



**APLICACIONES ENFOCADAS AL DESARROLLO
DE LAS EMPRESAS DE MANUFACTURA
MEDIANTE LA ROBÓTICA Y EL USO DE
HERRAMIENTAS DE CONTROL.**

a. Robótica

La robótica es la ciencia y la técnica que está involucrada en el diseño, la fabricación y la utilización de robots. Un robot es, por otra parte, una máquina que puede programarse para que interactúe con objetos y lograr que imite, en cierta forma, el comportamiento humano o animal.

La informática, la electrónica, la mecánica y la ingeniería son sólo algunas de las disciplinas que se combinan en la robótica. El objetivo principal de la robótica es la construcción de dispositivos que funcionen de manera automática y que realicen trabajos dificultosos o imposibles para los seres humanos.

Actualmente la robótica ha ido evolucionando a pasos agigantados y ha dado lugar al desarrollo de una serie de disciplinas como sería el caso de la cirugía robótica. En este caso, la misma tiene como claro objetivo el mejorar la salud del ser humano y para ello lleva a cabo una serie de intervenciones quirúrgicas muy complejas que requieren una gran precisión. Así, mediante robots se consigue eliminar los peligros que trae consigo el que sea acometidas por la mano del hombre.

De esta manera, hay que resaltar, por ejemplo, la existencia de un robot llamado Da Vinci que se ha convertido en uno de los pilares de la mencionada cirugía. Se trata de un dispositivo a través del cual se han conseguido llevar a cabo con éxito operaciones tan importantes como las de cirugía transoral.

Asimismo, la robótica ha conseguido también crear robots que sean útiles para asistir y ayudar a todas aquellas personas que se encuentran con algún tipo de discapacidad física. Y eso sin olvidar el conjunto de robots que se están diseñando en el ámbito militar para, por ejemplo, llevar a cabo operaciones de salvamento.

El escritor Isaac Asimov (1920–1992) suele ser considerado como el responsable del concepto de robótica. Este autor, especializado en obras de ciencia ficción y divulgación científica, propuso las Tres Leyes de la Robótica, una especie de normativa que regula el accionar de los robots de sus libros de ficción pero que, de

alcanzarse un grado de desarrollo tecnológico semejante, podrían aplicarse en la realidad futura. Dichas reglas son impresas como fórmulas matemáticas en los “*senderos positrónicos*” de la memoria del robot.

La Primera Ley de la Robótica señala que un robot no debe dañar a una persona o dejar que una persona sufra un daño por su falta de acción. La Segunda Ley afirma que un robot debe cumplir con todas las órdenes que le dicta un humano, con la salvedad que se produce si estas órdenes fueran contradictorias respecto a la Primera Ley. La Tercera Ley establece que un robot debe cuidar su propia integridad, excepto cuando esta protección genera un inconveniente con la Primera o la Segunda Ley.

Un robot industrial es un manipulador multifuncional reprogramable, capaz de mover materias, piezas, herramientas, o dispositivos especiales, según trayectorias variables, programadas para realizar tareas diversas.

Esta definición, ligeramente modificada, ha sido adoptada por la Organización Internacional de Estándares (ISO) que define al robot industrial como:

Manipulador multifuncional reprogramable con varios grados de libertad, capaz de manipular materias, piezas, herramientas o dispositivos especiales según trayectorias variables programadas para realizar tareas diversas.

Se incluye en esta definición la necesidad de que el robot tenga varios grados de libertad. Una definición más completa es la establecida por la Asociación Francesa de Normalización (AFNOR), que define primero el manipulador y, basándose en dicha definición, el robot:

Manipulador: mecanismo formado generalmente por elementos en serie, articulados entre sí, destinado al agarre y desplazamiento de objetos. Es

multifuncional y puede ser gobernado directamente por un operador humano o mediante dispositivo lógico.

Robot: manipulador automático servo-controlado, reprogramable, polivalente, capaz de posicionar y orientar piezas, útiles o dispositivos especiales, siguiendo trayectoria variables reprogramables, para la ejecución de tareas variadas. Normalmente tiene la forma de uno o varios brazos terminados en una muñeca. Su unidad de control incluye un dispositivo de memoria y ocasionalmente de percepción del entorno. Normalmente su uso es el de realizar una tarea de manera cíclica, pudiéndose adaptar a otra sin cambios permanentes en su material.

Por último, la Federación Internacional de Robótica (IFR) distingue entre robot industrial de manipulación y otros robots:

Por robot industrial de manipulación se entiende una máquina de manipulación automática, reprogramable y multifuncional con tres o más ejes que pueden posicionar y orientar materias, piezas, herramientas o dispositivos especiales para la ejecución de trabajos diversos en las diferentes etapas de la producción industrial, ya sea en una posición fija o en movimiento.

En esta definición se debe entender que la reprogramabilidad y la multifunción se consiguen sin modificaciones físicas del robot.

Común en todas las definiciones anteriores es la aceptación del robot industrial como un brazo mecánico con capacidad de manipulación y que incorpora un control más o menos complejo. Un sistema robotizado, en cambio, es un concepto más amplio. Engloba todos aquellos dispositivos que realizan tareas de forma automática en sustitución de un ser humano y que pueden incorporar o no a uno o varios robots, siendo esto último lo más frecuente.

La IFR distingue entre cuatro tipos de robots:

1. Robot secuencial.
2. Robot de trayectoria controlable.
3. Robot adaptativo.
4. Robot tele manipulado.

Clasificación de los robots industriales en generaciones	
1ª Generación	Repite la tarea programada secuencialmente. No toma en cuenta las posibles alteraciones de su entorno.
2ª Generación	Adquiere información limitada de su entorno y actúa en consecuencia. Puede localizar, clasificar (visión) y detectar esfuerzos y adaptar sus movimientos en consecuencia.
3ª Generación	Su programación se realiza mediante el empleo de un lenguaje natural. Posee la capacidad para la planificación automática de sus tareas.

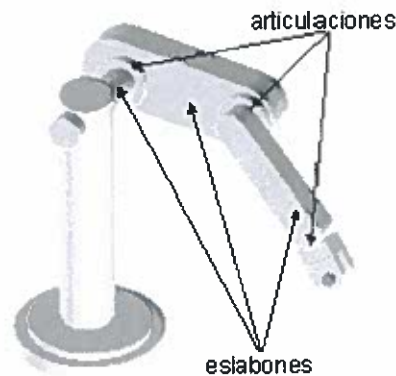
Clasificación de los robots según T. M. Knasel				
Generación	Nombre	Tipo de Control	Grado de movilidad	Usos más frecuentes
1 (1982)	<i>Pick & place</i>	Fines de carrera, aprendizaje	Ninguno	Manipulación, servicio de maquinas
2 (1984)	Servo	Servocontrol, trayectoria continua, progr. condicional	Desplazamiento por vía	Soldadura, pintura
3 (1989)	Ensamblado	Servos de precisión, visión, tacto,	Guiado por vía	Ensamblado, desbardado
4 (2000)	Móvil	Sensores inteligentes	Patas, ruedas	Construcción, mantenimiento

5 (2010)	Especiales	Controlados con técnicas de IA	Andante, saltarín	Militar, espacial
----------	------------	--------------------------------	-------------------	-------------------

Un sistema robotizado, en cambio, es un concepto más amplio. Engloba todos aquellos dispositivos que realizan tareas de forma automática en sustitución de un ser humano y que pueden incorporar o no a uno o varios robots, siendo esto último lo más frecuente.

1. Estructura de los robots industriales

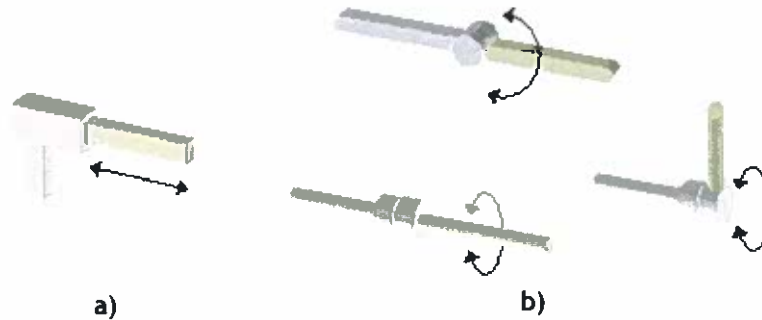
Un manipulador robótico consta de una secuencia de elementos estructurales rígidos, denominados enlaces o eslabones, conectados entre sí mediante juntas o articulaciones, que permiten el movimiento relativo de cada dos eslabones consecutivos.



Elementos estructurales de un robot industrial

Una articulación puede ser:

- Lineal (deslizante, traslacional o prismática), si un eslabón desliza sobre un eje solidario al eslabón anterior.
- Rotacional, en caso de que un eslabón gire en torno a un eje solidario al eslabón anterior.



Distintos tipos de articulaciones de un robot: a) lineal, b) rotacionales

El conjunto de eslabones y articulaciones se denomina cadena cinemática. Se dice que una cadena cinemática es abierta si cada eslabón se conecta mediante articulaciones exclusivamente al anterior y al siguiente, exceptuando el primero, que se suele fijar a un soporte, y el último, cuyo extremo final queda libre. A éste se puede conectar un elemento terminal o actuador final: una herramienta especial que permite al robot de uso general realizar una aplicación particular, que debe diseñarse específicamente para dicha aplicación: una herramienta de sujeción, de soldadura, de pintura, etc. El punto más significativo del elemento terminal se denomina punto terminal (PT). En el caso de una pinza, el punto terminal vendría a ser el centro de sujeción de la misma.



Punto terminal de un manipulador

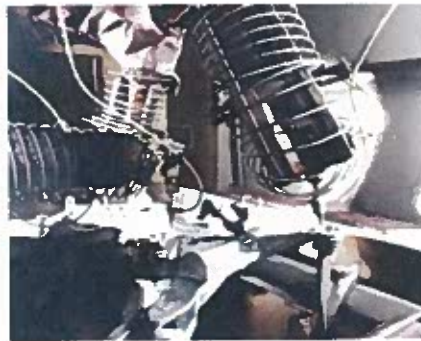
Los elementos terminales pueden dividirse en dos categorías:

- pinzas (*gripper*)
- herramientas

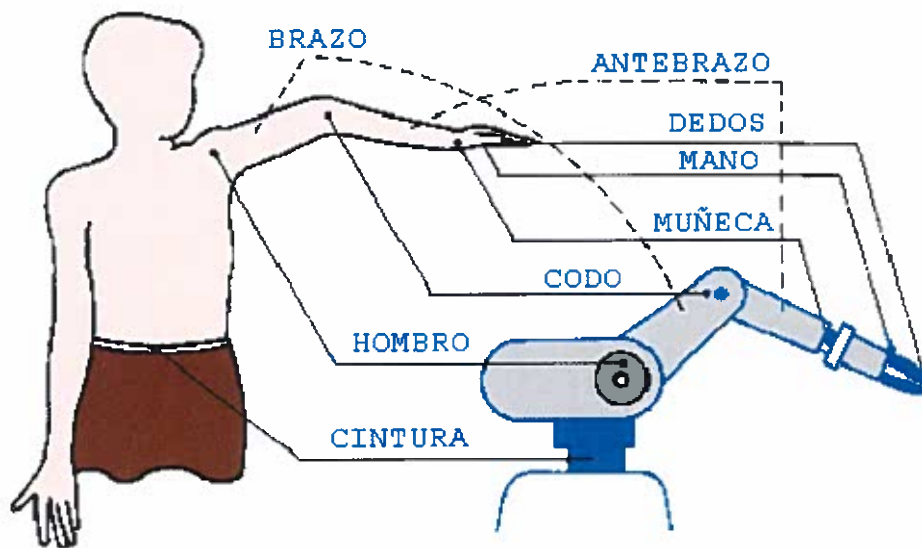
Las pinzas se utilizan para tomar un objeto, normalmente la pieza de trabajo, y sujetarlo durante el ciclo de trabajo del robot. Hay una diversidad de métodos de sujeción que pueden utilizarse, además de los métodos mecánicos obvios de agarre de la pieza entre dos o más dedos. Estos métodos suplementarios incluyen el empleo de casquillos de sujeción, imanes, ganchos, y cucharas.



Una herramienta se utiliza como actuador final en aplicaciones en donde se exija al robot realizar alguna operación sobre la pieza de trabajo. Estas aplicaciones incluyen la soldadura por puntos, la soldadura por arco, la pintura por pulverización y las operaciones de taladro. En cada caso, la herramienta particular está unida a la muñeca del robot para realizar la operación.



A los manipuladores robóticos se les suele denominar también brazos de robot por la analogía que se puede establecer, en muchos casos, con las extremidades superiores del cuerpo humano.



Semejanza de un brazo manipulador con la anatomía humana

Se denomina grado de libertad (g.d.l.) a cada una de las coordenadas independientes que son necesarias para describir el estado del sistema mecánico del robot (posición y orientación en el espacio de sus elementos). Normalmente, en cadenas cinemáticas abiertas, cada par eslabón-articulación tiene un solo grado de libertad, ya sea de rotación o de traslación. Pero una articulación podría tener dos o más g.d.l. que operan sobre ejes que se cortan entre sí.



Distintos grados de libertad de un brazo de robot

Para describir y controlar el estado de un brazo de robot es preciso determinar:

- La posición del punto terminal (o de cualquier otro punto) respecto de un sistema de coordenadas externo y fijo, denominado el sistema mundo.
- El movimiento del brazo cuando los elementos actuadores aplican sus fuerzas y momentos.

El análisis desde el punto de vista mecánico de un robot se puede efectuar atendiendo exclusivamente a sus movimientos (estudio cinemático) o atendiendo además a las fuerzas y momentos que actúan sobre sus partes (estudio dinámico) debidas a los elementos actuadores y a la carga transportada por el elemento terminal.

2. Configuraciones morfológicas y parámetros característicos de los robots industriales

Según la geometría de su estructura mecánica, un manipulador puede ser:

- a) Cartesiano, cuyo posicionamiento en el espacio se lleva a cabo mediante articulaciones lineales.

- b) Cilíndrico, con una articulación rotacional sobre una base y articulaciones lineales para el movimiento en altura y en radio.
- c) Polar, que cuenta con dos articulaciones rotacionales y una lineal.
- d) Esférico (o de brazo articulado), con tres articulaciones rotacionales.
- e) Mixto, que posee varios tipos de articulaciones, combinaciones de las anteriores. Es destacable la configuración SCARA (*Selective Compliance Assembly Robot Arm*)
- f) Paralelo, posee brazos con articulaciones prismáticas o rotacionales concurrentes.

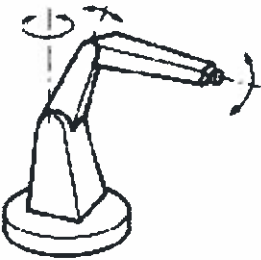
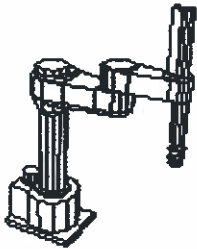
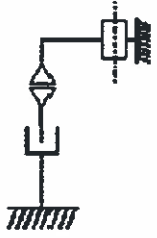
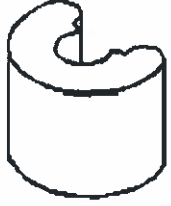

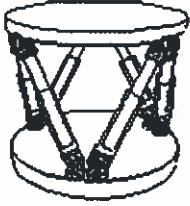
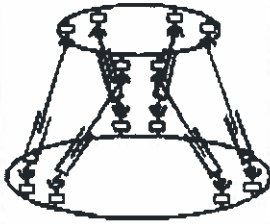


Los principales parámetros que caracterizan a los robots industriales son:

- Número de grados de libertad. Es el número total de grados de libertad de un robot, dado por la suma de g.d.l. de las articulaciones que lo componen. Aunque la mayoría de las aplicaciones industriales requieren 6 g.d.l., como las de soldadura, mecanizado y almacenamiento, otras más complejas requieren un número mayor, tal es el caso de las labores de montaje.
- Espacio de accesibilidad o espacio (volumen) de trabajo. Es el conjunto de puntos del espacio accesibles al punto terminal, que depende de la configuración geométrica del manipulador. Un punto del espacio se dice totalmente accesible si el PT puede situarse en él en todas las orientaciones que permita la constitución del manipulador y se dice parcialmente accesible si es accesible por el PT pero no en todas las orientaciones posibles. En la figura inferior se aprecia el volumen de trabajo de robots de distintas configuraciones.
- Capacidad de posicionamiento del punto terminal. Se concreta en tres magnitudes fundamentales: resolución espacial, precisión y repetibilidad, que miden el grado de exactitud en la realización de los movimientos de un manipulador al realizar una tarea programada.



- Capacidad de carga. Es el peso que puede transportar el elemento terminal del manipulador. Es una de las características que más se tienen en cuenta en la selección de un robot dependiendo de la tarea a la que se destine.
- Velocidad. Es la máxima velocidad que alcanzan el PT y las articulaciones.

Configuración geométrica	Estructura cinemática	Espacio de trabajo	Ejemplo
<p>cartesianos</p>  <p>tipo cantilever</p>  <p>tipo pórtico</p>			
<p>cilíndrico</p> 			
<p>polar</p> 			
<p>esférico</p>			

			
<p>SCARA</p> 			
<p>paralelo</p> 			

Configuraciones geométricas, estructura cinemática, espacio de accesibilidad y ejemplos de robots industriales

El trabajo desarrollado abarcó algunas ideas básicas en cuanto a un robot industrial se refiere, en rasgos generales se trató sobre como reconocer los diferentes tipos de robots existentes en el mercado, cabe recalcar que en la actualidad existen otros muchos modelos o configuraciones sobre estos robots, lo que se deja para otro tema de estudio. Los modelos para poder estudiar al robot incluye un análisis matemático profundo el cual para muchos será muy tedioso e incluso aburrido, pero solamente los verdaderos apasionados por la robótica serán capaces de asumir este reto.

En la actualidad los robots industriales han mejorado el tiempo de trabajo y calidad de algunos productos en las grandes fábricas, aun mas también han ayudado al hombre en algunas tareas que son de alto peligro, dejando el trabajo a un brazo mecánico, pero esto también nos lleva a crearnos otras preguntas en la cabeza, ¿reemplazara la mano robótica a la del hombre?, ¿se convertirá el hombre en un sedentario viviendo de máquinas?, tal vez sean preguntas que cada uno de nosotros nos debemos responder.

