en cuenta esto, se ha procedido a efectuar este programa para la programación y monitoreo de la variable **nivel** como si se llevara a cabo en un proceso industrial real.

## ❖ MATERIALES

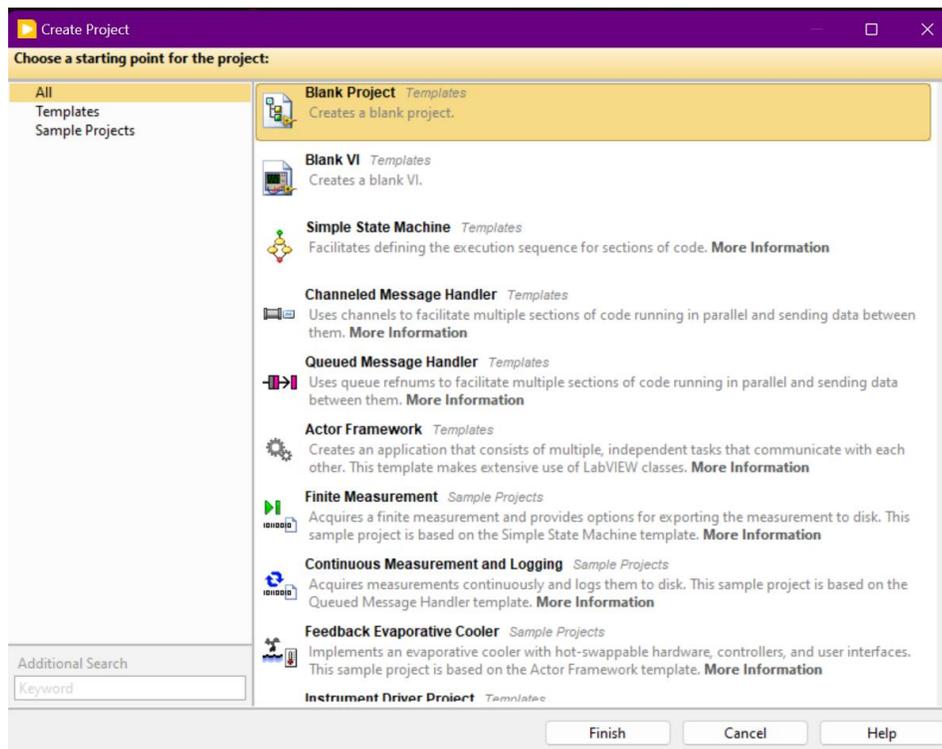
-  PC
-  Software LabVIEW

## ★ DESARROLLO

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).

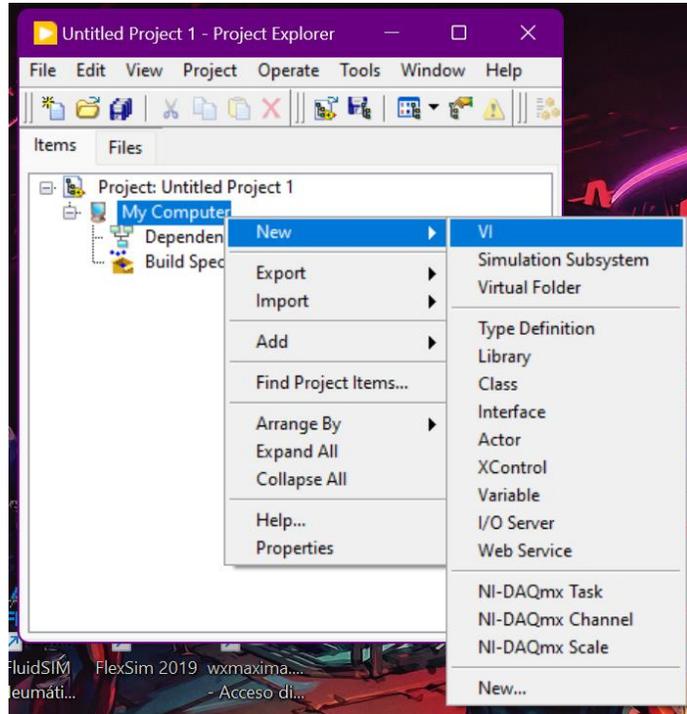


Sucesivamente de ello, pulsaremos dos veces en “Blank Project”

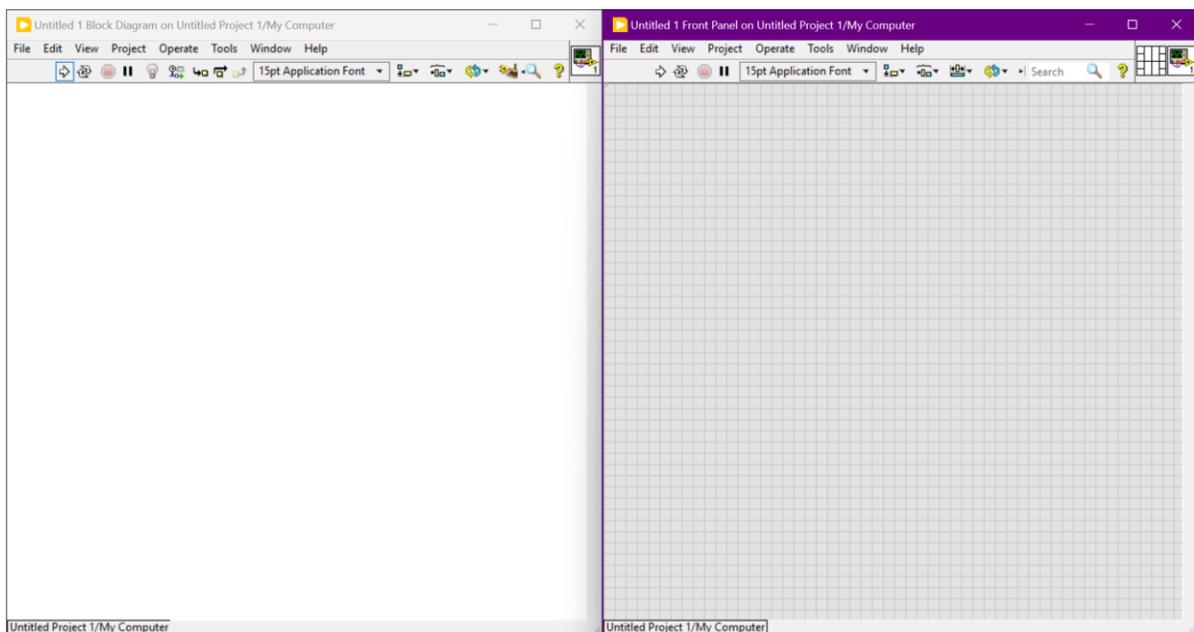


Ahora bien, para comenzar a programar y diseñar nuestro instrumento virtual, debemos dar click derecho sobre “My Computer”, luego seleccionar “New” y finalmente “VI”.

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).

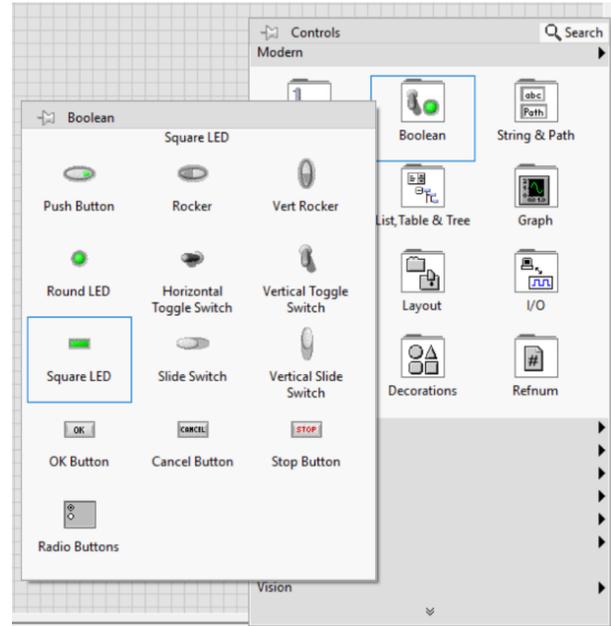


Los elementos que ocuparemos son los siguientes:

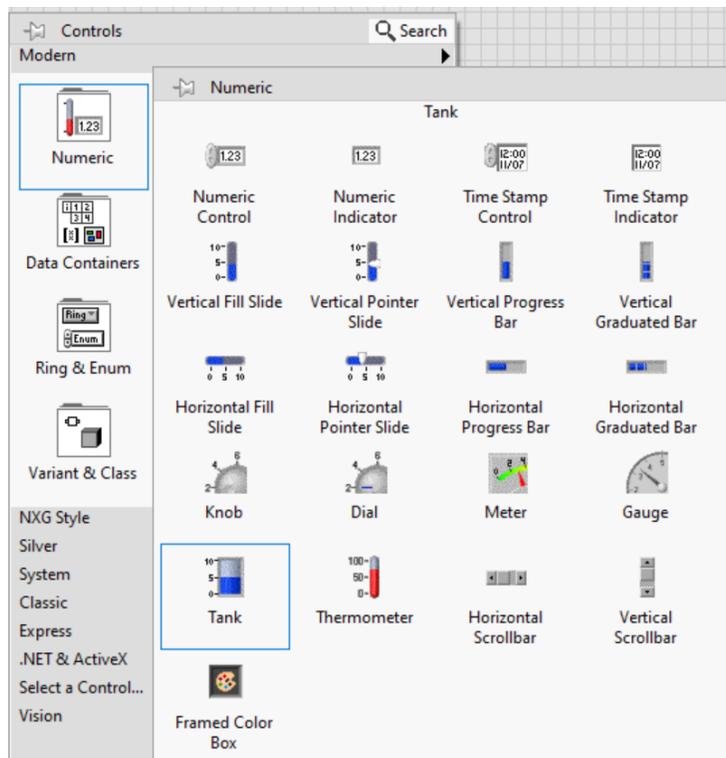
✓ 3 Square LED

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 Tank

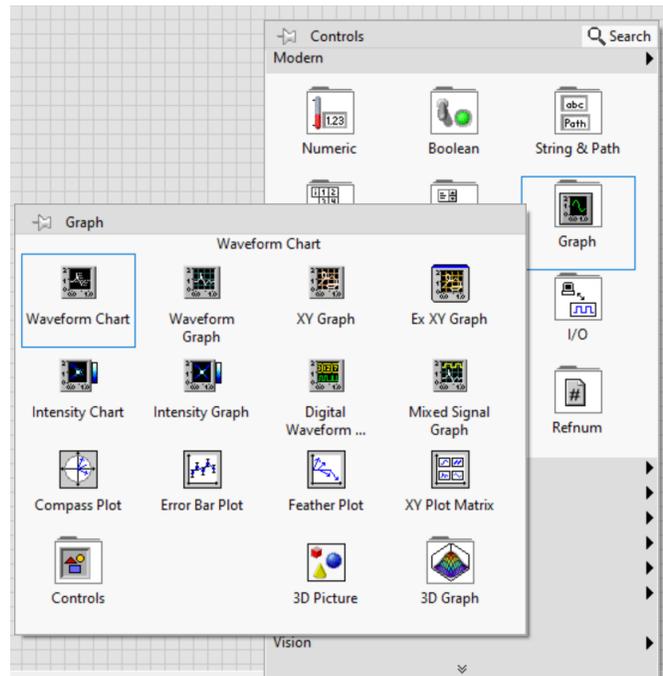


Para agregar este elemento, al igual que en el punto anterior, debemos dar click derecho en cualquier parte de la ventana “Front Panel”, “Numeric” – “Tank”.

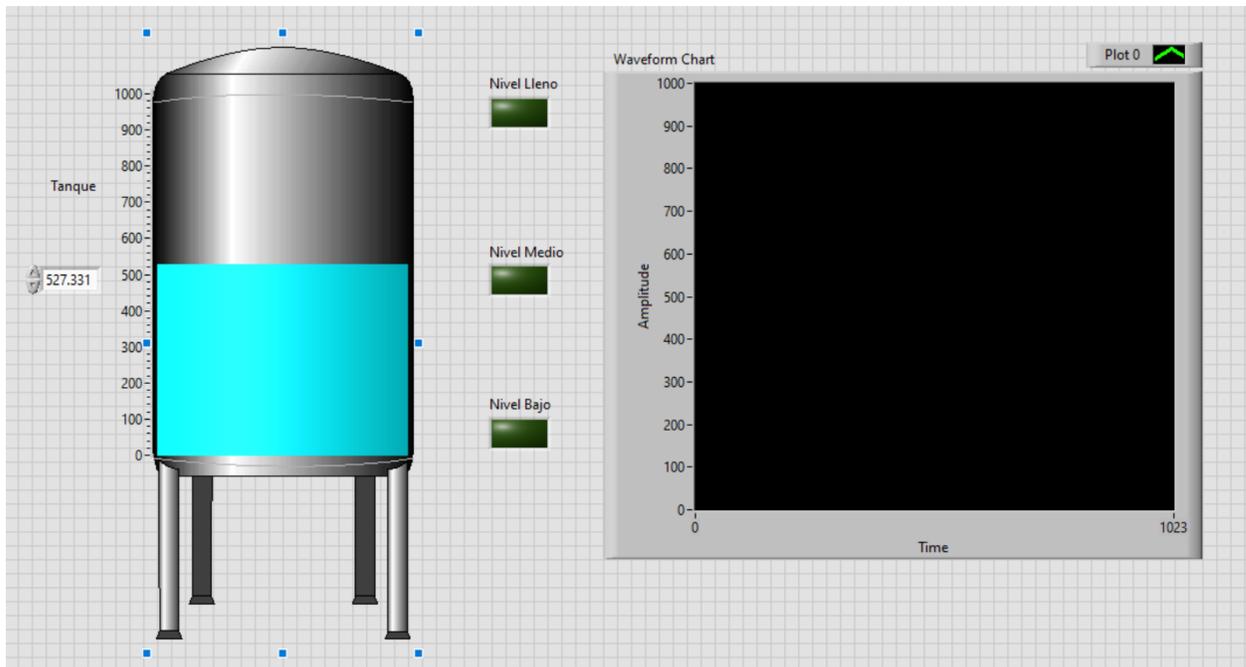
NOTA: El tanque que utilizamos varia en cuestiones de estilización ya que fue editado de manera personal pero el principio es el mismo.

✓ 1 Waveform Chart

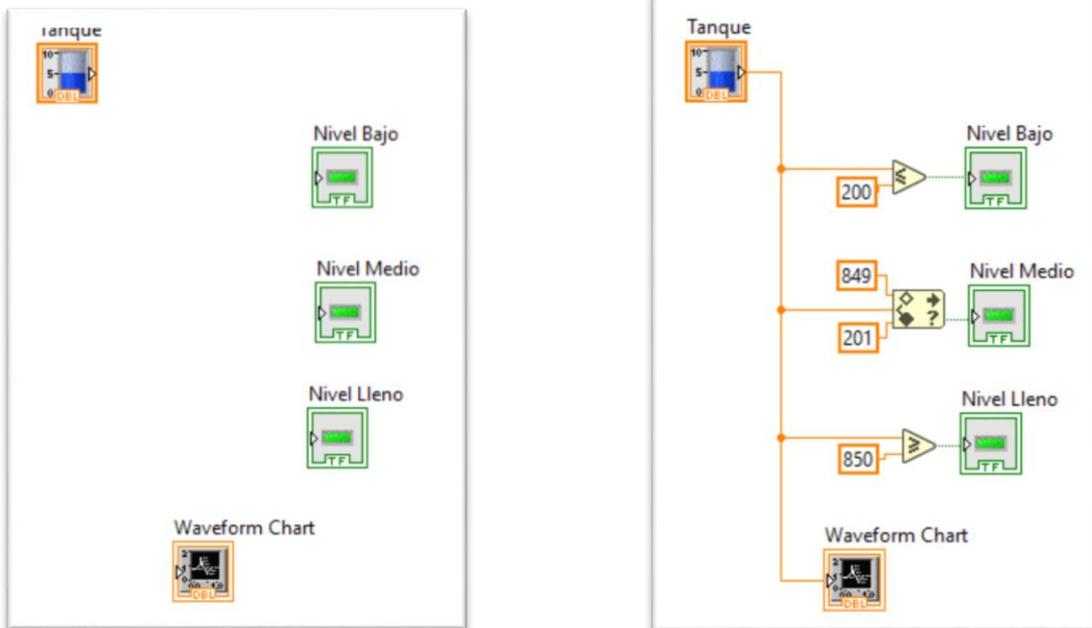
Volvemos a repetir los pasos anteriores pero con la diferencia que en vez de escoger “Numeric” – “Tank”, elegiremos “Graph” - “Waveform Chart”.