Resulta interesante el pensar que cuando éramos niños imaginábamos un robot y jugamos con nuestra mente en ser uno de ellos, ahora ya, con un poco más de experiencia nos damos cuenta que seguimos jugando con nuestra mente para poder hacer realidad esas ideas de crear un robot, y la ingeniería con sus grandes y asombrosas matemáticas es uno de los caminos que nos ayudan a hacer realidad este sueño, basta con darnos cuenta la complejidad que resulta obtener el modelo dinámico del robot para poder decir: "como seria esto posible sin un conocimiento cálculo, dinámica, etc."

b. Aplicaciones de la robótica en el sistema industrial

- Automotriz:

La introducción de los robots ha sido facilitada por la técnica de organización y división del trabajo, sobre todo en la producción en masa, basadas en la mayor especialización, simplificación y repetitividad de las tareas productivas, lo que ha facilitado el diseño y programación de los robots.

Entre las principales aplicaciones no industriales de los robots. Es necesario mencionar su utilización en plantas de energía nuclear, en la exploración submarina, la minería, construcciones, agricultura, medicina etc.

Las principales aplicaciones industriales son las siguientes:

- a. Fundición en molde (die-casting). Esta fue la primera aplicación industrial.
- b. Soldadura de Punto. Actualmente es la principal área la presente generación de robot. Ampliamente utilizada en la industria automotriz. En promedio, este tipo de robot. reduce a la mitad la fuerza laboral necesaria.
- c. Soldaduras de Arco. No requiere de modificaciones sustanciales en el equipo de soldadura y aumenta la flexibilidad y la velocidad.
- d. Moldeado por Extrusión. De gran Importancia por creciente demanda de partes especializadas de gran complejidad y precisión.
- e. Forjado (Forging). La principal aplicación es la manipulación de partes metálicas calientes.
- f. Aplicaciones de Prensado (press work). Partes y, panales de vehículos y estructuras de aviones, electrodomésticos y otros productos metalmecánicos. Esta es un área de rápido desarrollo de nuevos tipos de robot.
- g. Pinturas y Tratamiento de Superficies. El mejoramiento de las condiciones de trabajo y la flexibilidad han sido las principales razones para el desarrollo de estas aplicaciones.
- h. Moldeado Plástico. Descarga de máquinas de inyección de moldes, carga de moldes, paletización y empaque de moldes, etc. Alta contribución al mejoramiento de las condiciones de trabajo, al ahorro de mano obra, a la reducción del tiempo de producción, y al aumento de la productividad.
- i. Aplicaciones en la Fundición. Carga y descarga de máquinas, manejo de materiales calientes, manejo de moldes, etc. Las difíciles condiciones de trabajo hacen necesarios los robot., aunque ha sido muy difícil su diseño y eficacia.
- j. Carga y Descarga de Máquina Herramientas. Los robots aumentan la flexibilidad y versatilidad de las máquinas herramientas y permiten su articulación entre sí. Contribuyen a la reducción de stocks, minimizan costos

del trabajo directo e indirecto, aumentan la calidad de la producción y maximizar la utilización del equipo.

- k. En aparatos y maquinaria eléctrica y electrónica, juguetes, ingeniería mecánica, industrial automotriz, etc.

Estas diversas aplicaciones industriales implican la clasificación de los robots en cuatro tipos de operaciones efectuadas:

- A. Robots de manejo de materiales: carga y descarga de máquinas herramienta, moldeado de plástico.
- B. Robot. de tratamiento de superficie: pintura, las piezas,
- C. Robots de en ensamblaje y transferencia.
- D. Robot. de soldadura, y
- E. Robots de procesamiento por calor; moldeado, prensado, etc.

Los robots son utilizados por una diversidad de procesos industriales como lo son: la soldadura de punto y soldadura de arco, pinturas de spray, transportación de materiales, molienda de materiales, moldeado en la industria plástica, máquinas-herramientas, y otras más.

Aplicación de transferencia de material

Las aplicaciones de transferencia de material se definen como operaciones en las cuales el objetivo primario es mover una pieza de una posición a otra. Se suelen considerar entre las operaciones más sencillas o directas de realizar por los robots. Las aplicaciones normalmente necesitan un robot poco sofisticado, y los requisitos de enclavamiento con otros equipos son típicamente simples.

Carga y descarga de máquinas.

Estas aplicaciones son de manejo de material en las que el robot se utiliza para servir a una máquina de producción transfiriendo piezas a/o desde las máquinas.

Existen tres casos que caen dentro de esta categoría de aplicación:

-Carga/Descarga de Máquinas. El robot carga una pieza de trabajo en bruto en el proceso y descarga una pieza acabada. Una operación de mecanizado es un ejemplo de este caso.

-Carga de máquinas. El robot debe de cargar la pieza de trabajo en bruto a los materiales en las máquinas, pero la pieza se extrae mediante algún otro medio. En una operación de prensado, el robot se puede programar para cargar láminas de metal en la prensa, pero las piezas acabadas se permite que caigan fuera de la prensa por gravedad.

-Descarga de máquinas. La máquina produce piezas acabadas a partir de materiales en bruto que se cargan directamente en la máquina sin la ayuda de robots. El robot descarga la pieza de la máquina. Ejemplos de esta categoría incluyen aplicaciones de fundición de troquel y moldeado plástico. La aplicación se tipifica mejor mediante una célula de trabajo con el robot en el centro que consta de la máquina de producción, el robot y alguna forma de entrega de piezas.

Operaciones de procesamiento.

Además de las aplicaciones de manejo de piezas, existe una gran clase de aplicaciones en las cuales el robot realmente efectúa trabajos sobre piezas. Este trabajo casi siempre necesita que el efector final del robot sea una herramienta en lugar de una pinza. Por tanto la utilización de una herramienta para efectuar el trabajo es una característica distinta de este grupo de aplicaciones. El tipo de herramienta depende de la operación de procesamiento que se realiza.

Soldadura por puntos.

Como el término lo sugiere, la soldadura por puntos es un proceso en el que dos piezas de metal se sueldan en puntos localizados al hacer pasar una gran corriente eléctrica a través de las piezas donde se efectúa la soldadura.

Soldadura por arco continua.

La soldadura por arco es un proceso de soldadura continua en oposición a la soldadura por punto que podría llamarse un proceso discontinuo. La soldadura de arco continua se utiliza para obtener uniones largas o grandes uniones soldadas en las cuales, a menudo, se necesita una cierre hermético entre las dos piezas de metal que se van a unir. El proceso utiliza un electrodo en forma de barra o alambre de metal para suministrar la alta corriente eléctrica de 100 a 300 amperios.

Recubrimiento con spray

La mayoría de los productos fabricados de materiales metálicos requieren de alguna forma de acabado de pintura antes de la entrega al cliente.

La tecnología para aplicar estos acabados varía en la complejidad desde métodos manuales simples a técnicas automáticas altamente sofisticadas. Se dividen los métodos de recubrimiento industrial en dos categorías:

- 1.- Métodos de recubrimiento de flujo e inmersión.
- 2.- Métodos de recubrimiento al spray.

Los métodos de recubrimiento mediante flujo de inmersión se suelen considerar que son métodos de aplicar pintura al producto de baja tecnología. La inmersión simplemente requiere sumergir la pieza o producto en un tanque de pintura líquida.

Otras operaciones de proceso.

Además de la soldadura por punto, la soldadura por arco, y el recubrimiento al spray existe una serie de otras aplicaciones de robots que utilizan alguna forma de herramienta especializada como efector final. Operaciones que están en esta categoría incluyen:

- Taladro, acanalado, y otras aplicaciones de mecanizado.
- Rectificado, pulido, desbarbado, cepillado y operaciones similares.
- Remachado.
- Corte por chorro de agua.
- Taladro y corte por láser.

- Electrónica

La electrónica es la ciencia aplicada fundamental de la tecnología moderna, todo equipo que necesite un control rápido y efectivo hace uso de la tecnología electrónica. Con los componentes electrónicos se elaboran los circuitos de control y comunicación de los robots. A través de la circuitería electrónica se realizan todas las funciones de detección de las señales que se emplean para manipular y controlar los servomecanismos de los robots. La comunicación de datos e instrucciones de control y operación son realizados por elementos electrónicos. La electrónica esta en todos los dispositivos modernos, radios, televisión, robots, juguetes, lavadoras, automóviles, computadores, teléfonos y muchos más.

Esta ciencia la cual ya es muy antigua está teniendo una relevancia muy importante en los últimos tiempos. Existe un amplio desconocimiento de lo que significa robótica y de sus alcances. La robótica tiene muchos campos de aplicación desde la oceanografía, la industria, la medicina, el hogar y las armas. Hay muchos tipos de robots. La robótica es una ciencia mixta que abarca muchos campos del saber, mecánica, electrónica, informática, servomecanismos, ergonomía. Inteligencia artificial y otras.

- Manufactura

La automatización ha llegado a ser parte de los nuevos tiempos en el área de la industria, de no ser así, no podríamos obtener los resultados en la producción en masa de muchos de los implementos que usamos diariamente, la manufactura

integrada por computadora (CIM por sus siglas en inglés) logra automatizar y optimizar por medio de computadoras estos procesos.

La manufactura integrada por computadora (o CIM: Computer Integrated Manufacturing) consiste en la automatización de un proceso completo de manufactura mediante el uso de computadoras. La acuñación del nombre se atribuye generalmente al fallecido Dr. Joseph Harrington, cuando publica un libro con el mismo nombre en 1973.

CIM propone utilizar el poder de análisis, cálculo y procesamiento de las computadoras al servicio de la producción de bienes de mercado y cubre varios aspectos de la industria, que van desde el diseño, la ingeniería, la manufactura hasta la logística, el almacenamiento y la distribución de los productos. El objetivo de esta tecnología es incrementar la capacidad de manufacturar piezas, productos terminados o semielaborados usando el mismo grupo de máquinas. Para ello se requiere que las herramientas utilizadas sean flexibles y capaces de modificar su programación adaptándose a los nuevos requerimientos del mercado.

El diseño de un sistema (CIM) significa la aplicación de las teorías de sistemas a las empresas de manufactura, significa ver a la organización como unidad con ciertas entradas y ciertas salidas deseables, así como el diseño de sistemas basados en computadora e integrados por personas para lograr que las entradas se transformen en salidas. Sin embargo, la transformación de una compañía de manufactura, cuya automatización sea parcial, en una con manufactura integrada por computadora, es una tarea compleja y difícil. Los retos tecnológicos son los problemas menores; en general, estos lo pueden resolver profesionales competentes con presupuestos apropiados. Es fácil diseñar un proceso que sea automático por completo, sin embargo, es difícil diseñar una serie de pasos que

lleven de los sistemas manuales presentes y de la isla de automatización a la manufactura integrada por computadora (CIM) en una secuencia económica y sensata. Por esta razón, el sistema (CIM) no se puede vender como un producto o servicio en paquete; cada fábrica necesita consideraciones muy diferentes.

La automatización total abre perspectivas de flexibilidad que no se pueden lograr en la manufactura convencional; por lo tanto, las fábricas basadas en el sistema CIM responden con mayor rapidez que las convencionales a las cambiantes necesidades del mercado.

Aparte de la infraestructura, la administración de la empresa también debe de estar dentro del sistema CIM, ya que también se necesita que la empresa en general cambie el punto de vista para que la manufactura integrada por computadoras sea aún más efectiva.

Subsistemas de CIM.

Un sistema CIM debe ser flexible en su fabricación, ya que la empresa puede ser rápidamente modificada para producir productos diferentes, o donde el volumen de productos puede ser cambiado rápidamente con la ayuda de computadores, y esto es logrado gracias a subsistemas. Algunos subsistemas que se pueden encontrar en una operación CIM pueden ser:

Técnicas automatizadas:

- DAO (diseño asistido por el computador)
- CAE (ingeniería automatizada por computador)
- DISCO DE LEVA (fabricación automatizada)
- CAPP (planificación de proceso automatizada)

- CAQ (garantía de calidad automatizada)
- PPC (planificación de producción y control)
- ERP (planificación de recurso de la empresa)

Un sistema de negocio integrado por una base de datos común.

Dispositivos y equipo requerido:

- CNC, herramientas numéricas controladas por computadoras
- DNC, herramientas de control numéricas dirigidas por computadoras
- PLCs, reguladores Programables lógicos
- Robótica
- Ordenadores
- Software
- Reguladores
- Redes
- Interrevestimiento
- Supervisión de equipo

Tecnologías:

- FMS, (sistema flexible de la fabricación)
- ASRS, almacenaje automatizado y sistema de recuperación
- AGV, vehículo automatizado dirigido
- Robótica
- Sistemas de transporte automatizados

Hay 5 niveles fundamentales los que son:

Nivel de controlador de planta: Es el más alto nivel de la jerarquía de control, es representado por la(s)

Computadora(s) central(es) (mainframes) de la planta que realiza las funciones corporativas como: administración de recursos y planeación general de la planta.

Nivel de controlador de área: Es representado por las computadoras (minicomputadoras) de control de las

Operaciones de la producción. Es responsable de la coordinación y programación de las actividades de las celdas de manufactura, así como de la entrada y salida de material. Conectada a las computadoras centrales se encuentra(n) la(s) computador(as) de análisis y diseño de ingeniería donde se realizan tareas como diseño del producto, análisis y prueba. Adicionalmente, este nivel realiza funciones de planeación asistida por

Computadora (CAP, por sus siglas en inglés), diseño asistido por computadora (CAD, por sus siglas en inglés) y planeación de requerimientos de materiales (MRP, por sus siglas en inglés).

Nivel de controlador de celda: La función de este nivel implica la programación de las órdenes de manufactura y coordinación de todas las actividades dentro de una celda integrada de

Manufactura. Es representado por las computadoras (minicomputadoras, PC's y/o estaciones de trabajo). En general, realiza la secuencia y control de los controladores de equipo.

Nivel de controlador de procesos o nivel de controlador de estación de trabajo: Incluye los controladores de equipo, los cuales permiten automatizar el funcionamiento de las máquinas. Entre estos se encuentran los controladores de robots (RC's), controles lógicos programables (PLC's), CNC's,

y microcomputadores, los cuales habilitan a las máquinas a comunicarse con los demás (incluso en el mismo nivel) niveles jerárquicos.

Nivel de equipo: Es el más bajo nivel de la jerarquía, está representado por los dispositivos que ejecutan los comandos de control del nivel próximo superior. Estos dispositivos son los actuadores, relevadores, manejadores, switches y válvulas que se encuentra directamente sobre el equipo de producción. De una manera más general se considera a la maquinaria y equipo de producción como representativos de este nivel.

- Textil

De todas las actividades industriales la elaboración de textiles es una de las más antiguas por la necesidad masiva de vestir, cubrir y proteger el cuerpo de las condiciones ambientales, por moda, por necesidad o por razones de obligación social; gran parte de la tarea de la fabricación de telas se basa en la limpieza y tratamiento de la materia prima para su posterior procesamiento para obtener como producto, este trabajo era realizado casi en un 90% de manera manual hasta mediados del siglo XVIII cuando a paso lento comenzaron a notarse avances en materia de maquinaria que pudieron irse implementando para ser utilizadas en la manufactura de las telas y textiles.

Como toda historia la de la industria textil ha tenido altas y bajas en la búsqueda de desarrollo atravesando por guerras mundiales y desastres naturales, hoy por hoy está viviendo un florecimiento gracias a la automatización y robotización en la industria textil en donde se pueden encontrar robots con programaciones específicas de movimiento para trabajar y elaborar las telas además de la utilización de láser para acabados de corte más perfectos y exactos.

Una de las novedades que la robotización en la industria textil trajo fue la inserción de una máquina que se encarga de agregar los colores a los hilos con los que se realizara la manufacturación de la tela, dejando atrás el proceso de teñido antiguo donde se aplicaba el tinte a una solución de agua y sal para posteriormente integrar el tejido que se quería teñir, además de los tintes utilizados anteriormente como la tinta de moluscos y la corteza de roble.

- Energía

Un robot consta de tres componentes principales: el componente mecánico, que da cuenta de la estructura o anatomía del robot; el componente energético, que permite animar el robot desde el punto de vista de la energía eléctrica, la energía hidráulica, la energía neumática, la energía nuclear, la energía solar, etc.; y el componente de control, que equivale al cerebro del robot y que está representado por una computadora.

Mecánicamente, el robot puede tener una estructura sencilla o robusta. Puede ser estacionario o móvil. Si es móvil, puede estar provisto de ruedas, desplazarse sobre vías o rieles, o puede tener dispositivos que sustituyan las funciones de piernas y/o pies.

Desde el punto de vista del suministro de energía (potencia motriz) para animar el robot, es decir, darle movimiento, la fuente de energía puede ser eléctrica, hidráulica, neumática, solar y nuclear. Generalmente, en la industria, la mayoría de los robots utilizan energía eléctrica para moverse o desplazarse.

No obstante, también existen robots que se procuran movimientos a través de la energía hidráulica, neumática y nuclear.

El componente de control del robot está representado por la computadora. Es la computadora la que se encarga de hacer que el robot ejecute las tareas tal cual han sido programadas.

- **Petróleo y gas**

La industria del petróleo y del gas, en comparación con otras industrias que se encuentran totalmente automatizadas, como por ejemplo la industria automotriz (que emplea más robots que todas las otras industrias juntas). En nuestra industria, siempre se ha confiado en el ingeniero humano y en su experiencia para progresar y ejecutar sus operaciones. Sin embargo, el accidente de British Petroleum en el Golfo de México, en el año 2010, ayudó a cambiar esa actitud. Once trabajadores murieron cuando la plataforma Deepwater Horizon se incendió y se hundió. La compañía noruega Statoil ha proyectado que la automatización puede reducir a la mitad el número de trabajadores que se necesitan en una plataforma offshore, y ayudar a reducir en un 25% el tiempo de trabajo. Como otros robots de servicio, los robots en los campos petroleros realizan tareas sucias, aburridas y peligrosas. Estas actividades incluyen operaciones de perforación direccional automatizada y perforación continua de circuito cerrado. Los vehículos de operación remota (ROV) se han convertido en una herramienta importante en la perforación, el desarrollo y en la reparación offshore. Actualmente, los vehículos submarinos autónomos (AUV) están listos para ganar el mercado. Un ROV (Remote Operated Vehicle) es un robot submarino no tripulado y conectado a una unidad de control en la superficie por medio de un

cable umbilical. La energía y las órdenes se envían mediante un mando de control a través del cable al robot. Requieren un equipo humano para controlar y operar las funciones que el sistema es capaz de realizar.