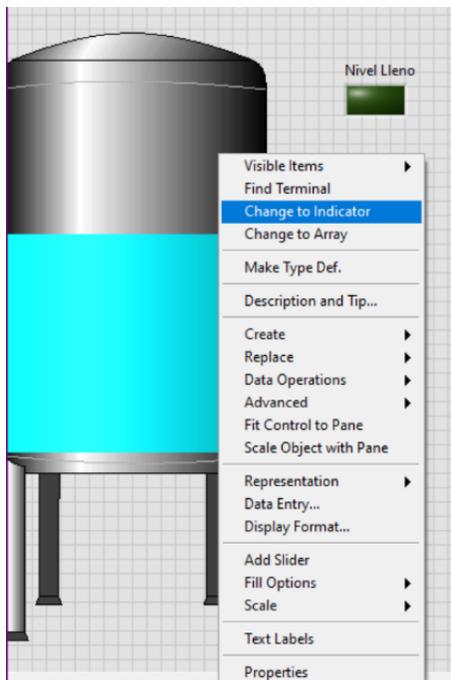
Finalmente ordenamos a nuestro gusto y obtenemos algo como lo siguiente:



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.

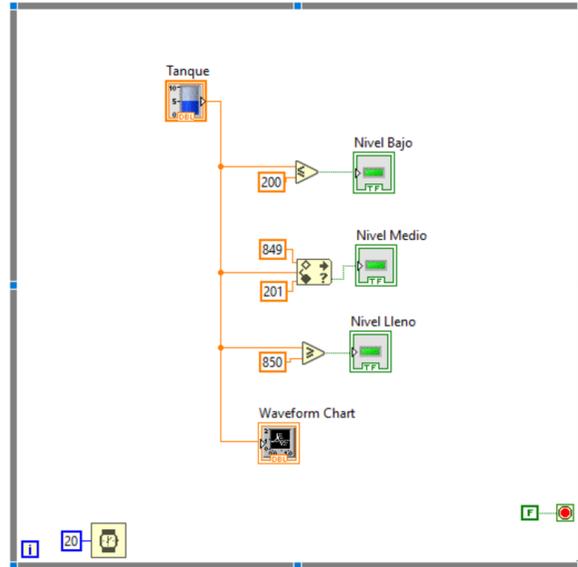
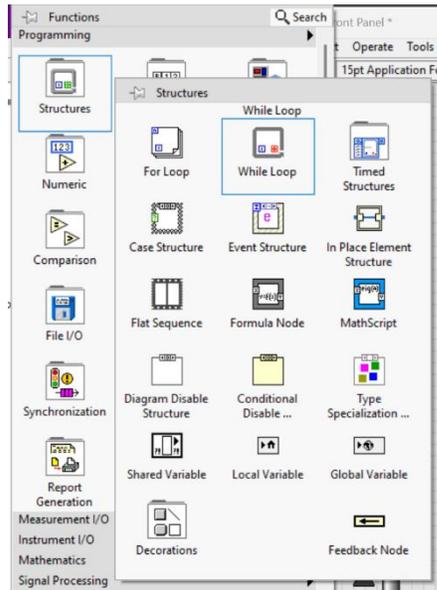


Como se puede observar, en la imagen de lado derecho solo se agregan unos comparadores para poder delimitar los rangos que corresponderán a los límites de los niveles especificados.

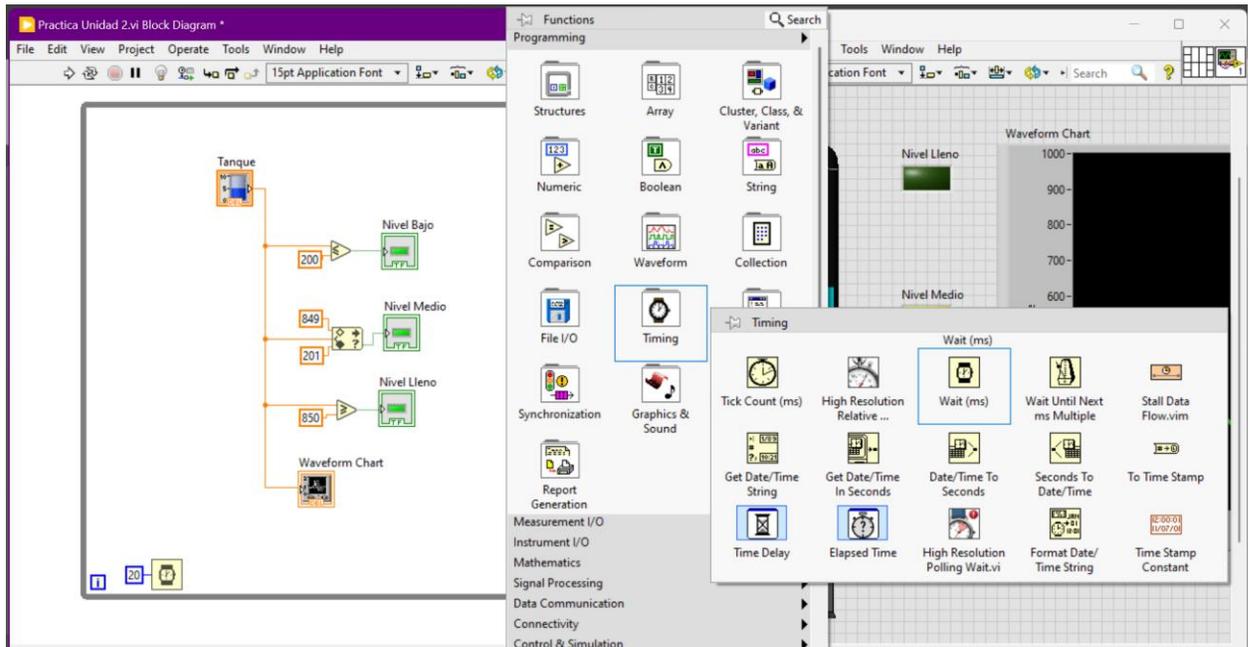


Aquí se realiza una configuración ya que el tanque se nos muestra como indicador, y nosotros lo que queremos es que sea un controlador. Para cambiar esto, debemos dar click derecho sobre el tanque y seleccionar “Change to Indicator”