Los AUV (Autonomus Underwater Vehicle) pueden desarrollar una tarea programada, interactuar con su entorno, cumplir su objetivo y retornar a un lugar pre-designado, sin intervención del operador. A diferencia de los ROV, no utiliza cable umbilical. Cuando las compañías petroleras comenzaron a explorar en aguas más profundas, el límite de profundidad del lecho marino para las perforaciones fue definido por la profundidad máxima de intervención humana. Con equipos de buceo especializado, ese límite llegó a los 300 metros. Los submarinos tripulados eran una opción, pero solo podían sumergirse hasta los 600 metros. Por debajo de estas profundidades, los ROV constituyen la única opción para las operaciones de intervención. Por este motivo, todos los equipos de perforación flotantes actualmente en operación poseen al menos un ROV. Incluso en aquellos pozos perforados en tirantes de agua en las que la interacción humana es posible. Los ROV han reemplazado a los hombres, como la principal forma de intervención subacuática. Los ROV pueden ejecutar una multitud de tareas, siempre que hayan sido diseñadas antes del comienzo de la operación. A diferencia de los operadores humanos, que pueden responder a condiciones cambiantes, para estos dispositivos es difícil ejecutar una tarea por lo más sencilla que sea, si esta difiere de lo planeado. La planeación del operador es la fase más crítica de la ejecución del trabajo, y la reacción es una función de la capacidad del operador del ROV para comprender y responder a la situación en cuestión.

c. Uso adecuado de los equipos especializados para automatización



PXI es una plataforma robusta basada en PC para sistemas de medidas y automatización. PXI combina características de bus eléctrico PCI con paquete Eurocard modular de CompactPCI y después añade buses de sincronización especializada y características clave de software. PXI es una plataforma de alto rendimiento y bajo costo de implementación para aplicaciones de pruebas de manufactura, militares y aeroespaciales, monitoreo de máquinas, automotrices y pruebas industriales. Desarrollado en 1997 y publicado en 1998, PXI es un estándar abierto en la industria gobernado por PXI Systems Alliance (PXISA), un grupo de más de 70 compañías contratadas para promocionar el estándar PXI , asegurar su interoperabilidad y mantener la especificación PXI.

Desde el invento del estándar en 1997 y la fundación del consorcio de la industria Alianza para Sistemas PXI en 1998, National Instruments ha sido líder en ofrecer los beneficios de PXI para los ingenieros y científicos en todo el mundo. Con la colección más grande de chasis, controladores y temporización y sincronización y más de 600 módulos, usted puede resolver prácticamente cualquier reto de ingeniería.

"El mercado de PXI está creciendo y ganando acción en comparación con los instrumentos de configuración rack-and-stack y National Instruments es sin lugar a duda el líder en equipos basados en PXI para pruebas y medidas. Lo cual no resulta sorprendente considerando la variedad de su línea de productos PXI."

Conexión PCI

Permite intercambiar información entre dispositivos como tarjetas gráficas, de sonido o de red con el procesador de tu equipo.

La historia de los diferentes buses, que es como se llaman de forma genérica a estos enlaces, es bien larga, desde los ISA a los EISA pasando por los más

recientes PCI. PCI Express evoluciona a partir de ellos para mejorar tanto la velocidad, como la eficiencia.

La diferencia fundamental entre PCI Express y su anterior versión es que utiliza una conexión de tipo serie. Esto, que ya lo hemos visto en otros dispositivos como los discos duros que pasaron de un estándar paralelo como PATA a uno serie como SATA permite mayores velocidades y que no existan caídas de rendimiento al conectar varios dispositivos al mismo equipo.

Controladores de automatización programables.

Tecnología industrial orientada al control automatizado, al diseño de prototipos y a la medición. El PAC se refiere al conjunto formado por un controlador (una CPU típicamente), módulos de entradas y salidas, y uno o múltiples buses de datos que lo interconectan todo.

Este controlador combina eficientemente la fiabilidad de control de un autómata (controlador lógico programable o PLC) junto a la flexibilidad de monitorización y cálculo de un PC. A veces incluso se le une la velocidad y personalización de la microelectrónica. Los PACs pueden utilizarse en el ámbito investigador (prototipaje rápido de controladores o RCP), pero es sobre todo en el industrial, para control de máquinas y procesos, donde más se utiliza. A destacar los siguientes: múltiples lazos cerrados de control independientes, adquisición de datos de precisión, análisis matemático y memoria profunda, monitorización remota, visión artificial, control de movimiento y robótica, seguridad controlada, etc.

Los PAC se comunican usando los protocolos de red abiertos como TCP/IP, OPC (OLE for process control), SMTP, puerto serie (con Modbus por ejemplo), etc, y es compatible con los privados (CAN, Profibus, etc).

Un ejemplo claro de utilización es en un sistema de control de un proceso determinado. El elemento controlador es el sitio donde se toman todas las decisiones sobre las acciones a tomar. Se le puede considerar el "cerebro" del sistema. Debe tomar decisiones basadas en ciertas pautas o valores requeridos. Los valores establecidos son introducidos en el sistema por el hombre.

El sistema PAC integra control, procesamiento de información y la creación de redes en un solo controlador. Debido a que la PAC es un equipo industrial con forma de PLC, que tiene todas las características de una PC de gama alta, incluyendo grande almacenamiento de memoria, disco duro, sistema operativo y, si se desea, incluso los procesadores multi-core. Independientemente de si la instalación es un sistema discontinuo, sistema de proceso continuo, híbrido; control de la máquina de alta velocidad o un sistema de visión artificial, el PAC proporciona un único sistema de control capaz de realizar todas estas tareas de aplicación y en algunos casos múltiples tareas al mismo tiempo. Los PACs pueden integrarse en el HMI local o un panel de visualización HMI local, puede estar ubicado cerca de la instalación PAC. Las PAC pueden incluso funcionar como servidores de datos móviles con el servidor SQL y la tecnología de almacenamiento de modo que los datos puedan ser pre-tratados, utilizados para el control de la PAC, y transmitidos a la base de datos de Enterprise. Tenga en cuenta que el motor de programación IEC 61131-3 es fundamental para la capacidad operativa de la PAC al igual que los otros de-facto y estándares abiertos como el OPC. Tenga en cuenta también que con el servidor SQL a bordo, los datos pueden ser transmitidos por medio de la base de datos con formato nativo de SQL directamente de la PAC a la DB Enterprise o de MS Access para funciones locales relacionadas con el control.

Una de las preocupaciones acerca de los PAC siempre ha sido el desempeño no determinista de Windows y sus derivados, y Ethernet y sus derivados. Advantech, por ejemplo, ha puesto a prueba su plataforma PAC de determinista (tiempo real) y su rendimiento y ha demostrado lo que dicen es 1 ms tiempo comprobado de actualización de 32 modelos I/O digitales. Esto confirma el uso continuado de los PAC para la adquisición de datos con alta rapidez, control de movimiento y las aplicaciones de visión artificial donde las tasas de actualización de menos de 20 ms son críticos para controlar el rendimiento.

Aplicaciones emergentes La versatilidad de la plataforma PAC se muestra claramente por la amplia variedad de aplicaciones para las que se ha usado, tanto en la fabricación y fuera de las aplicaciones tradicionales de fabricación establecidas. Integración del control, HMI, procesamiento de la información y la creación de redes en un único sistema de control hace que sea posible el uso de la plataforma en aplicaciones tan dispares como la gestión de la bomba de gas en una estación de gasolina, las aplicaciones GPS móviles en camiones y entrega, plataformas de pruebas automatizadas en montaje de la electrónica discreta , la inspección del producto final, la integración de RFID, e incluso la gestión de activos, además del proceso tradicional y las aplicaciones del sistema de control discreto. Y en las industrias de procesos tradicionales, la versatilidad de la PAC puede ser demostrado por su uso en un sistema concreto de control de lotes. Sustituyendo un PLC existente y la arquitectura IPC por una única plataforma de control en el Advantech Apex PAC que resuelven todos los problemas de comunicación entre el PLC (que utiliza comunicaciones serie) y el IPC, que fue el Ethernet habilitado. Permitted el programa de control SoftLogic y el software HMI para ejecutar de forma simultánea en el sistema PAC integrado. El software HMI incluye selección de recetas, iniciar/detener el control y la recolección de datos de fabricación y capacidades de revisión, mientras que el programa de control SoftLogic proporciona la medición del peso en tiempo real de alta velocidad.

Además, el uso de un PAC activado con programación de lenguaje estructurado para ser utilizado por el algoritmo de pesaje, en lugar de un diagrama de escalera o de programación de bloque de función.

La PAC incluye la capacidad de monitoreo remoto de datos en tiempo real directamente desde el centro de control con protocolo abierto estándar Modbus / TCP sobre Ethernet 100Mb estándar.

La PAC no es simplemente una computadora personal o computadora de placa única puesto en un recinto robusto. La naturaleza de las aplicaciones de automatización industrial, como la solicitud de procesamiento por lotes de cemento, significa que los PAC deben ser diseñados específicamente para ser PACs con los estándares más altos para el diseño y los componentes que los PC estándar o incluso el IPC estándar. Características tales como un alto rechazo de modo común, filtros de potencia de software, protección de voltaje, sobre protección de apagado por temperatura, sobre protección del apagado actual y cortocircuitos son estándares en el PAC moderno, mientras que ellos no pueden estar en cualquier diseño de la placa única adaptada. Mientras que la PAC es más versátil que el PLC, el diseño de una PAC debe ser de acuerdo a la norma IEC-61131-2, tipo 1. el mismo nivel de diseño como un PLC. El moderno PAC es generalmente más confiable que un PLC, y sin duda en una clase por sí mismo en comparación con el IPC y diseños de placa única OEM. Con su certificación estricta como un PLC, y arquitectura de controlador dual, entrada de poder doble y las funciones del sistema de copia de seguridad, es resistente, duradero y capaz. El moderno PAC tiene una topología flexible. Esto significa que su arquitectura física abierta permite la expansión ilimitada de funciones de I/O y de control. Algunos modelos de la PAC incluso permiten a la CPU para ser cambiado y mejorado sin perturbar el I/O, y, por el contrario, permiten el intercambio en caliente de módulos de I/O.

Una PAC te ofrece:

- Funcionalidad multidominio, incluyendo la lógica, el control continuo y el movimiento en una sola plataforma
- Abrir, arquitecturas modulares con estándares de-facto para las interfaces y protocolos de red
- Una sola plataforma de desarrollo multidisciplinar que incorpora el etiquetado común y una sola base de datos
- Las herramientas de software que permiten el diseño de un proceso de flujo a través de varias máquinas o unidades de proceso ARC Advisory Group ha cuantificado el éxito de la PAC. Señalan que el crecimiento del mercado PLC 2007-2012 se estima en 6.5% CAGR, mientras que se espera que el crecimiento del mercado PAC sea de 10,1% durante el mismo período.

Crio de Nationals Instruments

Es una plataforma de adquisición de datos y control impulsada por tecnología de entrada y salida reconfigurable (reconfigurable I/O, RIO). CompactRIO combina un procesador embebido en tiempo real, un arreglo de compuertas lógicas programables (FPGA por sus siglas en inglés) de alto rendimiento y módulos de entrada/salida intercambiables. Los módulos de E/S se conectan directamente al FPGA, proporcionando personalización de bajo nivel para temporización y procesamiento de señales de E/S. El FPGA se conecta a un procesador embebido en tiempo real vía un bus PCI de alta velocidad. Mediante la plataforma de desarrollo LabVIEW es posible la transferencia de datos entre los módulos de E/S y el FPGA, y entre el FPGA y el procesador embebido para análisis en tiempo real, procesamiento posterior, registro de datos o comunicación a un servidor conectado en red.

El sistema de circuitos FPGA de RIO es un motor de computación reconfigurable de procesamiento paralelo que ejecuta aplicaciones integradas de LabVIEW determinísticamente. CompactRIO permite implementar sistemas de control PID analógicos de lazos múltiples a razones de 100 [kmuestras/s] y sistemas de control digital a razones de lazo de hasta 1 [Mmuestras/s] y evaluar escalones de lógica booleana de menos de 25 [ns]. Los principales componentes de un sistema de LabVIEW FPGA son el Módulo LabVIEW FPGA y el hardware RIO. La programación del código en LabVIEW ofrece la ventaja de que la representación del tiempo, concurrencia y paralelismo del hardware del FPGA es intuitivo al momento de implementar. Adicionalmente, se puede integrar el hardware del FPGA con otros componentes de sistema de medición y control usando VIs de Interfaz de LabVIEW FPGA. Un sistema de control tradicional cuenta con tres etapas:

- El software de la aplicación corriendo en Windows o en un sistema operativo de tiempo real (RTOS).
- El software driver y las funciones para interactuar con el hardware.
- El hardware de Entrada/Salida. El esquema anterior varía cuando se usa un sistema basado en LabVIEW FPGA. El software de la aplicación se ejecuta en Windows o en un RTOS y la interfaz con el hardware se realiza por medio de un driver National Instruments RIO (NI-RIO). Es necesario utilizar hardware para interactuar con el proceso, que en este caso son dispositivos de la serie C, y la funcionalidad de este dispositivo es modificable usando LabVIEW FPGA. En síntesis, se trabaja en una plataforma FPGA de Entrada/Salida (I/O) que ofrece buen rendimiento, flexibilidad de aplicaciones y diseño de hardware a nivel de tarjeta.

Los controladores embebidos CompactRIO ofrecen ejecución autónoma para aplicaciones determinísticas de LabVIEW Real-Time. CompactRIO opera con un procesador industrial de 200 [MHz] con capacidad de efectuar operaciones de punto fijo de procesamiento de señales y análisis para ciclos de control de 1 kHz. Los controladores NI CompactRIO tienen entradas de doble suministro de 11 a 30 [VDC], un consumo de 7 a 10 [W] y un rango de temperatura de -40 a 70 [°C]. Los módulos de E/S de CompactRIO tienen un aislamiento de hasta 2,300 [VRMS] (tolerancia) y un aislamiento de 250 [VRMS] (continuo). Cada componente viene con una variedad de certificaciones y graduaciones internacionales de seguridad, ambientales y de compatibilidad. CompactRIO está diseñado para utilizar las herramientas de desarrollo gráfico de LabVIEW para adaptar el hardware reconfigurable a una variedad de industrias y aplicaciones. El sistema incluye el controlador en tiempo real cRIO-9002 con procesadores industriales de punto flotante de 200 [MHz], el chasis cRIO-9102 reconfigurable de ocho ranuras con FPGA de 1 millón de compuertas, el módulo de ocho Entradas Analógicas 9201 que acepta un rango de ± 10 [V], y finalmente el módulo de 4 Salidas Analógicas con un rango de salida de voltaje de ± 10 [V]. En esta tesis, se programó el hardware de CompactRIO usando LabVIEW 2009 en Windows XP, el Módulo de LabVIEW Real-Time y el Módulo de LabVIEW FPGA.

Programación de controladores mediante LAB VIEW

LabVIEW constituye un revolucionario sistema de programación gráfica para aplicaciones que involucren adquisición, control, análisis y presentación de datos. Las ventajas que proporciona el empleo de LabVIEW se resumen en las siguientes:

- Se reduce el tiempo de desarrollo de las aplicaciones al menos de 4 a 10 veces, ya que es muy intuitivo y fácil de aprender.

- Dota de gran flexibilidad al sistema, permitiendo cambios y actualizaciones tanto del hardware como del software.
- Da la posibilidad a los usuarios de crear soluciones completas y complejas.
- Con un único sistema de desarrollo se integran las funciones de adquisición, análisis y presentación de datos.
- El sistema está dotado de un compilador gráfico para lograr la máxima velocidad de ejecución posible.
- Tiene la posibilidad de incorporar aplicaciones escritas en otros lenguajes.

LabVIEW es un entorno de programación destinado al desarrollo de aplicaciones, similar a los sistemas de desarrollo comerciales que utilizan el *lenguaje C* o *BASIC*. Sin embargo, LabVIEW se diferencia de dichos programas en un importante aspecto: los citados lenguajes de programación se basan en líneas de texto para crear el código fuente del programa, mientras que LabVIEW emplea la programación gráfica o *lenguaje G* para crear programas basados en diagramas de bloques.

Para el empleo de LabVIEW no se requiere gran experiencia en programación, ya que se emplean iconos, términos e ideas familiares a científicos e ingenieros, y se apoya sobre símbolos gráficos en lugar de lenguaje escrito para construir las aplicaciones. Por ello resulta mucho más intuitivo que el resto de lenguajes de programación convencionales.

LabVIEW posee extensas librerías de funciones y subrutinas. Además de las funciones básicas de todo lenguaje de programación, LabVIEW incluye librerías específicas para la adquisición de datos, control de instrumentación VXI, GPIB y comunicación serie, análisis presentación y guardado de datos.

LabVIEW también proporciona potentes herramientas que facilitan la depuración de los programas.

Los programas desarrollados mediante LabVIEW se denominan *Instrumentos Virtuales*

(*Vis*), porque su apariencia y funcionamiento imitan los de un instrumento real. Sin embargo son análogos a las funciones creadas con los lenguajes de programación convencionales. Los *Vis* tienen una parte interactiva con el usuario y otra parte de código fuente, y aceptan parámetros procedentes de otros *Vis*.

Todos los *Vis* tienen un *panel frontal* y un *diagrama de bloques*. Las *paletas* contienen las opciones que se emplean para crear y modificar los *Vis*. A continuación se procederá a realizar una somera descripción de estos conceptos.

A) Panel Frontal

Se trata de la interfaz gráfica del *VI* con el usuario. Esta interfaz recoge las entradas procedentes del usuario y representa las salidas proporcionadas por el programa. Un *panel frontal* está formado por una serie de botones, pulsadores, potenciómetros, gráficos, etc.

Cada uno de ellos puede estar definido como un *control* (a) o un *indicador* (b). Los primeros sirven para introducir parámetros al *VI*, mientras que los indicadores se emplean para mostrar los resultados producidos, ya sean datos adquiridos o resultados de alguna operación.

B) Diagrama de bloques

El *diagrama de bloques* constituye el código fuente del *VI*. En el *diagrama de bloques* es donde se realiza la implementación del programa del *VI* para controlar o realizar cualquier procesado de las entradas y salidas que se crearon en el *panel frontal*.

El *diagrama de bloques* incluye *funciones* y *estructuras* integradas en las librerías que incorpora LabVIEW. En el *lenguaje G* las *funciones* y las *estructuras* son nodos elementales. Son análogas a los operadores o librerías de funciones de los lenguajes convencionales.

Los *controles* e *indicadores* que se colocaron previamente en el Panel Frontal, se materializan en el diagrama de bloques mediante los *terminales*.

El *diagrama de bloques* se construye conectando los distintos objetos entre sí, como si de un circuito se tratara. Los cables unen terminales de entrada y salida con los objetos correspondientes, y por ellos fluyen los datos.

LabVIEW posee una extensa biblioteca de *funciones*, entre ellas, aritméticas, comparaciones, conversiones, funciones de entrada/salida, de análisis, etc.

Las *estructuras*, similares a las declaraciones causales y a los bucles en lenguajes convencionales, ejecutan el código que contienen de forma condicional o repetitiva (bucle *for*, *while*, *case*,...).

Los cables son las trayectorias que siguen los datos desde su origen hasta su destino, ya sea una función, una estructura, un terminal, etc. Cada cable tiene un color o un estilo diferente, lo que diferencia unos tipos de datos de otros.

C) Paletas.

Las *paletas* de LabVIEW proporcionan las herramientas que se requieren para crear y modificar tanto el *panel frontal* como el *diagrama de bloques*. Existen las siguientes paletas:

Paleta de herramientas (Tools palette)

Se emplea tanto en el *panel frontal* como en el *diagrama de bloques*. Contiene las herramientas necesarias para editar y depurar los objetos tanto del *panel frontal* como del *diagrama de bloques*.

Las opciones que presenta esta paleta son las siguientes:

Operating tool – Cambia el valor de los controles.

Positioning tool – Desplaza, cambia de tamaño y selecciona los objetos.

Labeling tool – Edita texto y crea etiquetas.

Wiring tool – Une los objetos en el *diagrama de bloques*.

Object Pop-up Menu tool – Abre el menú desplegable de un objeto.

Scroll tool – Desplaza la pantalla sin necesidad de emplear las barras de desplazamiento.

Breakpoint tool – Fija puntos de interrupción de la ejecución del programa en *VIs*, *funciones* y *estructuras*.

Probe tool – Crea puntos de prueba en los cables, en los que se puede visualizar el valor del dato que fluya por dicho cable en cada instante.

Color Copy tool – Copia el color para después establecerlo mediante la siguiente herramienta.

Color tool – Establece el color de fondo y el de los objetos

Paleta de controles (Controls palette)

Se utiliza únicamente en el *panel frontal*. Contiene todos los *controles* e *indicadores* que se emplearán para crear la interfaz del *VI* con el usuario.

El menú *Controls* de la ventana correspondiente al panel frontal contiene las siguientes opciones:

Numeric – Para la introducción y visualización de cantidades numéricas.

Boolean – Para la entrada y visualización de valores booleanos.

String & Table – Para la entrada y visualización de texto.

List & Ring – Para visualizar y/o seleccionar una lista de opciones.

Array & Cluster – Para agrupar elementos.

Graph – Para representar gráficamente los datos.

Path & RefNum – Para gestión de archivos.

Decorations – Para introducir decoraciones en el *panel frontal*. No visualizan datos.

User Controls – Para elegir un *control* creado por el propio usuario.

ActiveX – Para transferir datos y programas de unas aplicaciones a otras dentro de Windows.

Select a Control – Para seleccionar cualquier *control*.

Al seleccionar objetos desde el menú *Controls* estos aparecen sobre el *panel frontal*, pueden colocarse donde convenga, y además tienen su propio menú desplegable que permite la configuración de algunos parámetros específicos de cada tipo de *control*.

Paleta de funciones (functions palette)

Se emplea en el diseño del *diagrama de bloques*. La *paleta de funciones* contiene todos los objetos que se emplean en la implementación del programa del VI, ya sean *funciones* aritméticas, de entrada/salida de señales, entrada/salida de datos a fichero, adquisición de señales, temporización de la ejecución del programa,...

Para seleccionar una *función* o *estructura* concretas, se debe desplegar el menú *Functions* y elegir entre las opciones que aparecen. A continuación se enumeran todas ellas, junto con una pequeña definición.

Structures – Muestra las *estructuras* de control del programa, junto con las variables locales y globales.

Numeric – Muestra *funciones* aritméticas y constantes numéricas.

Boolean – Muestra *funciones* y constantes lógicas.

String – Muestra *funciones* para manipular cadenas de caracteres, así como constantes de caracteres.

Array – Contiene *funciones* útiles para procesar datos en forma de vectores, así como constantes de vectores.

Cluster – Contiene *funciones* útiles para procesar datos procedentes de gráficas y destinados a ser representados en ellas, así como las correspondientes constantes.

Comparison – Muestra *funciones* que sirven para comparar números, valores booleanos o cadenas de caracteres.

Time & Dialog – Contiene *funciones* para trabajar con cuadros de diálogo, introducir contadores y retardos, etc.

File I/O – Muestra *funciones* para operar con ficheros.

Communication – Muestra diversas *funciones* que sirven para comunicar varios ordenadores entre sí, o para permitir la comunicación entra distintos programas.

Instrument I/O – Muestra un submenú de *Vis*, que facilita la comunicación con instrumentos periféricos que siguen la norma ANSI/IEEE 488.2-1987, y el control del puerto serie.

Data Acquisition – Contiene a su vez un submenú donde puede elegirse entre distintas librerías referentes a la adquisición de datos.

Analysis – Contiene un submenú en el que se puede elegir entre una amplia gama de *funciones* matemáticas de análisis.

Tutorial – Incluye un menú de *Vis* que se utilizan en el manual LabVIEW Tutorial.

Advanced – Contiene diversos submenús que permiten el control de la ayuda, de los *Vis*, manipulación de datos, procesado de eventos, control de la memoria, empleo de programas ejecutables o incluidos en librerías DLL, etc.

Instrument drivers – En él se muestran los drivers disponibles de distintos instrumentos.

User Libraries – Muestra as librerías definidas por el usuario. En este caso, la librería mostrada contiene los drivers de la tarjeta de adquisición de datos de Advantech.

Application control – Contiene varias *funciones* que regulan el funcionamiento de la propia aplicación en ejecución.

Select a VI – Permite seleccionar cualquier *VI* para emplearlo como *subVI*.

PROGRAMACIÓN EN LABVIEW

Con el entorno gráfico de programación de LabVIEW se comienza a programar a partir del *panel frontal*.

En primer lugar se definirán y seleccionarán de la *paleta de controles* todos los *controles* (entradas que dará el usuario) e *indicadores* (salidas que presentará en pantalla el

VI) que se emplearán para introducir los datos por parte del usuario y presentar en pantalla los resultados.

Una vez colocados en la ventana correspondiente al *panel frontal* todos los objetos necesarios, debe pasarse a la ventana *Diagram* (menú *Windows > Show Diagram*), que es donde se realiza la programación propiamente dicha (*diagrama de bloques*). Al abrir esta ventana, en ella se encuentran los terminales correspondientes a los objetos situados en el *panel frontal*, dispuestos automáticamente por LabVIEW.

Se deben ir situando las *funciones*, *estructuras*, etc. que se requieran para el desarrollo del programa, las cuales se unen a los terminales mediante cables.

Para facilitar la tarea de conexión de todos los terminales, en el menú "*Help*" puede elegirse la opción "*Show Help*", con lo que al colocar el cursor del ratón sobre un elemento aparece una ventana con información relativa a éste (parámetros de entrada y salida).



Además, si se tiene seleccionado el cursor de cableado, al situar éste sobre un elemento se muestran los terminales de forma intermitente.



**DISEÑO DE APLICACIONES PARA LA
MEDICION, ANALISIS Y CONTROL DE
CALIDAD DE LOS PROCESOS DE
MANUFACTURA**

A. INTRODUCCION A LA PROGRAMACION EN .NET CON MODULOS DE CONTROL.

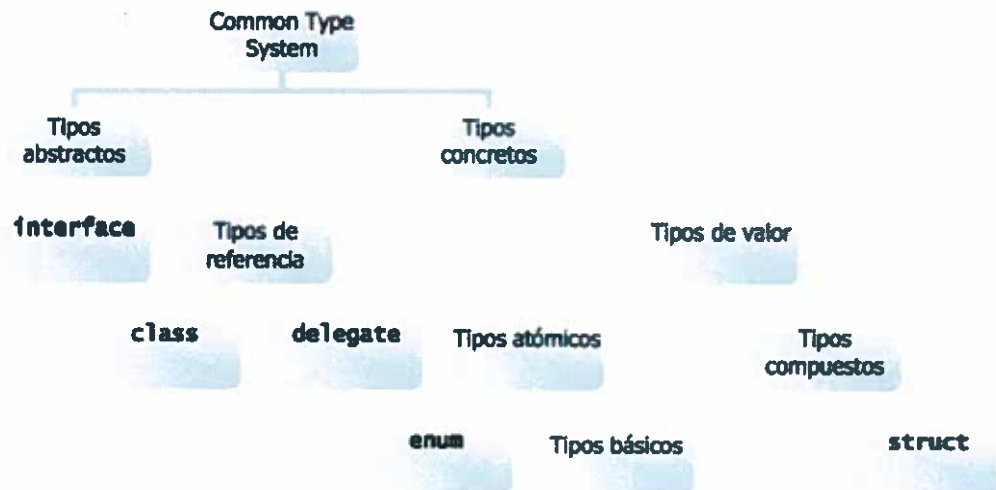
NET es la versión 3.0 de COM. Y no le falta razón, al menos desde el punto de vista histórico. Al parecer, las primeras ideas sobre .NET surgieron como parte del diseño de un sucesor para COM. ¿Qué es COM+? Una serie de protocolos que permiten la colaboración entre objetos desarrollados con diferentes lenguajes y herramientas, y que pueden residir en diferentes proveesos o incluso máquinas. ¿Qué es .NET, si la observamos a través de esta estrecha rendija? Es lo mismo pero mejor. Es por eso una ironía que incluso la versión 2.0 de la plataforma no haya logrado sustituir completamente la funcionalidad de COM+, y que deba colaborar con COM+ para poder aprovechar los llamados servicios corporativos, que estudiaremos en la última serie de este curso. NET es una herramienta de desarrollo.

1. Hay toda una familia de lenguajes que pueden utilizarse para desarrollar aplicaciones o componentes. En realidad, tras esa familia concreta de lenguajes, hay un conjunto muy bien definido de reglas del juego que deben ser satisfechas por cualquier aspirante a ingresar en tal selecto club.
2. Hay un sistema de soporte para la ejecución suficientemente complejo para que lo dividamos en subsistemas. Tenemos, para empezar, un sistema que carga las aplicaciones y las prepara para su ejecución. Tradicionalmente, esta ha sido una responsabilidad del sistema operativo y, efectivamente, Windows ha necesitado algunos retoques para satisfacer las exigencias de carga de las aplicaciones .NET.
3. Las aplicaciones .NET no sólo necesitan una tutela especial durante la carga, sino incluso durante la ejecución. Estas aplicaciones utilizan, por ejemplo, un recolector de basura para la gestión de memoria, y en el caso típico, es necesario traducir el código intermedio que generan los compiladores a código nativo ejecutable, a medida que la ejecución vaya activando una u otra ruta dentro de la aplicación. Hay tipos de datos cuya implementación se sintetiza en tiempo de ejecución: los tipos delgados, que se utilizan en la implementación de eventos y de funciones de respuesta, y los tipos genéricos en la versión. 2.0 de la plataforma.

4. Por último, están las numerosas bibliotecas de clases necesarias ejecutar hasta la más simple tarea.

EL SISTEMA DE TIPOS COMUNES

Para que todos los lenguajes soportados por .NET pueden comunicarse entre sí, deben compartir un mismo sistema de tipos y un mismo modelo de objetos. El siguiente diagrama muestra los distintos tipos de objetos definidos por el *Common Type System (CTS)*, que es el nombre del sistema de tipos de .NET:



Este es un sistema de tipos más rico y expresivos que el soportado por Java. Este último lenguaje sólo incluiría, de los tipos anteriores, esta lista reducida:

1. Clases
2. Tipos de Interfaz.
3. Tipos básicos o atómicos

Java intenta utilizar el concepto de clase para modelar absolutamente todo, lo cual es correcto desde el punto de vista conceptual... y un verdadero incordio desde el punto de vista práctico. En .NET también se considera que la clase es el concepto fundamental, pero

se toletan ciertas mutaciones del concepto para facilitar algunas técnicas de programación de uso frecuente.

TIPOS BASICOS

Antes de complicarnos demasiados la vida, veamos qué es un tipo “basico” y cuáles de ellos nos ofrece la platafomra. No es una definición sencilla, entre otras coasas porque todos los lenguajes intentan dismiular al máximo las posibles características especiales de estos tipos. Tomemos como ejemplos los enteros. Resulta que C# nos permite escribir expersiones como la siguiente:

```
999.ToString()
```

Claro, es una tontería escribir una expresión como la anterior... aunque no tanto, si la función *To String* incluye una cadena de formato:

```
999.ToString("C")
```

¿Está seguro de que se trata de una expresión constante? En España impime una cantidad en euros, claro... Lo que importa es que a medida que los lenguajes orientados a objetos progresan, es cada vez más difícil distinguir entre tipos básicos y objetos. Esto es bueno para el programador, porque así debe recordar menos excepciones a las reglas, pero en el fondo se trata de un disfraz. Los objetos, ya sean instancias de clases o de estructuras, son tipos *compuestos* que, obviando la recursividad, deben definirse con la ayuda de tipos más simples.

A pesar de todo lo dicho, hay una forma inequívoca para diagnosticar si un tipo determindado se puede considerar “básico”: basta con mirar la documentación del *Common Language Runtime*, y averi gurar si el tipo dado es reconocido como tal por la máquina virtual de la ejecución. Por ejemplo, la máquina reconoce sin mas los tipos enteros de 32 bits. Aunque no se puede trabajar directamente con enteros de 16 bits en la pila, existen operaciones para la conversión en las direcciones con los enteros de 32 bits. Hay otro indicio necesario, aunque no suficiente: todos los tipos básicos son *tipos de valor*, como veremos más adelante.

C# asigna una palabra clave para cada una de los tipos básicos. Además estos tipos tienen un nombre completo, como cualquier tipo de datos normal del *Common Type System*. Estos son los tipos básicos y sus nombres, tanto desde el punto de vista de C# como del *CTS*.

C#	CLR	Significado
Bool	<i>System.Boolean</i>	Verdadero y Falso ¿no es así?
Char	<i>System.Char</i>	Caracters Unicode (¡16 bits cada uno!
Sbyte	<i>System.Sbyte</i>	Enteros de 8 bits con signo.
Byte	<i>System.Byte</i>	Enteros de 8 bits sin signo.
Short	<i>System.Int16</i>	Enteros de 16 bits con signo.
Ushort	<i>System.UInt16</i>	Enteros de 16 bits sin signo.
Int	<i>System.Int32</i>	Enteros de 32 bits sin signo.
UInt	<i>System.UInt16</i>	Enteros de 32 bits con signo.
Long	<i>System.Int64</i>	Enteros de 64 bits con signo.
Ulong	<i>System.UInt64</i>	Enteros de 64 bits sin signo.
Float	<i>System.Single</i>	Valores reales representados en 4 bytes.
Double	<i>System.Double</i>	Valores reales representados en 8 bytes.

Hay autores, y entre ellos hay algunos famosos, que aconseja no usar los nombres de tipos abreviados de C#. Aducen que así se facilita la conversión de código entre lenguajes soportados por la plataforma. De modo que, de hacer caso al consejo, para declarar una variable local de tipo entero tendríamos que escribir lo siguiente:

```
// ...
System.Int32 i = 0;
// ...
```

Cuando declaramos un campo, lo normal es que se reserve memoria para éste en cada una de las instancias de la clase. Si le parece anormal la palabra “normal” para hablar de estos campos , Itambien se les llama campos de instancia. Podemos tambien usar el modificador **static** en la declaración de campos, propiedades, eventos, métodos e incluso constructores. Por ejemplo:

```
// Un campo de instancia  
public string nombre;  
  
// un campo “estatico”  
public static string prefijo;
```

Fundamentos del Entorno de LabVIEW—Examine los bloques fundamentales de construcción para cualquier aplicación de LabVIEW, incluyendo el panel frontal,

Bienvenido a LabVIEW. El propósito de este manual es ponerlo al día con los conceptos básicos de LabVIEW y la programación gráfica.

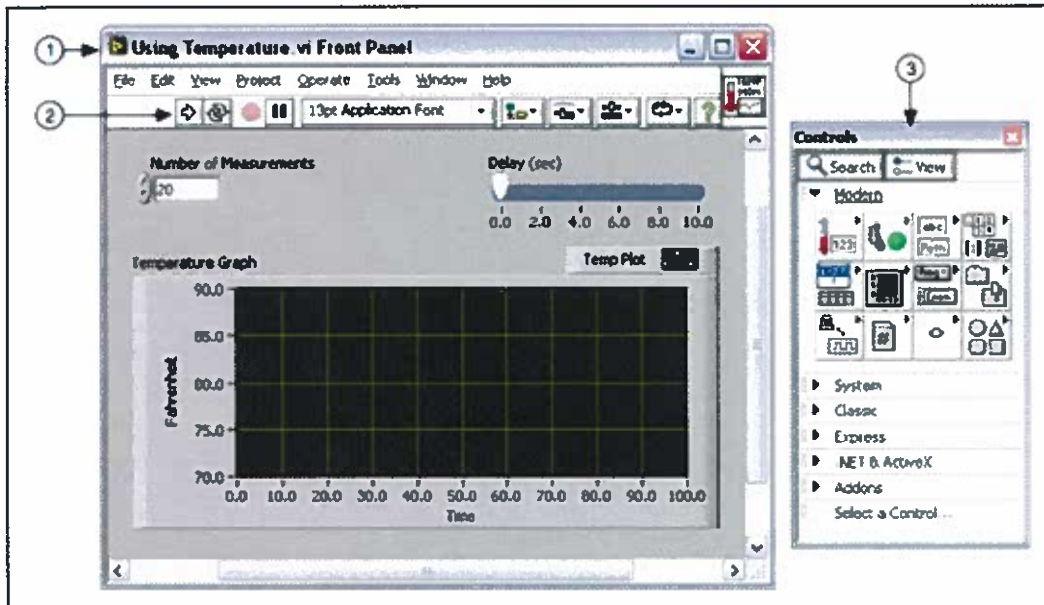
Los programas de LabVIEW son llamados instrumentos virtuales o VIs ya que su apariencia y operación generalmente imitan a los instrumentos físicos, como osciloscopios y multímetros. LabVIEW contiene una extensa variedad de herramientas para adquirir, analizar, visualizar y almacenar datos, así como herramientas para ayudarle a solucionar problemas en el código que escriba.

Cuando crea un nuevo VI, ve dos ventanas: la ventana del panel frontal y el diagrama de bloques.

Panel Frontal

Regresar al inicio

Cuando abre un VI nuevo o existente, aparece la ventana del panel frontal del VI. La ventana del panel frontal es la interfaz de usuario para el VI. La Figura 1 muestra un ejemplo de una ventana del panel frontal.