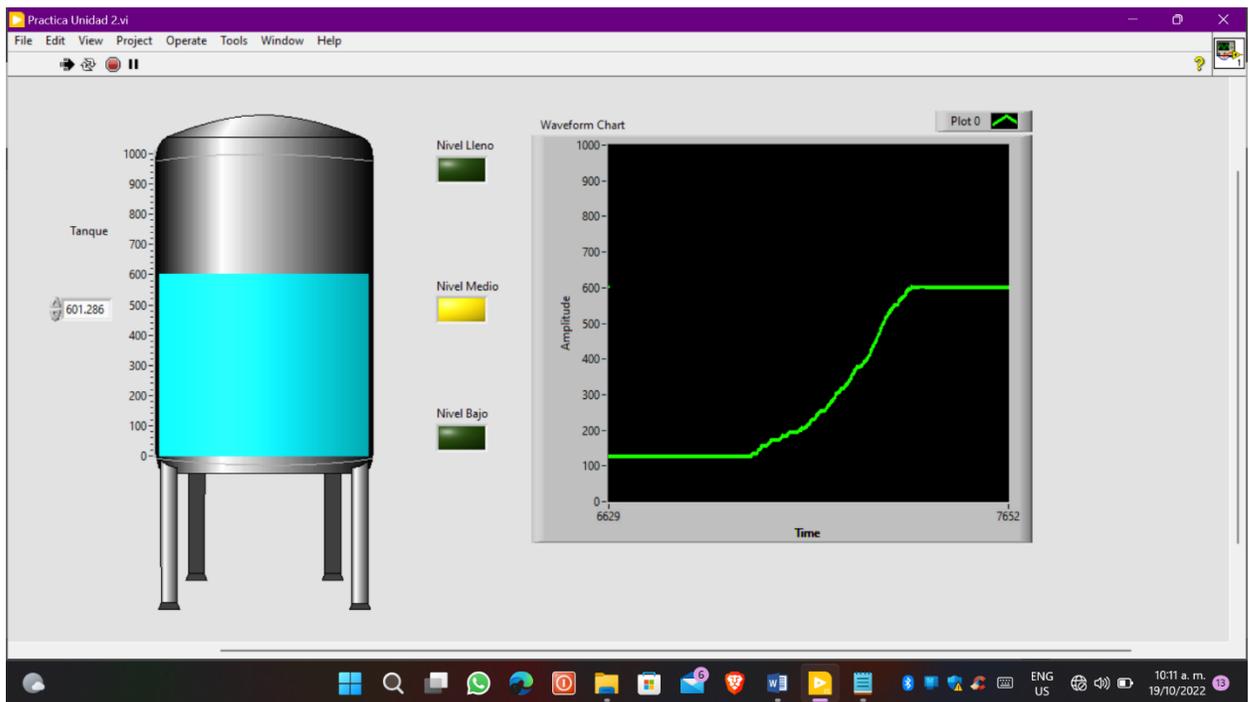
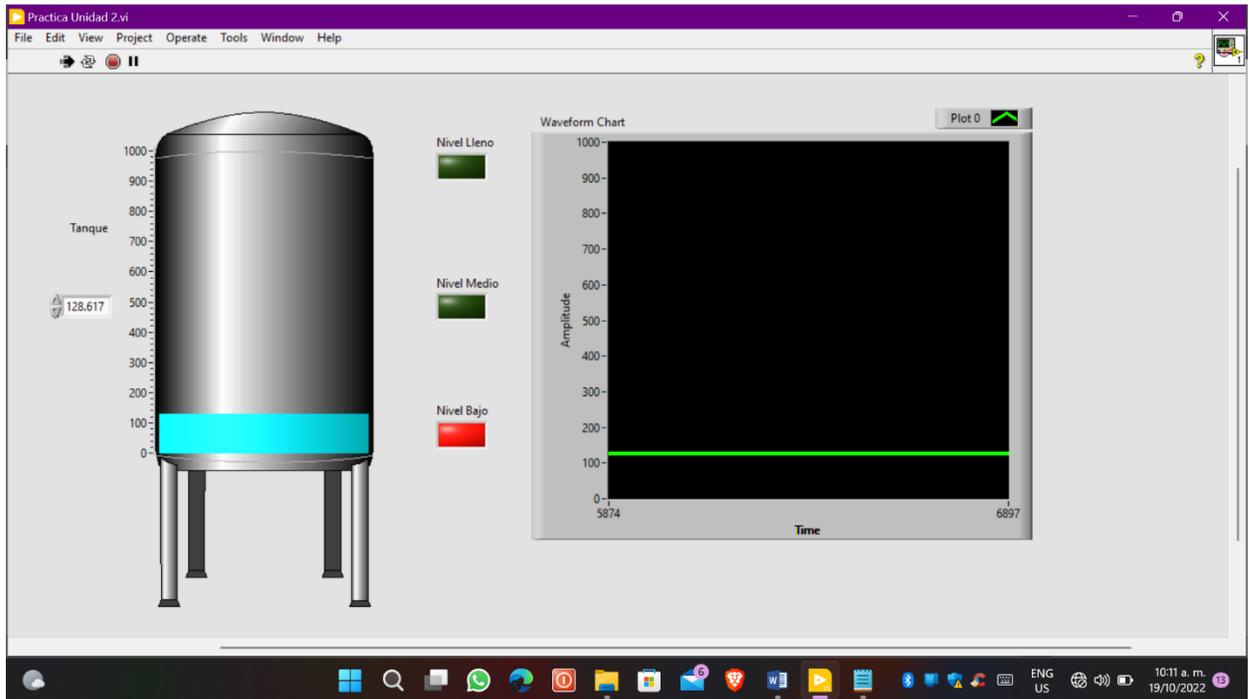
Finalmente encerramos este conjunto de diagramas dentro de una “Structure” “while Loop”

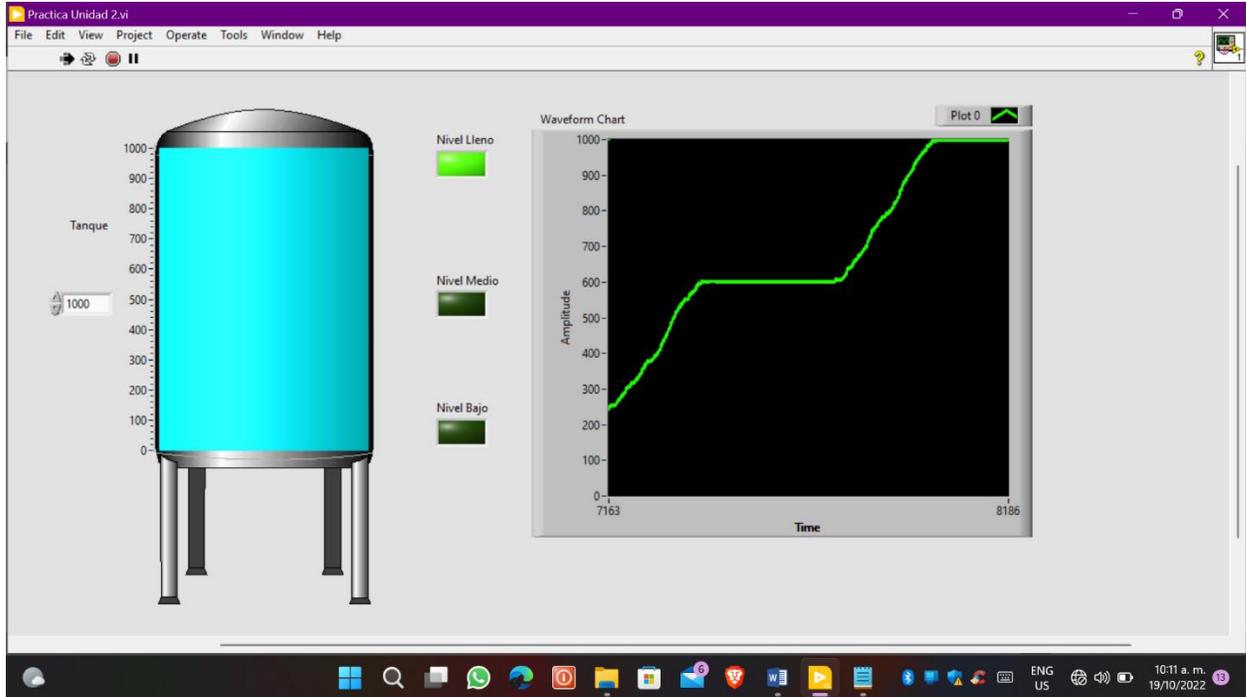


En este último paso, agregamos un “Wait (ms)” en el apartado “Timing”.



## ★ SIMULACION





## ❖ CONCLUSIÓN

Se puede decir que hoy en día la tecnología es una herramienta sumamente importante en las actividades que el ser humano realiza día a día.