(1) Ventana de Panel Frontal | (2) Barra de Herramientas | (3) Paleta de Controles

Figura 1. Ejemplo de un Panel Frontal

Paleta de Controles

Regresar al inicio

La paleta de Controles contiene los controles e indicadores que utiliza para crear el panel frontal. Puede tener acceso a la paleta de Controles de la ventana del panel frontal al seleccionar View»Controls Palette o al dar clic con botón derecho en cualquier espacio en blanco en la ventana del panel frontal. La paleta de Controles está dividida en varias categorías; puede exponer algunas o todas estas categorías para cumplir con sus necesidades. La Figura 2 muestra la paleta de Controles con todas las categorías expuestas y la categoría Moderna expandida.

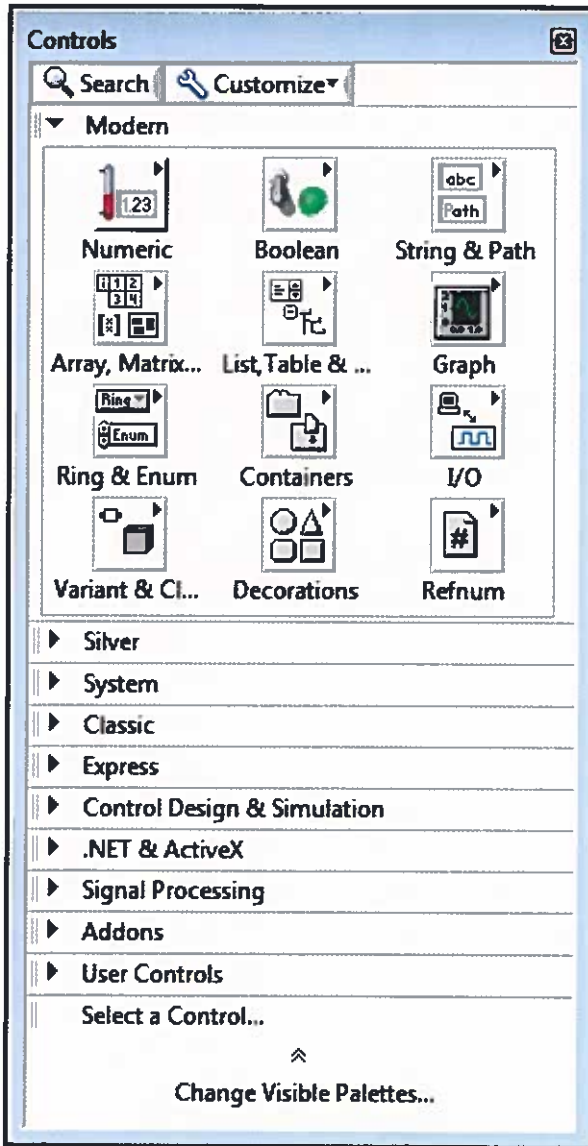


Figura 2. Paleta de Controles

Para ver o esconder categorías (subpaletas), haga clic en el botón Customize y seleccione Change Visible Palettes.

Controles e Indicadores

Regresar al inicio

Cada VI tiene un panel frontal que usted puede diseñar como una interfaz de usuario. Usted también puede usar paneles frontales como la manera de transmitir entradas y recibir salidas cuando se llama al VI desde otro diagrama de bloque. Usted crea la interfaz de usuario de un VI al colocar controles e indicadores en el panel frontal de un VI. Cuando interactúa con un panel frontal como una interfaz de usuario, puede modificar los controles para transmitir entradas y ver los resultados en los indicadores. Los controles que definen las entradas y los indicadores muestran las salidas.

Los controles generalmente son perillas, botones, perillas, deslizadores y secuencias. Simulan dispositivos de entrada de instrumentos y suministran datos al diagrama de bloques del VI. Los indicadores generalmente son gráficas, tablas, LEDs y secuencias de estado. Los indicadores simulan dispositivos de salida de instrumentos y muestran los datos que diagrama de bloques adquiere o genera.

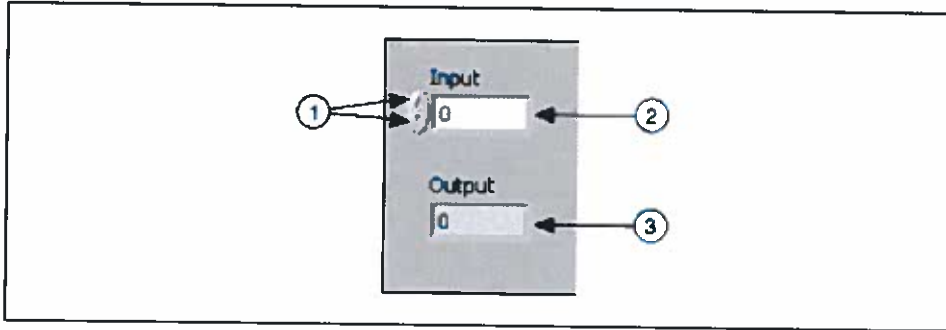
La Figura 1 tiene dos controles: Number of Measurements y Delay (sec). Tiene un indicador: una gráfica XY llamada Temperature Graph.

El usuario puede cambiar el valor de entrada para los controles Number of Measurements and Delay (sec). El usuario puede ver el valor generado por el VI en el indicador Temperature Graph. El VI genera los valores para los indicadores basados en el código creado en el diagrama de bloques.

Cada control o indicador tiene un tipo de datos asociado con él. Por ejemplo, el slide horizontal Delay (sec) es un tipo de datos numérico. Los tipos de datos utilizados con mayor frecuencia son numéricos, Booleano y cadena de caracteres.

Controles e Indicadores Numéricos

El tipo de datos numérico pueden representar números de varios tipos como un entero o real. Los dos objetos numéricos comunes son el control numérico y el indicador numérico, como se muestra en la Figura 3. Los objetos como medidores y perillas también representan datos numéricos.



(1) Botones de Incremento/Reducción | (2) Control Numérico | (3) Indicador Numérico

Figura 3. Controles e Indicadores Numéricos

Para proporcionar o cambiar valores en un control numérico, haga clic en el botón de incremento y reducción o haga doble clic en el número, introduzca un nuevo número y presione la tecla <Enter>.

Controles e Indicadores Booleano

El tipo de datos Booleano representa datos que solamente tienen dos estados posibles, como TRUE y FALSE u ON y OFF. Use los controles e indicadores Booleano para proporcionar y visualizar valores Booleano. Los objetos Booleano simulan interruptores, botones y LEDs. El interruptor de encendido vertical y los objetos LED redondos se muestran en la Figura 4.



Figura 4. Controles e Indicadores Booleano

Controles e Indicadores de Cadena de Caracteres

El tipo de datos cadena de caracteres es una secuencia de caracteres ASCII. Use controles en cadena para recibir texto desde el usuario como una contraseña o nombre de usuario. Use indicadores en cadena para mostrar texto al usuario. Los objetos en cadena más comunes son tablas y cuadros de texto, como se muestra en la Figura 5.

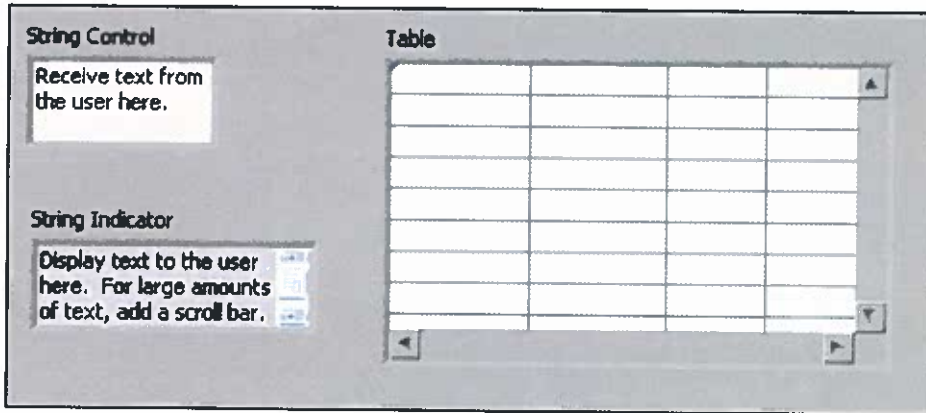
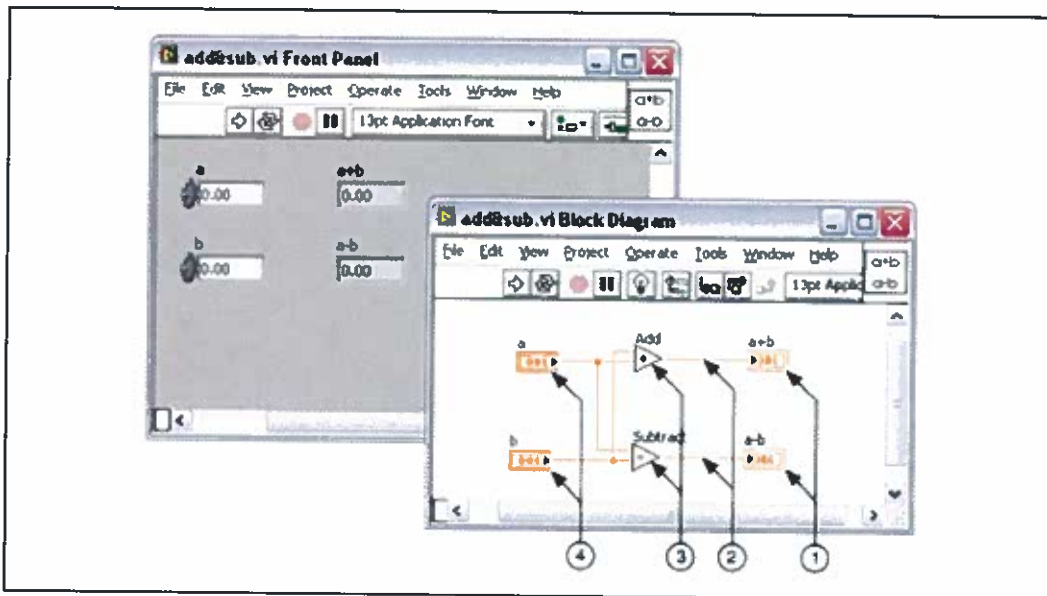


Figura 5. Controles e Indicadores de cadena de caracteres

Diagrama de Bloques

Regresar al inicio

Los objetos del diagrama de bloques incluyen terminales, subVIs, funciones, constantes, estructuras y cables, los cuales transfieren datos junto con otros objetos del diagrama de bloques.



(1) Terminales de Indicador | (2) Cables | (3) Nodos | (4) Terminales de Control

Figura 6. Ejemplo de un Diagrama de Bloques y Panel Frontal Correspondiente

Después de que crea la ventana del panel frontal, añade código usando representaciones gráficas de funciones para controlar los objetos del panel frontal. La ventana del diagrama de bloques contiene este código de fuente gráfica.

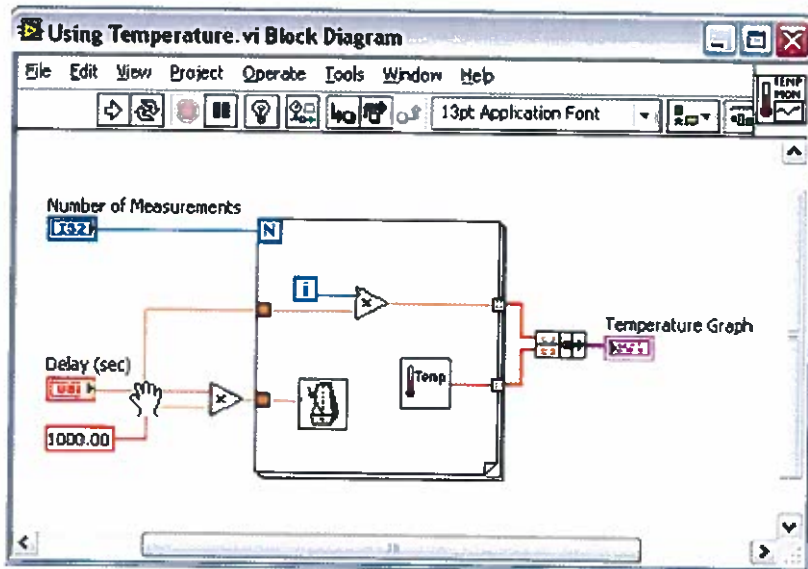


Figura 7. Diagrama de Bloques

Terminales

Regresar al inicio

Los objetos en la ventana del panel frontal aparecen como terminales en el diagrama de bloques. Las terminales son puertos de entrada y salida que intercambian información entre el panel frontal y diagrama de bloques. Son análogos a parámetros y constantes en lenguajes de programación basados en texto. Los tipos de terminales incluyen terminales de control o indicador y terminales de nodo. Las terminales de control e indicador pertenecen a los controles e indicadores del panel frontal. Los puntos de datos que usted proporciona en los controles del panel frontal (a y b en el panel frontal anterior) pasan al diagrama de bloques a través de las terminales de control. Entonces los puntos de datos ingresan las funciones de Suma y Resta. Cuando las funciones de Suma y Resta terminan sus cálculos, producen nuevos valores de datos. Los valores de datos van a las terminales de indicador, donde actualizan los indicadores del panel frontal (a+b y a-b en el panel frontal anterior).

Controles, Indicadores y Constantes

Regresar al inicio

Los controles, indicadores y constantes se comportan como entradas y salidas del algoritmo del diagrama de bloques. Considere la implementación del algoritmo para el área de un triángulo:

$$\text{Área} = .5 * \text{Base} * \text{Altura}$$

En este algoritmo, Base y Altura son entradas y Área es una salida, como se muestra en la Figura 8.

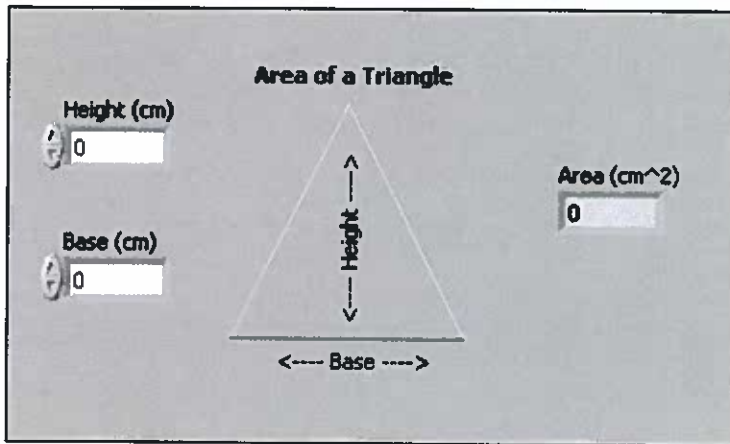
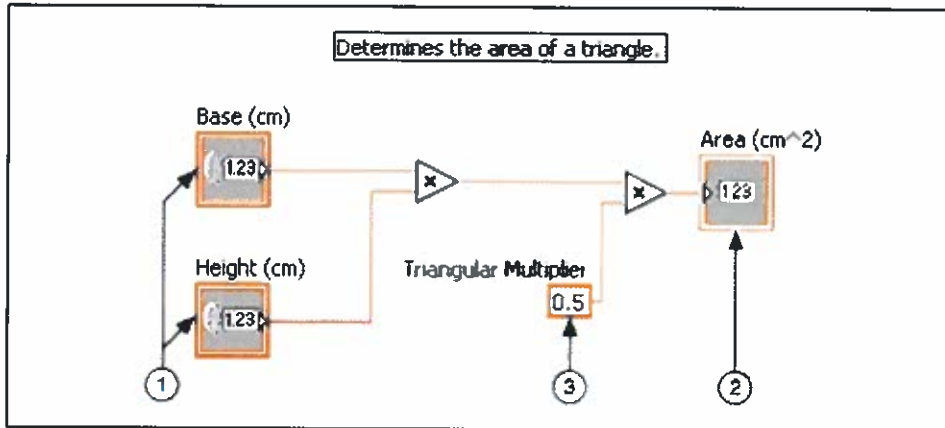


Figura 8. Panel Frontal del Área de un Triángulo

El usuario no cambiará o tendrá acceso a la constante 0.5, así que no aparecerá en el panel frontal a menos que se incluya como documentación del algoritmo.

La Figura 9 muestra una posible implementación de este algoritmo en un diagrama de bloques de LabVIEW. Este diagrama de bloques tiene cuatro terminales diferentes creadas por dos controles, una constante y un indicador.



(1) Controles | (2) Indicador | (3) Constante

Figura 9. Diagrama de Bloques del Área de un Triángulo con Vista de Terminal de Ícono

Note que las terminales del diagrama de bloques Base (cm) y Altura (cm) tienen una apariencia diferente de la terminal Área (cm²). Existen dos características distintivas entre un control y un indicador en el diagrama de bloques. La primera es una flecha en la terminal que indica la dirección del flujo de datos. Los controles tienen flechas que muestran los datos que salen de la terminal, mientras que el indicador tiene una flecha que muestra los datos que entran a la terminal. La segunda característica distintiva es el borde alrededor de la terminal. Los controles tienen un borde grueso y los indicadores tienen un borde delgado.

Puede ver terminales con o sin vista de íconos. La Figura 10 muestra el mismo diagrama de bloques sin usar la vista de ícono de las terminales; sin embargo existen las mismas características distintivas entre los controles y los indicadores.

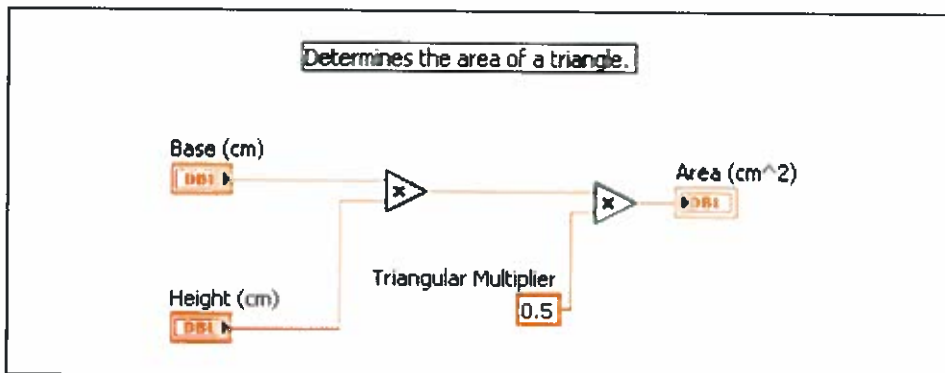


Figura 10. Diagrama de Bloques del Área de un Triángulo sin Vista de Terminal de Ícono

Nodos de Diagrama de Bloques

Regresar al inicio

Los nodos son objetos en el diagrama de bloques que tienen entradas y/o salidas y realizan operaciones cuando el VI se ejecuta. Son análogos a instrucciones, operaciones, funciones y sub rutinas en lenguajes de programación basados en texto. Los nodos pueden ser funciones, subVIs, Express VIs o estructuras. Las estructuras son elementos de control de procesos, como Estructuras de Casos, Ciclos For o Ciclos While.