Esto debido a que antes de que esta apareciera las formas de trabajar eran un poco más pesadas y difíciles y una forma clara eran las formas de mediciones que estas utilizaban como por ejemplo son: los codos, las brazadas, la vara, el plomo, etc.

En los últimos años han aparecido diferentes softwares que nos han facilitado aprender y así llevar a cabo actividades que parecían difíciles e imposibles, unos de estos ha sido LabVIEW el cual es un software que nos proporciona un ambiente gráfico sumamente importante y en el cual podemos realizar distintos tipos de actividades y en cual aprendimos a diseñar y a medir lo que fue un tanque de agua y así mismo interactuar con el nivel del agua.



**EDUCACIÓN**  
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO  
NACIONAL DE MÉXICO



INSTITUTO TECNOLÓGICO SUPERIOR DE  
SAN ANDRÉS TUXTLA



VERACRUZ  
GOBIERNO  
DEL ESTADO



SEV  
Secretaría  
de Educación

SEMSyS  
Subsecretaría de Educación  
Media Superior y Superior



DET  
Departamento de Educación  
Tecnológica del Estado

## ★ BIBLIOGRAFÍA

<https://youtu.be/ejC3pLn8V68>

INSTITUTO TECNOLÓGICO SUPERIOR DE SAN ANDRÉS TUXTLA.

ALUMNOS;

- ✓ ACOSTA GUILLEN ANGEL DANIEL
- ✓ MARTINEZ HERNANDEZ JUAN ALBERTO
- ✓ PELAXO XOLO LUIS MIGUEL
- ✓ POLANCO POLITO ESTEBAN BERNABE}
- ✓ ROQUE CONDE JONATHAN

DOCENTE:

- ✓ JOSE ANGEL NIEVES VASQUEZ
- ✓ INSTRUMENTACION VIRTUAL

✓ PERIODO AGOSTO-DICIEMBRE 2022

UNIDAD II

711-A

INGENIERIA MECATRONICA.

TEMA: 2.1 AMBIENTES DE PROGRAMACION

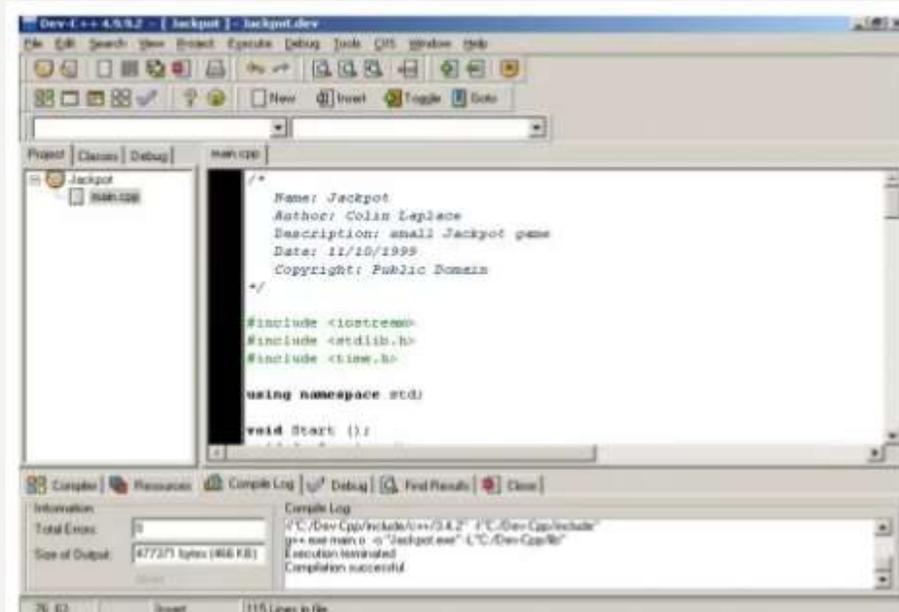
SAN ANRES TUXTLA VER.

21/09/2022



# ENTORNO DE DESARROLLO INTEGRADO

- Un entorno de desarrollo integrado, llamado también IDE, es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien puede utilizarse para varios.
- Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación; es decir, que consiste en un editor de código, un compilador, un depurador, y un constructor de interfaz gráfica (GUI).
- Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes.

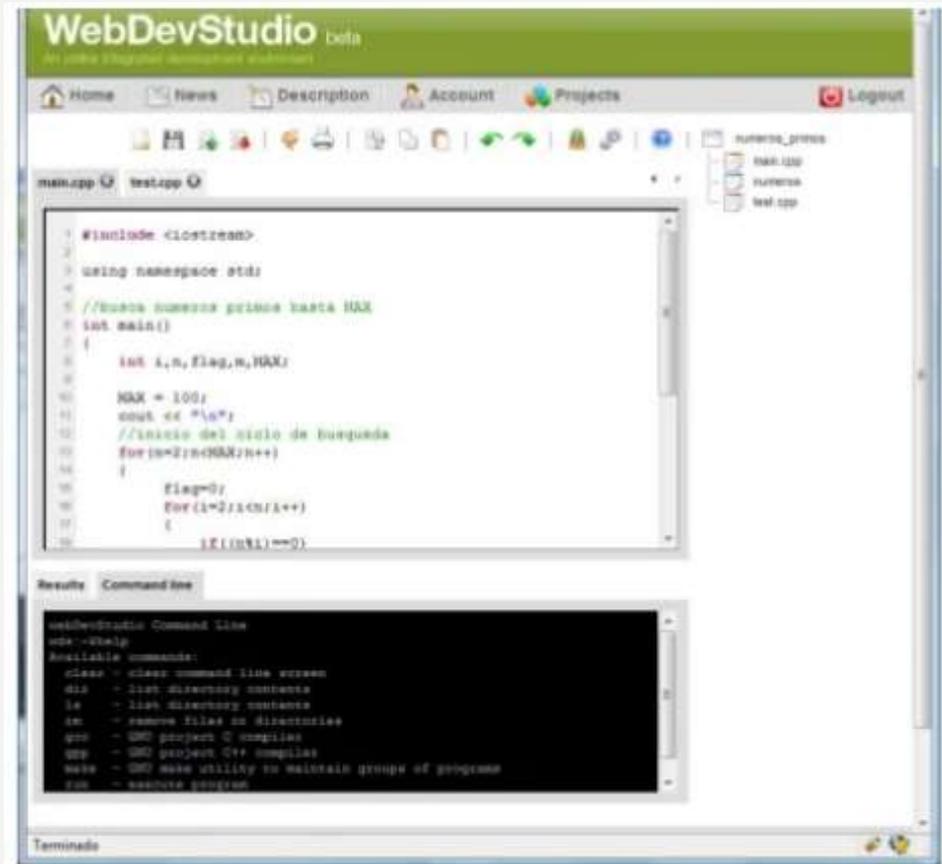


# ENTORNO DE DESARROLLO INTEGRADO

- Los IDE proveen un marco de trabajo amigable para la mayoría de lenguajes de programación tales como c++, PHP, phyton, Java, C#, Delphi, visual basic, gambas, etc.
- En algunos lenguajes, un IDE puede funcionar como un sistema en tiempo de ejecución, en donde se permite utilizar el lenguaje de programación en forma interactiva, sin necesidad de trabajo orientado a archivos de texto, como es el caso de Smaltalk u Objective-C.

# COMPONENTES.

- Un editor de texto.
- Un compilador.
- Un interprete
- Un depurador
- Un cliente
- Posibilidad de ofrecer un sistema de control de versiones
- Factibilidad para la ayuda en la construcción de interfaces graficas de usuario.



WebDevStudio, un IDE en línea para el lenguaje de programación C/C++.

# LENGUAJES.

- Algunos entornos son compatibles con múltiples lenguajes de programación, como visual studio (soporta mas de 50 lenguajes diferentes) Eclipse o NetBeans, ambos basados en java; o MoonDevelop , basado en C#. También puede incorporarse la funcionalidad para lenguajes alternativos mediante el uso de plugins. Por ejemplo, Eclipse y NetBeans tienen plugins para C, C++,Ada, perl, python, Ruby y PHP, entre otros, o visual studio que soporta no solo múltiples lenguajes si no también múltiples dispositivos como móviles.

# ENLAZADORES.

- Por simplificación y para facilitar la comprensión de los conceptos anteriores se ha señalado que los compiladores y los ensambladores generan un código maquina que pueden ser ejecutados por el ordenador, lo habitual es que durante la estructura de un programa sea necesario utilizar los subprogramas, en forma de bibliotecas de funciones o bien que el propio programa este formado realmente por varios programas en diferentes ficheros.

# DEPURADORES.

- Una vez editado y compilado el programa es necesario ejecutarlo, pero es habitual que durante el desarrollo de una aplicación generen ficheros ejecutables, estos depuradores son capaces de ejecutar el programa paso a paso incluyendo además un conjunto de facilidades que permiten observar el valor de las variables y estructuras de datos, permitiendo así una mejor localización de errores no evidentes.



# GUÍA DE OBSERVACIÓN PARA EXPOSICIONES

## U2 Instrumentos Virtuales.

Equipo: \_1\_ Alumno: Pelayo Xolo Luis Miguel

Dominio del tema	15.0%	14%
Orden y claridad	5.0%	4%
Dicción	2.0%	2%
Material utilizado	3.0%	3%
Entrega en tiempo y forma	5.0%	5%
Total	30%	28%

# LISTA DE COTEJO INVESTIGACION

## U1 Instrumentos Virtuales.

Nombre del estudiante: Pelayo Xolo Luis Miguel.

Tema: Los correspondientes a la unidad 2.

Portada	2%	2%
Introducción	5%	5%
Desarrollo	15%	15%
Conclusiones	5%	5%
Referencias	1%	1%
Entrega en tiempo y forma	2%	2%
Total	30%	30%



## LISTA DE COTEJO DE PRÁCTICAS

### U2 Instrumentos Virtuales.

Nombre del estudiante: Pelayo Xolo Luis Miguel.

Tema: Realizar un programa de ambiente grafico (GUI).

Portada	2 %	2%
Objetivo	2%	0%
Introducción (Antecedentes)	5 %	5%
Desarrollo (Materiales, Diagramas, Observaciones)	10 %	10%
Conclusiones	4 %	4%
Referencias	2 %	1%
Simulación o Video	10 %	10%
Entrega en tiempo y forma	5 %	5%
Total	40%	37%

# **INSTITUTO TECNOLÓGICO SUPERIOR DE SAN ANDRÉS TUXTLA**

**CARRERA  
INGENIERIA MECATRONICA**

**ASIGNATURA  
INSTRUMENTACION VIRTUAL**

**DOCENTE  
JOSE ANGEL NIEVES VAQUES**

**UNIDAD: 1**

**GRUPO: 711-A**

**FECHA: 24/10/2022**

**INTEGRANTES**

**✓ PELAYO XOLO LUIS MIGUEL-191U0461**



## Puntualidad

El estudiante debe estar puntual en clases

## Asistencia

3 faltas no tiene derecho a revisión de firmas

## Justificantes

Los justificantes de inasistencias serán por enfermedad y solo serán válidos 3 días después de la ausencia.

## Disciplina

Debe tener buena disciplina en clase, de lo contrario perderá su asistencia.

## Uso de objetos ajenos

Debe evitarse dentro del salón de clase el uso de:

> Dispositivos electrónicos

> Pearsen

> Gorras

> Lentes oscuros

> El uso de laptop queda a uso del docente

## Alimentos

Queda prohibido introducir alimentos dentro de clases.

## Criterios de evaluación

Investigación 30%

Práctica 30%

Exposición 40%

## Ultima unidad

Investigación 40%

Proyecto 60%

DU

10

# Temario

## Unidad 1 Introducción a la instrumentación virtual

- 1.1 Evolución de la instrumentación
- 1.2 Adquisición de datos
- 1.3 Lenguaje de programación virtual
- 1.4 Buses de comunicación
- 1.5 Aplicaciones reales de la instrumentación

20

## Unidad 2 Instrumentos virtuales

- 2.1 Ambientes de programación
- 2.2 Funciones y subrutinas
- 2.3 Ciclos y temporización
- 2.4 Arreglos y grupos de datos
- 2.5 Cadenas de archivos de entrada - salida

## Unidad 3 Elementos de adquisición de datos

- 3.1 Plataforma de software (~~Labview~~, Matlab, VEE)
- 3.2 Diseño de interfaz hombre máquina (HMI)

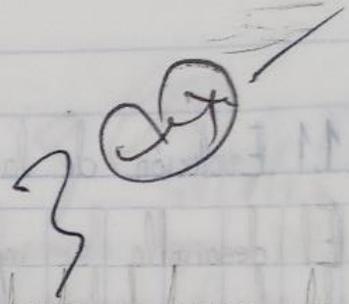
## Unidad 4 Adquisición de datos

- 4.1 Acondicionadores de señal utilizando amplificadores operacionales
- 4.2 Filtros de señal
- 4.3 Características de la conversión analógico - digital
- 4.4 Adquisición de datos analógicos
- 4.5 Adquisición de datos digitales.

Luis Miguel Pelayo Xold

711-A

06/09/2022



1. ¿Qué entiendes por instrumentación virtual?

Emprego de herramientas para un determinado proceso mediante entornos de desarrollo

2. ¿Conoces alguna aplicación de la instrumentación virtual?

Simulaciones virtuales

3. Menciona un ejemplo de instrumentación virtual

Medición de nivel

4. ¿Cómo se relaciona esta materia con tu carrera?

Comprensión de algunos procesos industriales

## 1.1 Evolución de la instrumentación

7/9/2022

El desarrollo se inició con los manómetros, termómetros y válvulas manuales localmente montados. En esta fase eran necesarios muchos operadores para observar los instrumentos y maniobrar las válvulas. Los procesos y los instrumentos eran proyectados empíricamente basándose en la intuición y en la experiencia acumulada y no estaban centralizados para conseguir una mayor eficiencia en las funciones del operador.

La siguiente etapa fue la centralización de las funciones de medida y de control más importantes, pertenecientes a una operación del proceso, en un panel localmente montado. De este modo podía observarse y controlarse el funcionamiento de cada elemento particular de la instalación de una manera semiautomática coordinada y eficaz. Para hacer esto posible, se desarrollaron instrumentos galvanométricos operados por termopar, termómetros con largos capilares y caudalímetros con largos tubos de conducción de la presión diferencial.

El desarrollo de los transmisores neumáticos permitió la centralización de las funciones de medida y de regulación de toda una unidad del proceso en una sala de control, utilizándose como receptores los instrumentos registradores controladores neumáticos de caja grande que aparecieron hacia el año 1940.

A principios de los años 50 aparecen los primeros instrumentos electrónicos a válvulas.

Más tarde se perfecciona la unidad automático-manual neumática, consiguiéndose el cambio en un solo paso, sin que se produzcan saltos en la señal de salida a la válvula y aparecen paralelamente los instrumentos electrónicos miniatura alrededor de los años 1960.

## 1.2 Adquisición de datos

9  
12/7/22

Quando hablamos de adquisición de datos, comunmente abreviado como DAQ o DAS, nos referimos al proceso de realizar mediciones de fenómenos físicos y registrarlos de alguna manera para analizarlos.

En general, se acepta que la adquisición de datos es distinta de las formas anteriores de grabación en cinta o gráficos de papel.

A diferencia de esos métodos, las señales se convierten del dominio analógico al dominio digital y luego se graban en un medio digital como ROM, medios flash o unidades de disco duro.

Los sistemas modernos de adquisición de datos digitales constan de cuatro componentes esenciales que forman toda la cadena de medición de los fenómenos físicos:

- Sensores
- Acondicionamiento de señal
- Convertidores analógico-digital
- Computadora con software DAQ

El sistema típico de adquisición de datos (DAQ) tiene múltiples canales de circuitos de acondicionamiento de señales que proporcionan la interfaz entre los sensores externos y el subsistema de conversión A/D.