Funciones

Las funciones son los elementos de operación fundamentales de LabVIEW. Las funciones de Suma y Resta en la Figura 6 son nodos de función. Las funciones no tienen ventanas del panel frontal o ventanas del diagrama de bloques pero no tienen paneles conectores. Al hacer doble clic en la función solamente selecciona la función. Una función tiene un fondo amarillo pálido en su ícono.

SubVIs

Después de desarrollar un VI, usted puede usarlo en otro VI. Un VI llamado desde el diagrama de bloques de otro VI es llamado un subVI. Usted puede reutilizar un subVI en otros VIs. Para crear un subVI, necesita desarrollar un panel conector y crear un ícono.

Un nodo de subVI corresponde a una llamada de subrutina en lenguajes de programación basados en texto. El nodo no es subVI, solamente una instrucción de llamada de subrutina en un programa que no es la propia subrutina. Un diagrama de bloques que contiene varios nodos de subVI idénticos llama al mismo subVI varias veces.

Los indicadores y controles de subVI reciben datos y regresan datos al diagrama de bloques del VI que hace el llamado. Cuando hace doble clic en un subVI en el diagrama de bloques, aparece la ventana del panel frontal. El panel frontal incluye controles e indicadores. El diagrama de bloques incluye cables, íconos, funciones, subVIs probables u otros objetos de LabVIEW.



Cada VI muestra un ícono en la esquina superior derecha de la ventana del panel frontal y la ventana del diagrama de bloques. Un ejemplo del ícono predeterminado se muestra aquí. Un ícono es una representación gráfica de un VI. El ícono puede contener texto e imágenes. Si usa un VI como un subVI, el ícono identifica al subVI en el diagrama de bloques del VI. El ícono predeterminado contiene un número que indica cuántos nuevos VI abrió después de iniciar LabVIEW.



Para usar un VI como un subVI, necesita construir un panel conector, como se muestra arriba. El panel conector es un conjunto de terminales en el ícono que corresponde a los controles e indicadores de ese VI, similares a la lista de parámetros de una función llamada en lenguajes de programación basados en texto. Obtenga acceso al panel conector al dar clic con botón derecho en el ícono en la parte superior derecha de la ventana del panel frontal. Usted puede tener acceso al panel conector desde el ícono en la ventana del diagrama de bloques. Un ícono de subVI tiene un fondo blanco en su ícono.

Express VIs

Los Express VIs son nodos que requieren cableado mínimo ya que usted los configura con ventanas de diálogo. Use Express VIs para tareas de medidas comunes. Consulte el tema de Express VIs de la Ayuda de LabVIEW para más información. Aparecen en el diagrama de bloques como nodos expandibles con iconos rodeados por un campo azul.

Paleta de Funciones

Regresar al inicio

La paleta de Funciones contiene los VIs, funciones y constantes que usted utiliza para crear el diagrama de bloques. Usted tiene acceso a la paleta de Funciones del diagrama de bloques al seleccionar View»Functions Palette. La paleta de Funciones está dividida en varias categorías; usted puede mostrar y esconder categorías para cumplir con sus necesidades. La Figura 11 muestra la paleta de Funciones con todas las categorías expuestas y la categoría de Programación expandida.

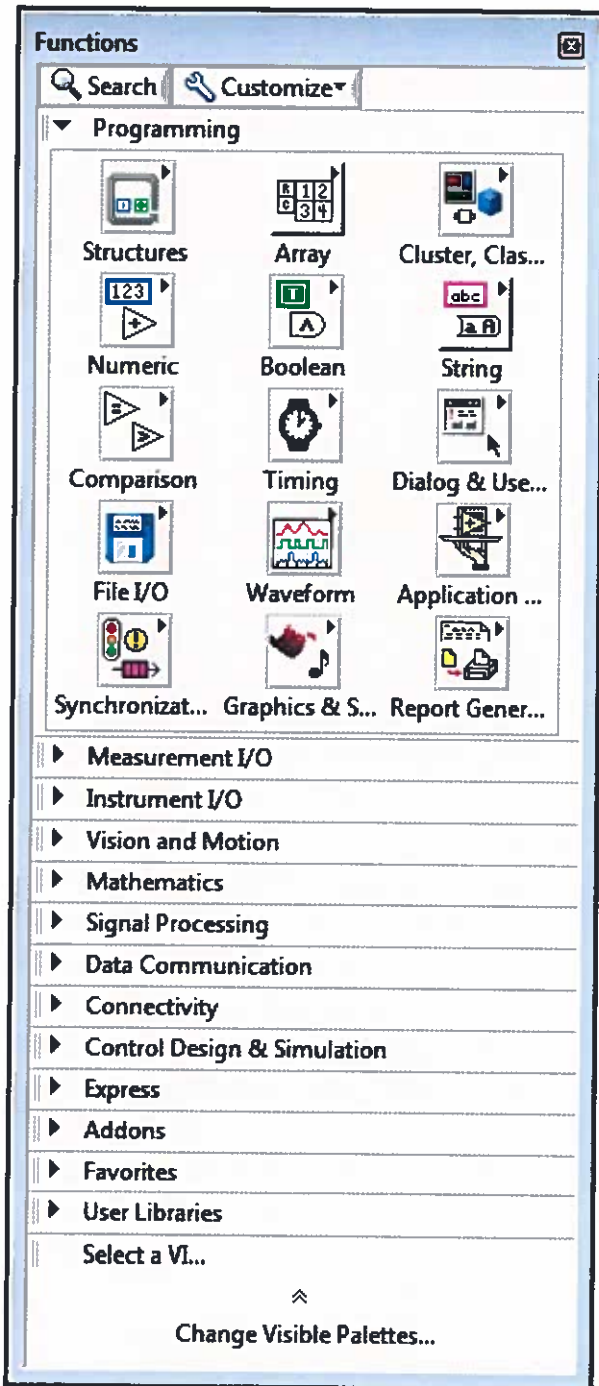


Figura 11. Paleta de Funciones

Para ver o esconder categorías, haga clic en el botón Customize y seleccione Change Visible Palettes.

Buscar Controles, VIs y Funciones

Regresar al inicio

Cuando usted selecciona View»Controls o View»Functions para abrir las paletas Controles y Funciones, aparecen dos botones en la parte superior de la paleta.



Search—Cambia la paleta para el modo buscar, así usted puede realizar búsquedas basadas en texto para ubicar los controles, VIs o funciones en las paletas. Mientras una paleta está en modo de búsqueda, haga clic en el botón Return para salir del modo de búsqueda y regresar a la paleta.



Customize—Proporciona opciones para seleccionar un formato para la paleta actual, mostrar y esconder categorías para todas las paletas y clasificar alfabéticamente elementos en los formatos de Texto y Árbol. Seleccione Options del menú de acceso directo para mostrar la página Controls/Functions Palettes de la ventana de diálogo Options, en la cual puede seleccionar un formato para todas las paletas. Este botón aparece solamente si hace clic en la tachuela en la esquina superior izquierda para prender la paleta.

Hasta que se familiarice con la ubicación de los VIs y funciones, busque la función o VI usando el botón Search. Por ejemplo, si desea encontrar la función Random Number, haga clic en el botón Search en la barra de herramientas de la paleta de Funciones y comience a escribir Random Number en la ventana de diálogo en la parte superior de la paleta. LabVIEW enlista todos los elementos parecidos que comienzan con o contienen el texto que escribió. Puede dar clic en alguno de los resultados de búsqueda y arrástrelo al diagrama de bloques, como se muestra en la Figura 12.

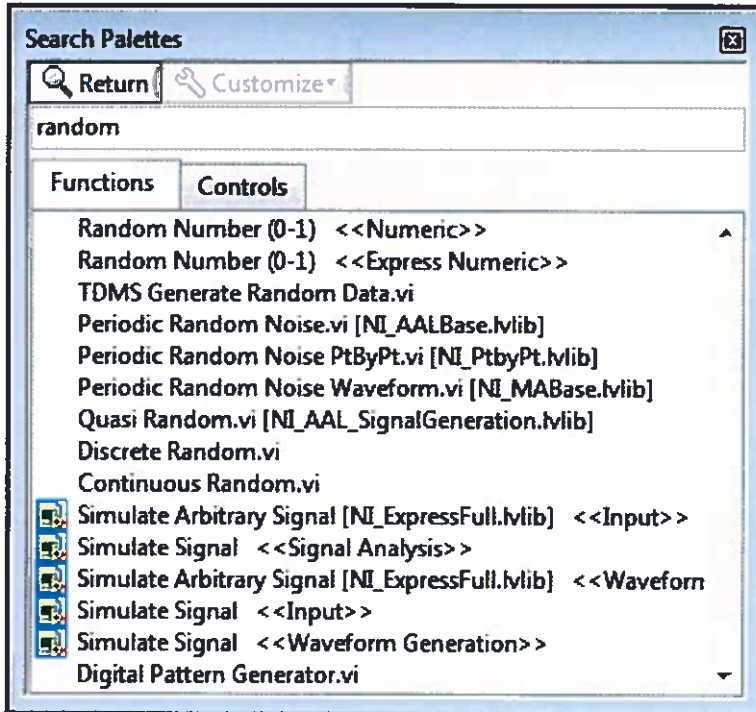


Figura 12. Buscar un Objeto en la Paleta de Funciones

diagrama de bloques, paletas, controles e indicadores.

Fundamentos de la Programación Gráfica—Vea cómo conectar funciones y trabajar con una variedad de tipos de datos al desarrollar aplicaciones.

Programación Gráfica

Programación de Flujo de Datos en LabVIEW

LabVIEW sigue un modelo de flujo de datos para ejecutar VIs. Un nodo de diagrama de bloques se ejecuta cuando recibe todas las entradas requeridas. Cuando el nodo se ejecuta, produce datos de salida y pasa los datos al siguiente nodo en la trayectoria del flujo de datos. El movimiento de datos a través de los nodos determina el orden de ejecución de los VIs y las funciones en el diagrama de bloques.

Visual Basic, C++, Java y la mayoría de otros lenguajes de programación basados en texto siguen un modelo de flujo de control para ejecución del programa. En el

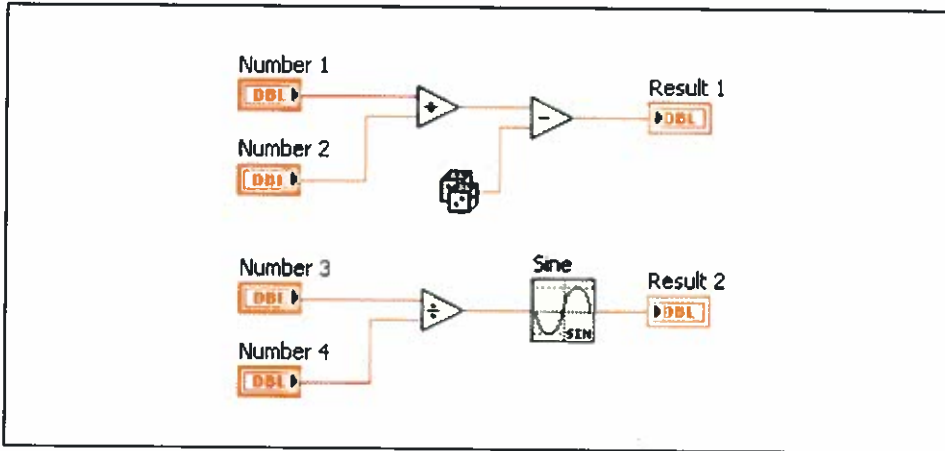


Figura 2. Ejemplo de Flujo de Datos para Múltiples Segmentos de Código

Cables







Regresar al inicio

Transfiere datos entre objetos del diagrama de bloques a través de cables. En las figuras 1 y 2, los cables conectan las terminales de control e indicador a la función de Suma y Resta. Cada cable tiene una sola fuente de datos, pero puede cablearlo a varios VIs o funciones que leen los datos. Los cables son de diferentes colores, estilos y grosores dependiendo de sus tipos de datos.



Un cable roto aparece como una línea negra punteada con una X roja a la mitad, como se muestra arriba. Los cables rotos ocurren por una variedad de razones, como cuando intenta cablear dos objetos con tipos de datos no compatibles.

La Tabla 1 muestra los cables más comunes.

Tipo de Cable	Escalar	Arreglo de 1D	Arreglo en 2D	Color
Numérico	 	 	 	Naranja (punto flotante),

flujo de control, el orden secuencial de los elementos del programa determina el orden de ejecución de un programa.

Para un ejemplo de programación de flujo de datos, considere un diagrama de bloques que suma dos números y después resta 50.00 del resultado de la suma, como se muestra en la Figura 1. En este caso, el diagrama de bloques se ejecuta de izquierda a derecha, no porque los objetos están colocados en ese orden, sino porque la función de Resta no puede ejecutarse hasta que la función de Suma termina ejecutarse y pasa los datos a la función de Resta. Recuerde que un nodo se ejecuta solamente cuando los datos están disponibles en todas sus terminales de entrada y proporciona los datos a las terminales de salida solamente cuando el nodo termina la ejecución.

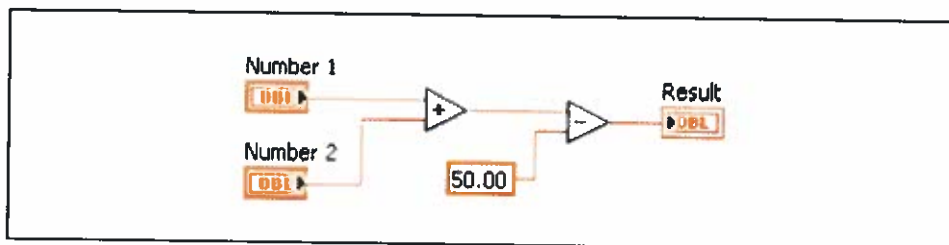


Figura 1. Ejemplo de Programación de Flujo de Datos

En la Figura 2, considere cuál segmento de código debe ejecutarse primero—la función de Suma, Número Aleatorio o División. No se puede saber porque las entradas a las funciones de Suma y División están disponibles al mismo tiempo y la función de Números Aleatorios no tiene entradas. En una situación donde un segmento de código debe ejecutarse antes que otro y no existe ninguna dependencia de datos entre las funciones, use otros métodos de programación como estructuras de secuencia o clusters de error para forzar el orden de la ejecución.

automáticamente los objetos que ya están en el diagrama de bloques. LabVIEW conecta las terminales que corresponden mejor y no conecta las terminales que no corresponden.

El cableado automático es habilitado de forma predeterminada cuando selecciona un objeto en la paleta Functionso cuando copia un objeto que ya está en el diagrama de bloques al presionar la tecla <Ctrl> y arrastrar el objeto. El cableado automático está deshabilitado de forma predeterminada cuando usa la herramienta de Ubicación para mover un objeto que ya está en el diagrama de bloques.

Puede ajustar las configuraciones del cableado automático al seleccionar Tools»Options y seleccionar Block Diagram en la lista Category.

Cablear Objetos Manualmente

Regresar al inicio

Cuando pasa la herramienta de Cableado sobre una terminal, aparece una etiqueta con el nombre de la terminal. Además, la terminal parpadea en la ventana Context Help y en el ícono para ayudarle a verificar que está cableando a la terminal correcta. Para cablear objetos juntos, pase la herramienta de Cableado sobre la primera terminal, haga clic y coloque el cursor sobre la segunda terminal y haga clic otra vez. Después de cablear, puede dar clic con botón derecho en el cable y seleccione Clean Up Wire en el menú corto para hacer que LabVIEW escoja automáticamente una trayectoria para el cable. Si tiene que eliminar cables rotos, presione <Ctrl-B> para eliminar todos los cables rotos en el diagrama de bloque

A. DISEÑO DE CONTROLES UI PARA INGENIERIA

Construyendo la base de UI

Crea tres layouts, cada uno con su propia hoja de eventos. Los llamaremos "Menú Principal", "UI", y el "Nivel 1".

UI, cambia el nombre de default layer a "UI" y crea una nueva capa (layer) debajo de ella llamada "Juego". Añade cualquier objeto que desees a la capa UI, tales como objetos de texto llamados SaludJugador, VidasJugador y RecordActual.







Tipo de Cable	Escalar	Arreglo de 1D	Arreglo en 2D	Color
				Azul (entero)
Booleano				Verde
Cadena de caracteres				Rosa

Tabla 1. Tipos Comunes de Cable

Video: Tipos de Datos en LabVIEW

En LabVIEW, puede usar cables para conectar múltiples terminales para pasar datos en un VI. Debe conectar los cables a las entradas y salidas que son compatibles con los datos que son transferidos con el cable. Por ejemplo, no puede cablear una salida de tipo arreglo a una entrada numérica. Además la dirección de los cables debe ser correcta. Debe conectar los cables solamente a una entrada y por lo menos a una salida. Por ejemplo, no puede cablear dos indicadores juntos. Los componentes que determinan la compatibilidad del cableado incluyen los tipos de datos del control y/o el indicador y los tipos de datos de la terminal. Por ejemplo, si un interruptor tiene un borde verde, puede cablear un interruptor a cualquier entrada con una etiqueta verde en un VI Express. Si una perilla tiene un borde naranja, puede cablear una perilla a cualquier entrada con una etiqueta naranja. Sin embargo, no puede cablear una perilla naranja a una entrada con una etiqueta verde. Note que los cables son del mismo color que la terminal.

Cablear Objetos Automáticamente

Regresar al inicio

Conforme acerca un objeto seleccionado a otros objetos en el diagrama de bloques, LabVIEW dibuja cables temporales para mostrarle conexiones válidas. Cuando suelta el botón del mouse para colocar el objeto en el diagrama de bloques, LabVIEW conecta los cables automáticamente. También puede cablear

Colócalos cómo quieras. Ahora, ve a cada objeto de UI que creaste y, en el panel de propiedades, establece cada objeto como "Global: yes".

Establecer la propiedad Parallax a 0,0.

- Establecer las capas a "UI" y establece la propiedad Parallax de la capa "UI" a 0,0.
- Los eventos
- Desde tu Menú Principal, crea un evento [Mouse: On Left button Clicked on NewGame -> System: Go to "UI"].
- A continuación, en UI, crea un evento [System: On start of layout -> System:
-

1. DISEÑO DE GRAFICAS DE ONDA Y DIAGRAMAS DE DISPERSION.

- Diagrama de Dispersión

- El diagrama de dispersión es una herramienta gráfica que ayuda a identificar la posible relación entre dos variables. Representa la relación entre dos variables de forma gráfica, lo que hace más fácil visualizar e interpretar los datos.

- El coeficiente de correlación lineal

- Ejemplo de Diagrama de Dispersión

- diagrama de dispersión

- El cálculo del coeficiente de correlación lineal

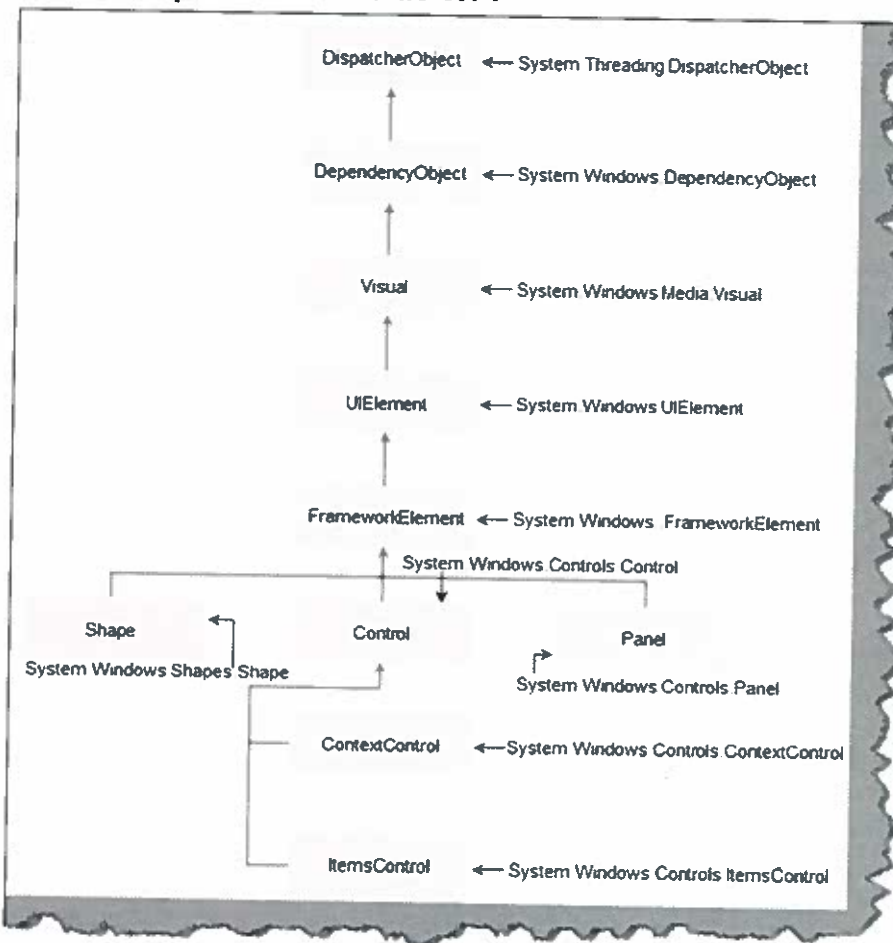
CONECTIVIDAD DE RED E INTEGRACION DE DATOS.

CONTROLES WPF

Arquitectura básica de WPF

Layout y Presentación de la Información en WPF (Contenedores Primera Parte)

- WPF por defecto solo admite la colocación de un solo elemento en una ventana y esta debe de ser un contenedor, cuando creamos un usercontrol o page o window su hijo debe de ser un contenedor. y en este iremos añadiendo controles
- En los contenedores de WPF tiene lugar en dos estados: Measure (Medición) y Arrange (distribución).
- **Jerarquía de Clases de WPF**



MEASURE

- En estado de medición (Measure) el contenedor comprueba por cada uno de los elementos hijos su tamaño deseado, es decir, les “pregunta” que tamaño pretende ocupar en el contenedor.

ARRANGE

- En este estado de distribución (Arrange), el contenedor coloca a cada uno de los elementos hijos en su posición apropiada.
- Veremos que entender estos estados son importantes cuando desarrollemos nuestros propios contenedores

Clase Panel

- La clase abstracta `System.Windows.Controls.Panel` es de la que derivan todos los contenedores de WPF. La clase `Panel` expone un pequeño grupo de miembros incluyendo las siguientes propiedades públicas:

BACKGROUND

- Utilizando el `Brush` dado pinta el fondo del panel. Como nota importante recalcar que debemos asignar un valor no nulo si queremos recibir eventos

de
ratón.

Podemos utilizar como color de fondo el transparente y así disponemos de los eventos del ratón.

CHILDREN

- Importante colección donde se almacenan los elementos alojados en el panel.

Nos referimos al primer nivel de elementos hijos, por lo que es posible que existan más subniveles.

SYSTEM.THREADING.DISPATCHEROBJECT

- Todos los objetos de WPF derivan de DispatcherObject. WPF trabaja en STA, y el principal objetivo es manejar la concurrencia y threading. Cuando cualquier mensaje como por ejemplo un click del ratón son enviados al DispatcherObject que verifica si el código se está ejecutando en el thread correcto

SYSTEM.WINDOWS.DEPENDENCYOBJECT

- La clase base para todos los objetos que tienen las propiedades de dependencia, las dependency properties únicamente se usan para definir las propiedades de los controles de interfaz de usuario y de los objetos de negocio que interactúan con la UI

SYSTEM.WINDOWS.MEDIA.VISUAL

- Esta clase se encarga del renderizado de la UI, del clipping, calculo del recuadro de dibujado, transformaciones y coordenadas. Es una clase muy importante en WPF, ya que cualquier clase que se tenga que renderizar debe heredar de ella.

SYSTEM.WINDOWS.FRAMEWORKELEMENT

- Es una clase fundamental dentro de WPF , implementa miembros de la clase UIElement y define los estados de Measure y Arrange, ademas se encarga del enlace a datos, estilos, animaciones, define los eventos del ciclo de vida del objeto.

B. HERRAMIENTAS DE CONEXION DE HARDWARE CONTROLADORES NATIVOS EN .NET PARA CONEXION CON HARDWARE PXI Y DE TECNOLOGIA NATIONAL INSTRUMENTS.

Ya que la comunicación de hardware generalmente es el componente más desafiante en cualquier aplicación de medidas, los APIs de controladores .NET de National Instruments proporcionan bibliotecas de clases de hardware orientadas a objetos e interfaces para instrumentos GPIB, VISA, DAQ, PCI, USB y PXI.

Mientras estas interfaces simplifican los procesos de adquisición de datos y conectividad con instrumentos al abstraer los detalles del controlador de bajo nivel, Measurement Studio extiende estas habilidades con características de procesamientos de datos y visualización para ayudarle a construir un sistema completo.

Asistentes de Hardware de Measurement Studio

Measurement Studio ofrece las siguientes herramientas de productividad para ayudarle a acelerar el desarrollo y crear aplicaciones de adquisición de datos más potentes.

2. EL ASISTENTE DE PARAMETROS NI-DAQ MX

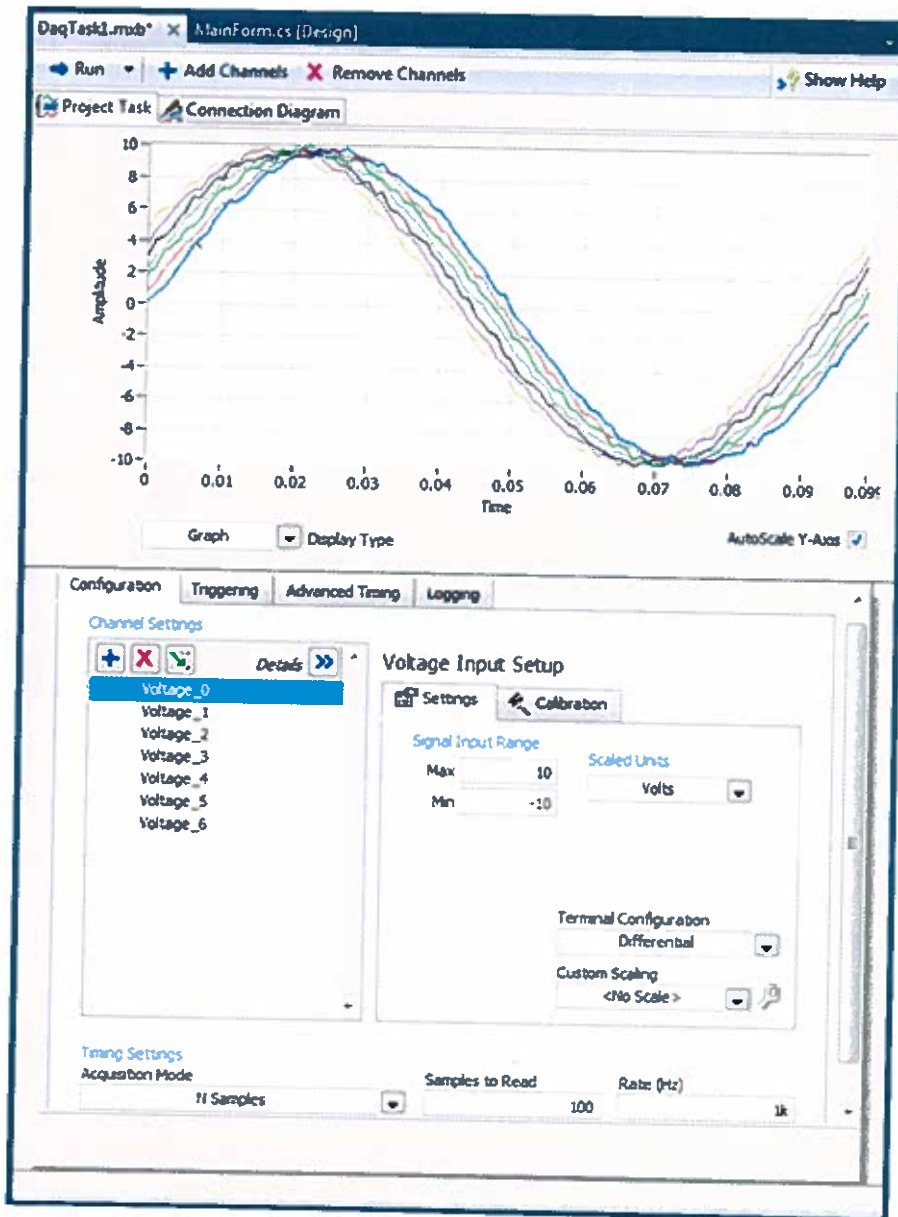
Asistentes de Hardware de Measurement Studio

Measurement Studio ofrece las siguientes herramientas de productividad para ayudarle a acelerar el desarrollo y crear aplicaciones de adquisición de datos más potentes.

DAQ Assistant

El DAQ Assistant está completamente integrado al entorno Visual Studio .NET. Úselo para configurar, probar y generar código fuente de Visual Basic .NET o Visual C# sin escribir una sola línea de código.

El DAQ Assistant ofrece una interfaz de usuario para configurar de manera interactiva tareas, canales y escalas de medidas. Solamente especifique cual dispositivo NI DAQ desea conectar, establezca el número de canales de entrada y salida, configure la tasa de adquisición y cantidad de muestreos y vea la salida inmediatamente antes de generar el código listo para ejecutar.



El Asistente de Parámetros para NI-DAQmx

- Con este asistente, usted puede descubrir e insertar valores de parámetros válidos en su código para métodos de NI-DAQmx. El Asistente de Parámetros muestra los dispositivos, canales y escalas de NI-DAQmx

actuales en el sistema, así como las tareas, terminales, disparos, interruptores y canales virtuales de NI-DAQmx actuales.

C. TDM(TECHNICAL DATA MANAGMENT)

1. INTERFASE TDMS

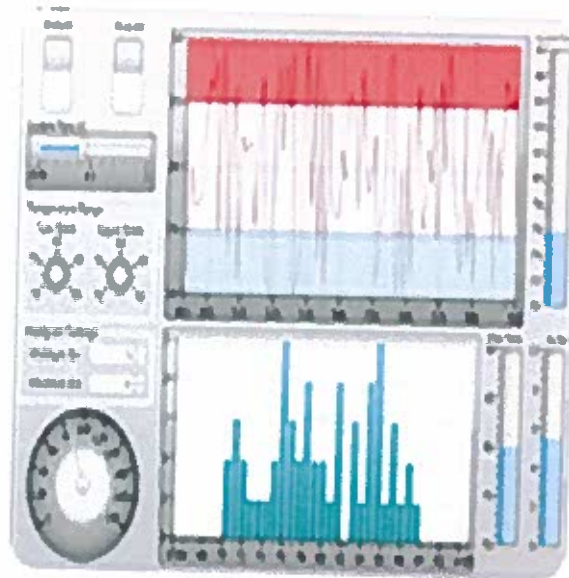
2. API PARA TDMS

Asistente de Controladores de Instrumentos IVI

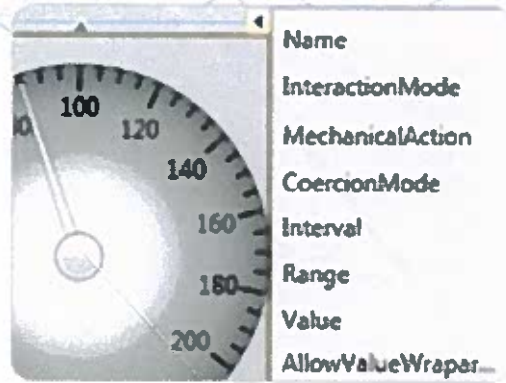
- La mayoría de los controladores de instrumentos no funcionan con .NET sin reescribir la mayoría del código fuente. Con Measurement Studio, usted puede usar el Asistente de Controlador de Instrumentos .NET para envolver inmediatamente controladores de instrumentos IVI, VXI plug-and-play y de legado en el código fuente de Visual Basic .NET o Visual C#.

¿Qué es Measurement Studio?

- Measurement Studio es el único paquete de herramientas diseñadas para desarrollar aplicaciones profesionales de ingeniería en Microsoft Visual Studio 2010, 2012 y 2013 para adquirir, analizar y visualizar datos de medidas. Ofrece clases de medidas extensibles y controles de interfaz de usuario (UI) para Windows Presentation Foundation (WPF), Windows y Formas Web para ayudarle a diseñar un sistema completo de medidas o pruebas en Visual Basic .NET y Visual C#.
- Controles UI para Ingeniería:
 - Presente Datos en un Formato Intuitivo e Impactante
 - Use gráficas, tablas, indicadores de aguja, perillas, medidores y LEDs que están diseñados para aplicaciones de ingeniería y son compatibles con tipos de datos adquiridos por hardware.
 - Cree Fácilmente UIs Profesionales
 - Escoja entre un juego completo de controles WPF para crear una experiencia de usuario sofisticada y resolución independiente.
- Extensibilidad y Personalización
 - Amplíe las Clases, Controles e Interfaces de Hardware



- Use clases basadas en los componentes de Measurement Studio como una base y después arrástrelas y extiéndalas para crear clases personalizadas.
- Personalice la Experiencia del Usuario
- Configure rápidamente propiedades de control con etiquetas inteligentes en tiempo de diseño y enlace propiedades o valores de control a una fuente de datos .NET.



D. CONECTIVIDAD EN RED E INTEGRACION DE DATOS. PRACTICA.

1. USO DE MEASUREMENT STUDIO PARA COMPARTIR DATOS

2. ESPECIFICACION DE FUENTE DE DATOS Y DISPOSITIVOS A TRAVES DE ESQUEMAS URL

3. ENLACE A CONTROLES MEDIANTE SISTEMAS WEB Y OTROS MEDIOS ACTUALES (APPS)

ESTE MANUAL ES PARA USO EXCLUSIVO DE LOS PARTICIPANTES EN ESTE CURSO. Y ES PROPIEDAD DE LA EMPRESA TECAPSA SA DE CV. POR LO QUE SE PROHIBE LA SE PROHÍBE LA REPRODUCCIÓN, TOTAL O PARCIAL, DE ESTE DOCUMENTO SIN LA AUTORIZACIÓN PREVIA DE TECAPSA SA DE CV