Práctica U1 Aplicaciones reales de instrumentación virtual

Período  
Introducción  
Antecedentes  
Desarrollo  
Conclusión  
Presencia

Estructura

Como se medirá alguna variable en la interfaz

26 septiembre

## 1.3 Lenguaje de programación virtual

Son lenguajes más portables que los lenguajes compilados puesto que el código que se genera tras la compilación es un código intermedio o bytecode. Este código puede ser a su vez interpretado por una máquina virtual instalada en cualquier equipo. Tienen una ejecución lenta pero su versabilidad de poder ejecutarse en cualquier entorno los hace muy apreciados.

La programación visual cumple igualmente los paradigmas clásicos de los lenguajes de programación textuales tales como:

### » Programación imperativa.

Cada programa consta de una serie de bloques cada uno de los cuales realiza una operación simple y cuya ejecución se sucede siguiendo un determinado orden.

### » Programación orientado a objetos

Las aplicaciones se componen de objetos cada uno de los cuales constan de sus propios programas y variables que determinan su comportamiento.

### » Programación orientada a eventos

Los programas pueden activarse ante ciertos sucesos que inicien su ejecución (pulsaciones de teclado, ratón, detección de sensores en dispositivos externos, etc.).

La principal ventaja de la programación visual dada su naturaleza es que no requiere aprender y trazar la sintaxis típica de un lenguaje de programación textual resultando más sencilla e intuitiva.

1.4

## Buses de comunicación

Un bus es un medio compartido de comunicación, constituido por un conjunto de líneas (conductores) que conecta las diferentes unidades de un computador. La principal función de un bus será, pues, servir de soporte para la realización de transferencias de información entre dichas unidades. La unidad que inicia y controla la transferencia se conoce como master del bus para dicha transferencia, y la unidad sobre la que se realiza la transferencia se conoce como slave. Los papeles de master y slave son dinámicos, de manera que una misma unidad puede realizar ambas funciones en transferencias diferentes. Por ejemplo, una unidad de DMA hace de slave en la inicialización que realiza el master, la CPU, para una operación de E/S. Sin embargo, cuando comienza la operación, la unidad DMA juega el papel de master frente a la memoria, que en esta ocasión hace de slave.

Para garantizar el acceso ordenado al bus, existe un sistema de arbitraje, centralizado o distribuido, que establece las prioridades cuando dos o más unidades pretenden acceder al mismo tiempo al bus, es decir, garantiza que en cada momento solo exista un master.

Para establecer el tiempo de duración de las transferencias y que sea conocido tanto por el master como por el slave, un bus debe disponer de los medios necesarios para la sincronización master-slave.

Las líneas de un bus podemos clasificarlos en grupos, atendiendo al papel que cumplen en las transferencias.

Aunque existen diseños muy diversos, las líneas que forman el bus del sistema pueden clasificarse en tres grupos de funciones: datos, dirección y control.

También pueden existir líneas de alimentación para suministrar energía a los módulos conectados al bus.

## 1.5 Aplicaciones reales de la instrumentación

Los sensores, transductores, equipos de adquisición de datos o sistemas de telemetría, son empleados habitualmente en todo tipo de aplicaciones de instrumentación electrónica.

Existe una gran variedad de sectores en los que encontramos aplicación de instrumentación electrónica. Automatización industrial, obra civil, automóvil, ferrocarril, energía etc., son sectores donde podemos encontrar diferentes aplicaciones para sensores, transductores, electrónicos o equipos de adquisición de datos.

Alguna de estas aplicaciones pueden ser, procesos de automatización, ensayos estáticos y dinámicos en vehículo o ferrocarril, energía, etc. Son sectores donde podemos encontrar diferentes aplicaciones para sensores, transductores, electrónicos o equipo de adquisición de datos.

A continuación se muestra un resumen de alguna de las aplicaciones y sectores en los que SENSING está presente en la instrumentación electrónica.

- Análisis de señales dinámicas DSA
- Instrumentación para el sector energético
- Instrumentación en la logística y el transporte
- Instrumentación para el sector industrial
- Instrumentación para ensayos en el automóvil
- Instrumentación para ensayos en Ingeniería Civil
- Instrumentación para ensayos en tren y vía
- Instrumentación para seguridad laboral.

17/17/22

# **INSTITUTO TECNOLÓGICO SUPERIOR DE SAN ANDRÉS TUXTLA**

**CARRERA  
INGENIERIA MECATRONICA**

**ASIGNATURA  
INSTRUMENTACION VIRTUAL**

**DOCENTE  
JOSE ANGEL NIEVES VAQUES**

**UNIDAD: 2**

**GRUPO: 711-A**

**FECHA: 24/10/2022**

**INTEGRANTES**

**✓ PELAYO XOLO LUIS MIGUEL-191U0461**



## 2.1 Ambientes de programación

29  
22/9/12

Un entorno de desarrollo integrado, llamado también IDE (Sigla en inglés de Integrated Development Environment), es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien puede utilizarse para varios.

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación; es decir, que consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Los IDEs pueden ser aplicaciones por sí solos o pueden ser parte de aplicaciones existentes.

Los IDE proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación tales como C++, PHP, Python, Java, C#, Delphi, Visual Basic, Gambas, etc. En algunos lenguajes, un IDE puede funcionar como un sistema en tiempo de ejecución en donde se permite utilizar el lenguaje de programación en forma interactiva, sin necesidad de trabajo orientado a archivos de texto, como es el caso de Smalltalk y Objective-C.

### Componentes

- » Un editor de texto
- » Un compilador
- » Un intérprete
- » Un cliente
- » Posibilidad de ofrecer un sistema de control de versiones
- » Facilidad para ayuda en la construcción de interfaces gráficas de usuario.

Investigar los siguientes lenguajes de programación  
C, C++, Ada, Perl, Python, Ruby y PHP

29

## 2.2 Funciones y Subrutinas

21/7/12

- El diseño descendente permite obtener un programa que resuelva un problema dividiendo este en subproblemas cada vez más sencillos.
- Cada subproblema tiene asociado un pseudocódigo de alto nivel compuesto por acciones no primitivas.
- Cuando una de estas acciones no primitivas se repite en varios puntos del algoritmo es interesante darle un nombre y reutilizarlo.
- Estas acciones con nombre se denominan subprogramas, pudiendo ser, a su vez funciones y subrutinas.

### Ventajas de los subprogramas:

- \* Facilitan la modularidad y estructuración de los algoritmos
- \* Facilitan la lectura e inteligibilidad de los algoritmos.
- \* Permiten economizar el esfuerzo del programador al permitir reutilizar el mismo código en varios puntos del mismo algoritmo.
- \* Facilitan la depuración y mantenimiento de los programas

### Funciones

- Las funciones son subprogramas con 0 o más argumentos que siempre devuelven un valor de retorno.
- Las funciones pertenecen entonces a un tipo determinado
- La invocación de una función puede formar parte de:
  - » Una expresión
  - » La parte derecha de una asignación
- Una invocación de una función no puede formar:
  - » Una sentencia aislada
  - » La parte izquierda de una asignación

# Lenguajes de programación

## C/C++

Desarrollado entre 1969 y 1973 por Dennis Ritchie en los laboratorios Bell y diseñado inicialmente para ser usado en un entorno de programación bajo UNIX, su eficiencia y flexibilidad le permitieron rápidamente su expansión a otros sistemas operativos y campos de aplicación, alcanzando su estandarización en 1989 bajo el nombre de ANSI C.

Para desarrollar prácticamente cualquier software necesitamos un entorno de desarrollo, es decir, un programa en el que escribir el código que queremos producir. Uno de los más conocidos y extendidos es Visual Studio Code, con su extensión de C++, que incluye compilador y debug.

## Aplicaciones en Sistemas Embebidos:

### » Electrodomésticos

Posiblemente nuestra lavadora posea un sistema con software embebido para controlar el programa de lavado, la pantalla donde se muestra información al usuario, los botones que permiten que el usuario interactúe con el aparato y por ejemplo la apertura/cerradura de válvulas que permiten el paso de agua.

Normalmente hay sistemas embebidos también en nuestros microondas para controlar la pantalla, temperaturas y tiempos, o incluso en las básculas para que manden sus resultados a nuestros teléfonos.

### » Domótica o casa inteligente

### » Medios de transporte

### » Asistentes virtuales

Uno de los asistentes virtuales más conocidos es el Asistente de Google, que está programado en C++, pudiendo encontrarse en móviles, wearables y domótica.

## Ada

Es un lenguaje de programación orientado a objetos y fuertemente tipado de forma estática que fue diseñado por Jean Ichbiah de CII Honeywell Bull por encargo del Departamento de Defensa de los Estados Unidos. Es un lenguaje multipropósito, orientado a objetos y concurrente, pudiendo llegar desde la factibilidad de Pascal hasta la flexibilidad de C++.

Ada usa principalmente en entornos en los que se necesita una gran seguridad y fiabilidad, como la defensa, la aeronáutica (Boeing y Airbus), la gestión de tráfico aéreo (como Indra en España) y la industria aeroespacial entre otros.

## Perl

Practical Extraction and Report Language, fue creado a principio de los noventa por Larry Wall.

Es un lenguaje pensado para la manipulación de cadena de caracteres, archivos y procesos. No es compilado anteriormente (precompilado) pero aún así es más rápido que la mayoría de lenguajes interpretados, en especial que el Bourne Shell. Esto se debe a que los programas en Perl son analizados, interpretados y compilados por el intérprete perl antes de su ejecución.

Por todo esto, Perl es un lenguaje muy utilizado en los dos campos siguientes:

### 1. Administración de SO

Es muy potente en la creación de pequeños programas que pueden ser usados como filtros para obtener información de ficheros, realizar búsquedas, etc

### 2. Creación de formularios en la web

Se encarga de tratar y hacer llegar la información que el cliente WWW manda al servidor a través de un formulario.

## Python

Es un lenguaje de programación de alto nivel, orientado a objetos, con una semántica dinámica integrada, principalmente para el desarrollo web y de aplicaciones informáticas. Su filosofía hace hincapié en la legibilidad de su código, se utiliza para desarrollar aplicaciones de todo tipo, ejemplos: Instagram, Netflix, Spotify, Panda 3D, entre otros. Se trata de un lenguaje de programación multiparadigma, ya que soporta parcialmente la orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, dinámico y multiplataforma.

uby  
Es un lenguaje de programación, que principalmente está orientado a objetos pero también puede ser programación funcional, por ejemplo. Este lenguaje de programación Open-source se inspiró en lenguajes como Perl, Eiffel y Lisp.

Estar orientado a objetos significa que funciona con datos que el usuario creará en función de sus propias necesidades. Los objetos son formas de presentar objetos de la vida real dentro del software.

Se usa principalmente para el desarrollo de aplicaciones web. Muchas aplicaciones famosas utilizan el lenguaje Ruby, como los sitios web de Airbnb y SoundCloud, la red social Twitch y el sitio web de streaming de series y películas Hulu.

PHP

Es un lenguaje de programación destinado a desarrollar aplicaciones para la web y crear páginas web, favoreciendo la conexión entre los servidores y la interfaz de usuario.

Entre los factores que hicieron que PHP se volviera tan popular, se destaca el hecho de que es de código abierto.

Esto significa que cualquiera puede hacer cambios en su estructura. En la práctica, esto representa dos cosas importantes:

### 1. Código abierto

No hay restricciones de uso vinculadas a los derechos. El usuario puede usar PHP para programar en cualquier proyecto y comercializarlo sin problemas.

### 2. Constante perfeccionamiento

Gracias a una comunidad de desarrolladores proactiva y comprometida.

Una de las características principales de PHP es que es un lenguaje mucho más dinámico que la mayoría de las otras opciones que existen.

Por lo tanto, es esencial para desarrollar sitios que tienen aplicaciones más complejas y, para eso, necesitamos dos cosas: agilidad en el tiempo de respuesta y conexión a una gran base de datos. Por ejemplo Facebook

## 2.3 Ciclos y temporización.

La temporización es un elemento esencial para todas las aplicaciones de pruebas, control y diseño que debe ser consideradas para clave en cualquier sistema.

Las tecnologías de temporización y sincronización correlacionan eventos en el tiempo, lo cual es necesario para realizar actividades coordinadas. Para que el software pueda organizar estas actividades coordinadas, el programa debe estar sincronizado y tener un concepto de tiempo.

LabVIEW usa un componente de software llamado motor de nano segundos funciona en segundo plano e interactúa con el sistema operativo para administrar el tiempo. Hay una variedad de funciones y estructuras en Labview que usa el motor de nanosegundos para mantener el tiempo, como la función wait y la estructura de ciclo temporizado el motor de nanosegundos puede usar un reloj local en tiempo real (RTC) o puede ser impulsado por un reloj de referencia externo e integrado a través de N.I-Time Sync.

SA  
28/12/22

## 2.4 Arreglos y grupo de datos.

Un arreglo, el cual consta de elementos y dimensiones, es un control o un indicador; no puede contener una combinación de controles e indicadores. Los elementos son los datos o valores contenidos en el arreglo. Una dimensión es la longitud, altura o profundidad de un arreglo.

Los arreglos son muy útiles cuando trabaja con una colección de datos similares y cuando desea almacenar un historial de cálculos repetitivos.

Los elementos del arreglo son ordenados. Cada elemento en un arreglo tiene un valor de índice correspondiente, y puede usar el índice del arreglo para acceder a un elemento específico en esa matriz. En el software NI LabVIEW, el índice del arreglo está basado en cero. Esto significa que si un arreglo unidimensional (1D) contiene  $n$  elementos, el rango del índice es de 0 a  $n-1$ , donde el índice  $n-1$  apunta al último elemento en el arreglo.

Los clústeres agrupan elementos de datos de diferentes tipos. Un ejemplo de un cluster es similar a un registro o a una estructura en lenguajes de programación basados en texto.

Similar a los arreglos, un clúster es un control o un indicador y no puede contener una combinación de controles e indicadores. La diferencia entre los clústeres y los arreglos es que un clúster particular tiene un tamaño fijo y un arreglo particular puede variar en tamaño. Además, un clúster puede contener tipos de datos mixtos, pero un arreglo puede contener solo un tipo de datos.

60  
12/1/12

## 2.5 Cadena de archivos de entrada/salida

Son innumerables los casos en los que son necesarios guardar datos entre ejecuciones de un programa para poder ser recuperados en futuras sesiones. Los archivos de datos permiten almacenar información de cualquier tipo de modo permanente para ser accedida o alterada cuando sea necesario.

Cuando se trabaja con archivos secuenciales de datos, el primer paso es establecer un área de buffer, donde la información se almacena temporalmente mientras se está transfiriendo entre la memoria y el archivo de datos. Esta área de buffer permite leer y escribir información del archivo más rápidamente de lo que sería posible de otra manera. El área de buffer permite leer y escribir información y establece escribiendo `FILE *ptvar;` Donde `FILE` es un tipo de estructura que establece el área de buffer y `ptvar` la variable puntero (puntero a archivo) que indica el comienzo de esta área. El tipo de estructura `FILE` está definido en `stdio.h`. Aquí, a `ptvar` se le conoce como un flujo lo cual es un dispositivo lógico resultado de la transformación de un archivo con buffer.

Un archivo secuencial de datos puede crearse de dos formas distintas. Una es crear el archivo directamente usando un editor. La otra es escribir un programa que introduzca información en la computadora y la escriba en un archivo. Los archivos sin formato solo pueden crearse mediante programas.

Los archivos de datos que contienen solo cadenas de caracteres pueden crearse y leerse más fácilmente con programas que utilizan funciones de biblioteca especialmente orientados para cadenas: `fgets()` y `fputs()`.

7/25/12