TECAPSA SA DE CV

DIPLOMA DE PARTICIPACION

**LIC. MONICA LOZANO
RODRIGUEZ**

EN EL PROGRAMA DE

CAPACITACION EN AUTOMATIZACION Y CONTROL
PARA MEJORA DE PROCESOS INDUSTRIALES Y MANUFACTURA
DE EQUIPOS DE COMPUTO

Otorgado el: 23 DE JUNIO DEL 2016



MTI ROSA MARIA CANTON CRODA

TECAPSA SA DE CV

DIPLOMA DE PARTICIPACION

LIC. PERLA ELIZABETH^a
PARADA MORADO

EN EL PROGRAMA DE

CAPACITACION EN AUTOMATIZACION Y CONTROL
PARA MEJORA DE PROCESOS INDUSTRIALES Y MANUFACTURA
DE EQUIPOS DE COMPUTO

Otorgado el: 23 DE JUNIO DEL 2016



MTT ROSA MARIA CANTON CRODA

TECAPSA SA DE CV

DIPLOMA DE PARTICIPACION

LIC. RICARDO^a ORONA
RUBIO

EN EL PROGRAMA DE

CAPACITACION EN AUTOMATIZACION Y CONTROL
PARA MEJORA DE PROCESOS INDUSTRIALES Y MANUFACTURA
DE EQUIPOS DE COMPUTO

Otorgado el 23 DE JUNIO DEL 2016



MTI ROSA MARIA CANTON CRODA

TECAPSA SA DE CV

DIPLOMA DE PARTICIPACION

LIC. LILIA ANGELICA
MUÑOZ GUERECA

EN EL PROGRAMA DE

CAPACITACION EN AUTOMATIZACION Y CONTROL
PARA MEJORA DE PROCESOS INDUSTRIALES Y MANUFACTURA
DE EQUIPOS DE COMPUTO

Otorgado el: 23-DE JUNIO DEL 2016



MTI ROSA MARIA CANTON CRODA

TECAPSA SA DE CV

DIPLOMA DE PARTICIPACION

LIC. VICTOR³ ANTONIO
CHAVEZ FLORES

EN EL PROGRAMA DE

CAPACITACION EN AUTOMATIZACION Y CONTROL
PARA MEJORA DE PROCESOS INDUSTRIALES Y MANUFACTURA
DE EQUIPOS DE COMPUTO

Otorgado el: 23^{DE} JUNIO DEL 2016



MTI ROSA MARIA CANTON CRODA

TECAPSA SA DE CV

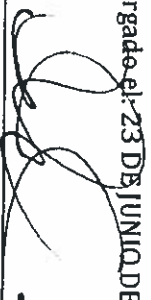
DIPLOMA DE PARTICIPACION

**LIC. DANIEL³ ENRIQUE
MAYNEZ TORRES**

EN EL PROGRAMA DE

CAPACITACION EN AUTOMATIZACION Y CONTROL
PARA MEJORA DE PROCESOS INDUSTRIALES Y MANUFACTURA
DE EQUIPOS DE COMPUTO

Otorgado el 23 DE JUNIO DEL 2016



MTI ROSA MARIA CANTON CRODA

TECAPSA SA DE CV

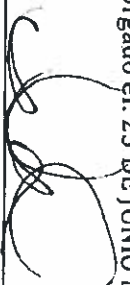
DIPLOMA DE PARTICIPACION

LIC. ULISES³ OLVERA
MEJÍA

EN EL PROGRAMA DE

CAPACITACION EN AUTOMATIZACION Y CONTROL
PARA MEJORA DE PROCESOS INDUSTRIALES Y MANUFACTURA
DE EQUIPOS DE COMPUTO

Otorgado el: 23 DE JUNIO DEL 2016



MTI ROSA MARIA CANTON CRODA

TECAPSA SA DE CV

DIPLOMA DE PARTICIPACION

ING. MARIO DANIEL
PEREZ ESPINOZA

EN EL PROGRAMA DE

CAPACITACION EN AUTOMATIZACION Y CONTROL
PARA MEJORA DE PROCESOS INDUSTRIALES Y MANUFACTURA
DE EQUIPOS DE COMPUTO

Otorgado el: 23 DE JUNIO DEL 2016



MTI ROSA MARIA CANTON CRODA

TECAPSA SA DE CV

DIPLOMA DE PARTICIPACION

LIC. ARTURO³ ARMANDO
FLORES CORDERO

EN EL PROGRAMA DE

CAPACITACION EN AUTOMATIZACION Y CONTROL
PARA MEJORA DE PROCESOS INDUSTRIALES Y MANUFACTURA
DE EQUIPOS DE COMPUTO

Otorgado el: 23 DE JUNIO DEL 2016



MTI ROSA MARIA CANTON CRODA

TECAPSA SA DE CV

DIPLOMA DE PARTICIPACION

ING. JOSE LUIS MURGUIA
ROJAS

EN EL PROGRAMA DE

CAPACITACION EN AUTOMATIZACION Y CONTROL
PARA MEJORA DE PROCESOS INDUSTRIALES Y MANUFACTURA
DE EQUIPOS DE COMPUTO

Otorgado el: 23 DE JUNIO DEL 2016



Nombre y puesto del presentador

TECAPSA SA DE CV

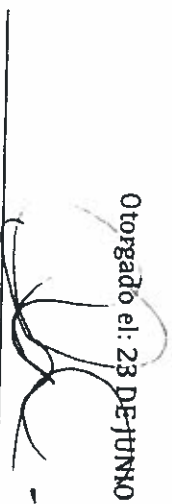
DIPLOMA DE PARTICIPACION

ING. RAUL AUGUSTO
AVITIA PALACIOS

EN EL PROGRAMA DE

CAPACITACION EN AUTOMATIZACION Y CONTROL
PARA MEJORA DE PROCESOS INDUSTRIALES Y MANUFACTURA
DE EQUIPOS DE COMPUTO

Otorgado el: 23 DE JUNIO DEL 2016



MTI ROSA MARIA CANTON CRODA

TECAPSA SA DE CV


DIPLOMA DE PARTICIPACION

ING. FELIPE DE JESUS
AGÜERO NAVA

EN EL PROGRAMA DE

CAPACITACION EN AUTOMATIZACION Y CONTROL
PARA MEJORA DE PROCESOS INDUSTRIALES Y MANUFACTURA
DE EQUIPOS DE COMPUTO

Otorgado el: 23 DE JUNIO DEL 2016



MTI ROSA MARIA CANTON CRODA

TECAPSA SA DE CV

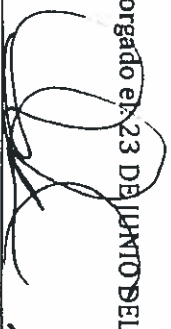
DIPLOMA DE PARTICIPACION

ING. EDGAR IVAN
AGÜERO NAVA

EN EL PROGRAMA DE

CAPACITACION EN AUTOMATIZACION Y CONTROL
PARA MEJORA DE PROCESOS INDUSTRIALES Y MANUFACTURA
DE EQUIPOS DE COMPUTO

Otorgado el 23 DE JUNIO DEL 2016



MTI ROSA MARIA CANTON CRODA

TECAPSA SA DE CV

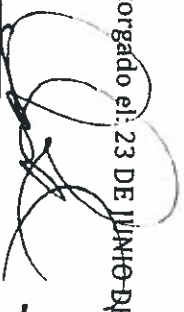
DIPLOMA DE PARTICIPACION

ING. JAIR³ OLVERA
CONCHAS

EN EL PROGRAMA DE

CAPACITACION EN AUTOMATIZACION Y CONTROL
PARA MEJORA DE PROCESOS INDUSTRIALES Y MANUFACTURA
DE EQUIPOS DE COMPUTO

Otorgado el: 23 DE JUNIO DEL 2016



MTI ROSA MARIA CANTON CRODA

TECAPSA SA DE CV

DIPLOMA DE PARTICIPACION

LIC. HILDA TATIANA
AGUILERA NUÑEZ

EN EL PROGRAMA DE

CAPACITACION EN AUTOMATIZACION Y CONTROL
PARA MEJORA DE PROCESOS INDUSTRIALES Y MANUFACTURA
DE EQUIPOS DE COMPUTO

Otorgado el: 23 DE JUNIO DEL 2016



MTI ROSA MARIA CANTON CRODA

TECAPSA SA DE CV

DIPLOMA DE PARTICIPACION

ING. JOSÉ LUIS
OYARZABAL VARELA

EN EL PROGRAMA DE

CAPACITACION EN AUTOMATIZACION Y CONTROL
PARA MEJORA DE PROCESOS INDUSTRIALES Y MANUFACTURA
DE EQUIPOS DE COMPUTO

OTORGADO EL 23 DE JUNIO DEL 2016



MTI ROSA MARIA CANTON CRODA

TECAPSA SA DE CV

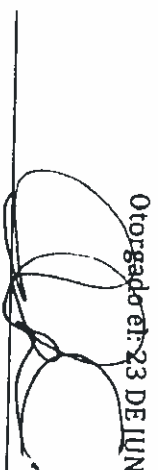
DIPLOMA DE PARTICIPACION

LIC. DULCE MARÍA
AGUILERA NUÑEZ

EN EL PROGRAMA DE

CAPACITACION EN AUTOMATIZACION Y CONTROL
PARA MEJORA DE PROCESOS INDUSTRIALES Y MANUFACTURA
DE EQUIPOS DE COMPUTO

Otorgado el 23 DE JUNIO DEL 2016



MTI ROSA MARIA CANTON CRODA

TECAPSA SA DE CV

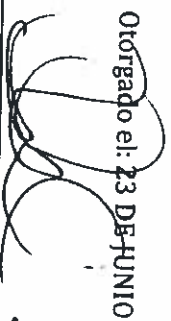
DIPLOMA DE PARTICIPACION

**LIC. SANDRA JULIANA
MAZCORRO CHAVEZ**

EN EL PROGRAMA DE

CAPACITACION EN AUTOMATIZACION Y CONTROL
PARA MEJORA DE PROCESOS INDUSTRIALES Y MANUFACTURA
DE EQUIPOS DE COMPUTO

Otorgado el: 23 DE JUNIO DEL 2016



MTI ROSA MARIA CANTON CRODA

TECAPSA SA DE CV

DIPLOMA DE PARTICIPACION

ING. JESSICA SUAREZ
OCHOA

EN EL PROGRAMA DE

CAPACITACION EN AUTOMATIZACION Y CONTROL
PARA MEJORA DE PROCESOS INDUSTRIALES Y MANUFACTURA
DE EQUIPOS DE COMPUTO

Otorgado el: 23 DE JUNIO DEL 2016



MTI ROSA MARIA CANTON CRODA

TECAPSA SA DE CV

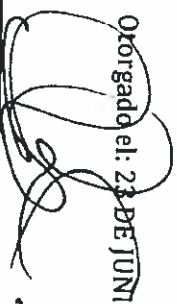
DIPLOMA DE PARTICIPACION

LIC. JESUS^A ANSELMO
CASTORENA CHAVEZ

EN EL PROGRAMA DE

CAPACITACION EN AUTOMATIZACION Y CONTROL
PARA MEJORA DE PROCESOS INDUSTRIALES Y MANUFACTURA
DE EQUIPOS DE COMPUTO

Otorgado el: 23 DE JUNIO DEL 2016



MTI ROSA MARIA CANTON CRODA

TECAPSA SA DE CV

DIPLOMA DE PARTICIPACION

ING. JORGE BURCIAGA
ZAMARRON

EN EL PROGRAMA DE

CAPACITACION EN AUTOMATIZACION Y CONTROL
PARA MEJORA DE PROCESOS INDUSTRIALES Y MANUFACTURA
DE EQUIPOS DE COMPUTO

Otorgado el 23 DE JUNIO DEL 2016



MTI ROSA MARIA CANTON CRODA

TECAPSA SA DE CV

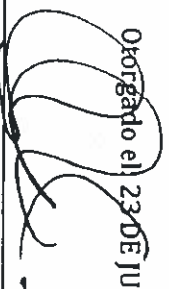
DIPLOMA DE PARTICIPACION

LIC. MIGUÉL ANGEL
FLORES REYES

EN EL PROGRAMA DE

CAPACITACION EN AUTOMATIZACION Y CONTROL
PARA MEJORA DE PROCESOS INDUSTRIALES Y MANUFACTURA
DE EQUIPOS DE COMPUTO

Orogado el 23 DE JUNIO DEL 2016



MTI ROSA MARIA CANTON CRODA

TECAPSA SA DE CV

DIPLOMA DE PARTICIPACION

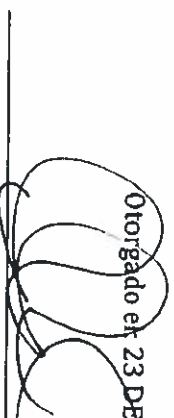
LIC. DAVID³ ISRAEL

RODRIGUEZ CISNEROS

EN EL PROGRAMA DE

CAPACITACION EN AUTOMATIZACION Y CONTROL
PARA MEJORA DE PROCESOS INDUSTRIALES Y MANUFACTURA
DE EQUIPOS DE COMPUTO

Otorigado el 23 DE JUNIO DEL 2016



MTI ROSA MARIA CANTON CRODA

TECAPSA SA DE CV

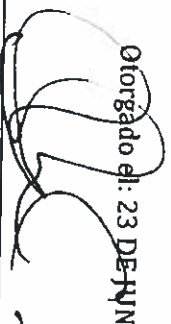
DIPLOMA DE PARTICIPACION

ING. JOSE DE JESUS
FLORES SEAÑEZ

EN EL PROGRAMA DE

CAPACITACION EN AUTOMATIZACION Y CONTROL
PARA MEJORA DE PROCESOS INDUSTRIALES Y MANUFACTURA
DE EQUIPOS DE COMPUTO

Otorgado el: 23 DE JUNIO DEL 2016



MTI ROSA MARIA CANTON CRODA

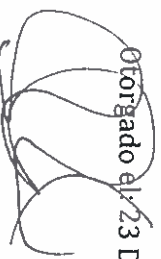
TECAPSA SA DE CV

DIPLOMA DE PARTICIPACION
ING. HUGO³ DAVID
RAMOS ESPINO

EN EL PROGRAMA DE

CAPACITACION EN AUTOMATIZACION Y CONTROL
PARA MEJORA DE PROCESOS INDUSTRIALES Y MANUFACTURA
DE EQUIPOS DE COMPUTO

Oforjado el 23 DE JUNIO DEL 2016



MTI ROSA MARIA CANTON CRODA

TECAPSA SA DE CV


DIPLOMA DE PARTICIPACION

**ING. ALEXIS RODRIGUEZ
GUERRERO**

EN EL PROGRAMA DE

CAPACITACION EN AUTOMATIZACION Y CONTROL
PARA MEJORA DE PROCESOS INDUSTRIALES Y MANUFACTURA
DE EQUIPOS DE COMPUTO

Orogrado el 23 DE JUNIO DEL 2016



MTI ROSA MARIA CANTON CRODA

TECAPSA SA DE CV

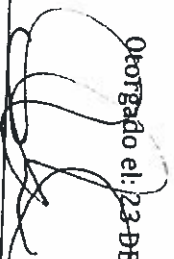
DIPLOMA DE PARTICIPACION

ING. ROCIO VALE³NZUELA
CRUZ

EN EL PROGRAMA DE

CAPACITACION EN AUTOMATIZACION Y CONTROL
PARA MEJORA DE PROCESOS INDUSTRIALES Y MANUFACTURA
DE EQUIPOS DE COMPUTO

Otrogado el: 23-DE-JUNIO DEL 2016



MTI ROSA MARIA CANTON CRODA

TECAPSA SA DE CV

DIPLOMA DE PARTICIPACION

LIC. REYNA³ KARINA
GOMEZ LEDEZMA

EN EL PROGRAMA DE

CAPACITACION EN AUTOMATIZACION Y CONTROL
PARA MEJORA DE PROCESOS INDUSTRIALES Y MANUFACTURA
DE EQUIPOS DE COMPUTO

Otorgado el: 23 DE JUNIO DEL 2016



MTI ROSA MARIA CANTON CRODA

TECAPSA SA DE CV

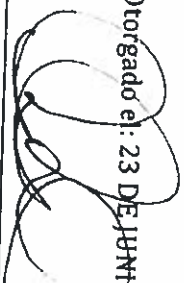
DIPLOMA DE PARTICIPACION

**ING. MAURICIO REYES
GONZALEZ**

EN EL PROGRAMA DE

CAPACITACION EN AUTOMATIZACION Y CONTROL
PARA MEJORA DE PROCESOS INDUSTRIALES Y MANUFACTURA
DE EQUIPOS DE COMPUTO

Otorgado el: 23 DE JUNIO DEL 2016



MTI ROSA MARIA CANTON CRODA

TECAPSA SA DE CV

DIPLOMA DE PARTICIPACION

LIC. YOSSELIN^a LOPEZ DE
LA TORRE

EN EL PROGRAMA DE

CAPACITACION EN AUTOMATIZACION Y CONTROL
PARA MEJORA DE PROCESOS INDUSTRIALES Y MANUFACTURA
DE EQUIPOS DE COMPUTO

Otorgado el 23 DE JUNIO DEL 2016



MTI ROSA MARIA CANTON CRODA

TECAPSA SA DE CV

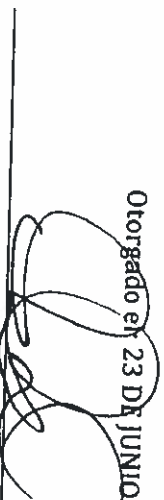
DIPLOMA DE PARTICIPACION

**LIC. MARIBEL³ LOPEZ
CEBALLOS**

EN EL PROGRAMA DE

CAPACITACION EN AUTOMATIZACION Y CONTROL
PARA MEJORA DE PROCESOS INDUSTRIALES Y MANUFACTURA
DE EQUIPOS DE COMPUTO

Otorgado el 23 DE JUNIO DEL 2016



MTI ROSA MARIA CANTON CRODA

TECAPSA SA DE CV

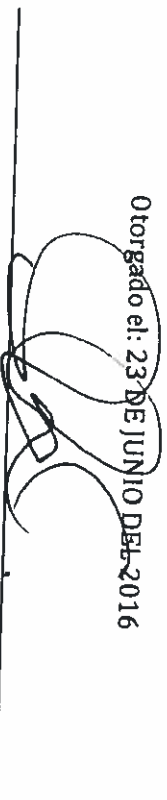
DIPLOMA DE PARTICIPACION

LIC. JOSE ANGE^lL GARCÍA
LUNA

EN EL PROGRAMA DE

CAPACITACION EN AUTOMATIZACION Y CONTROL
PARA MEJORA DE PROCESOS INDUSTRIALES Y MANUFACTURA
DE EQUIPOS DE COMPUTO

Otorgado el: 23 DE JUNIO DEL 2016



MTI ROSA MARIA CANTON CRODA

TECAPSA SA DE CV

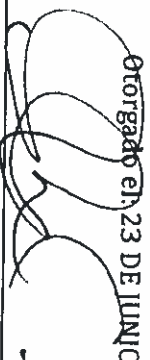
DIPLOMA DE PARTICIPACION

LIC. NIDIA FERNANDEZ
ROMERO

EN EL PROGRAMA DE

CAPACITACION EN AUTOMATIZACION Y CONTROL
PARA MEJORA DE PROCESOS INDUSTRIALES Y MANUFACTURA
DE EQUIPOS DE COMPUTO

Otorgado el 23 DE JUNIO DEL 2016



MTI ROSA MARIA CANTON CRODA

TECAPSA SA DE CV

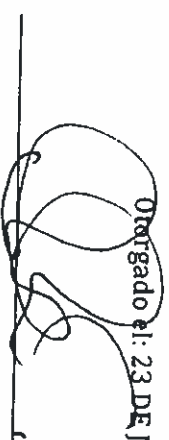
DIPLOMA DE PARTICIPACION

ING. OLGA LIDIA
CARRILLO GUERRERO

EN EL PROGRAMA DE

CAPACITACION EN AUTOMATIZACION Y CONTROL
PARA MEJORA DE PROCESOS INDUSTRIALES Y MANUFACTURA
DE EQUIPOS DE COMPUTO

Ortigado el: 23 DE JUNIO DEL 2016



MTI ROSA MARIA CANTON CRODA

TECAPSA SA DE CV

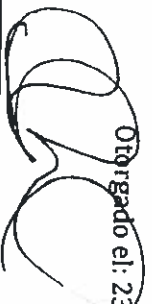
DIPLOMA DE PARTICIPACION

ING. ROGELIO^a OSWALDO
RODRIGUEZ LORIA

EN EL PROGRAMA DE

CAPACITACION EN AUTOMATIZACION Y CONTROL
PARA MEJORA DE PROCESOS INDUSTRIALES Y MANUFACTURA
DE EQUIPOS DE COMPUTO

Otorgado el: 23 DE JUNIO DEL 2016



MTI ROSA MARIA CANTON CRODA