





***Índice‌***

***I.Resúmen‌ ‌……………………………………………………………………………‌ ‌3‌ ‌***

 ‌

***II.Marco‌ ‌teórico……………………………………………………………………. 3***

1. ***Control lógico programable …………………………………………. 3***
2. ***Lógica cableada ‌……………………………………………………… ‌3***
3. ***Automatización …...……………………………………………………. 4***
4. ***Codesys …………….……………………………………………………….. 4***
5. ***Factory IO ……….…………………………………………………………. 5***

***III.Objetivo‌ ‌…………………………………………………………………………. 5***

1. ***Generales ………………………………………………………………….. 5***
2. ***Particulares ………………………………………………………………. 5***

***‌‌***

***IV.Materiales‌ ‌y‌ ‌métodos‌ ‌………………………………………………………. 5‌ ‌‌ ‌***

***‌***

***V.Resultados‌ ‌……………………………………………………………………….. 5***

***‌***

***VII.Conclusiones……………………………………………………………………***

***‌***

***VIII.Bibliografía……………………………………………………………………***

1. ***Resumen***

En esta práctica, se realiza la programación de un sistema de ensamblado mediante codesys y se simula la interfaz virtual en Factory IO

1. ***Marco teórico***
2. ***Control lógico programable***

Estos dispositivos fueron inventados a finales de la década de 1960 reemplazando los sistemas de circuitos eléctricos realizados con relés, interruptores y otros componentes comúnmente utilizados para el control de los sistemas de lógica combinacional.

Los controles lógicos programables, son ordenadores que a diferencia de las computadoras comunes, están diseñados para múltiples señales de entrada y de salida, rangos de temperatura ampliados, inmunidad al ruido eléctrico, resistencia a la vibración y el impacto, para el procesamiento de información codificable que permite automatizar un sin fin de procesos normalmente industriales.

Los procesos que se pueden automatizar son diversos siendo esta una muy solicitada tecnología en las industrias, estos procesos pueden ser: electromecánicos, electroneumáticos, electrohidráulicos. Los PLC son utilizados con frecuencia en el control de maquinaria de fábrica en líneas de montaje u otros procesos de producción.

A lo largo de los años los PLC han ido evolucionando desde su creación, se ha incluido el control de relé secuencial, control de movimiento, control de procesos, sistemas de control distribuido y comunicación por red.

Ejemplos de aplicaciones generales:

|  | Maniobra de máquinas |
| --- | --- |
|  | Maquinaria industrial de plástico |
|  | Máquinas transfer |
|  | Maquinaria de embalajes |
|  | Maniobra de instalaciones: |
|  | Instalación de aire acondicionado, calefacción... |
|  | Instalaciones de seguridad |
|  | Señalización y control |
|  | Chequeo de programas |
|  | Señalización del estado de procesos |

**Ventajas e inconvenientes**

No todos los autómatas ofrecen las mismas ventajas sobre la lógica cableada, ello es debido, principalmente, a la variedad de modelos existentes en el mercado y las innovaciones técnicas que surgen constantemente, así que en referente a este ámbito se debería analizar cada uno de los casos en los que es utilizado.

1. ***Lógica cableada***

La lógica cableada es una forma de realizar controles, en la que el tratamiento de datos (botonería, fines de carrera, sensores, presostatos, etc.), se efectúa en conjunto con contactores o relés auxiliares, frecuentemente asociados a temporizadores y contadores***.***

A través de la conexión de los contactos de los diferentes elementos involucrados, se ejecutan secuencias de activación, desactivación y temporizaciones de los diferentes elementos que permiten realizar el manejo de la maquinaria (contactores, válvulas, pistones, calefactores, motores, etc.).

El uso de relés auxiliares hace posible incrementar la cantidad de contactos disponibles para realizar la lógica, lo cual habitualmente es necesario, además de servir de interfaz al manejar diferentes niveles de voltaje (24 a 220 Vac y viceversa, por ejemplo). Las temporizaciones también son recurrentes, por lo cual uno o más temporizadores son comúnmente encontrados en estos sistemas.

1. ***Automatización***

La automatización se define como el uso de sistemas o elementos computarizados, electromecánicos, electroneumáticos y electrohidráulicos para fines industriales.

Como una disciplina de la ingeniería más amplia que un sistema de control, abarca la [instrumentación industrial](https://es.wikipedia.org/wiki/Instrumentaci%C3%B3n_industrial), que incluye los sensores, los transmisores de campo, los [sistemas de control](https://es.wikipedia.org/wiki/Sistema_de_control) y supervisión, los sistemas de transmisión y recolección de datos y las aplicaciones de software en tiempo real para supervisar y controlar las operaciones de plantas o procesos industriales.

Algunos sistemas que componen la automatización industrial, son:

* Electricidad industrial.
* Neumática industrial.
* Oleohidráulica industrial.
* Autómatas programables.
* Comunicaciones industriales.
* Robótica industrial.

1. ***Codesys***

Este es un entorno de desarrollo para la programación de controladores conforme con el estándar industrial internacional [*IEC 61131-3*](https://es.wikipedia.org/wiki/IEC_61131-3)*.* El término CODESYS es un [acrónimo](https://es.wikipedia.org/wiki/Acr%C3%B3nimo) y significa Sistema de Desarrollo de Controladores.

CODESYS es desarrollado y comercializado por la empresa de software alemán 3S-Smart de Soluciones de Software situado en la ciudad bávara de Kempten. La Primera Versión (1,0) fue creada en 1994.

CODESYS puede descargarse desde el sitio web de la compañía.

Los seis lenguajes de programación para aplicaciones vienen definidos en el IEC 61131-3 y están disponibles en el entorno de desarrollo CODESYS.

### Lenguajes de texto

IL (lista de instrucciones) es un lenguaje de programación parecido al lenguaje ensamblador.

[ST (texto estructurado)](https://es.wikipedia.org/wiki/Structured_Text) es similar a la programación en PASCAL o C

### Lenguajes gráficos[[editar](https://es.wikipedia.org/w/index.php?title=CoDeSys&action=edit&section=3)]

[LD (Diagrama Ladder)](https://es.wikipedia.org/wiki/Lenguaje_Ladder) permite al programador combinar los contactos de relé y las bobinas. Es el lenguaje de programación de PLC por excelencia.

FBD (diagrama de bloques de función) permite al usuario programar rápidamente, tanto expresiones como en lógica booleana.

SFC (Bloques de función secuenciales) es conveniente para los procesos de programación secuencial

Dispone también de un editor gráfico que no está definido en la norma IEC:

CFC (Continuous Function Chart) es una especie de editor de FBD libre. Es un editor orientado a FBD donde las conexiones entre las entradas, salidas y los operadores se fijan automáticamente. Todas las cajas se pueden colocar libremente, lo que permite programar ciclos de retroalimentación provisional sin variables.

1. ***Factory IO***

Factory I / O es una simulación de fábrica en 3D para el aprendizaje de tecnologías de automatización. Diseñado para ser fácil de usar, permite construir rápidamente una fábrica virtual utilizando una selección de piezas industriales comunes. Factory I / O también incluye muchas escenas inspiradas en aplicaciones industriales típicas, que van desde niveles de dificultad principiantes hasta avanzados.

El escenario más común es utilizar Factory I / O como una plataforma de capacitación de PLC, ya que los PLC son los controladores más comunes que se encuentran en aplicaciones industriales. Sin embargo, también se puede utilizar con microcontroladores, SoftPLC, Modbus, entre muchas otras tecnologías.

1. ***Ensamblador***

Los ensambladores dentro de la industria y como su nombre lo dice, son líneas de producción que hacen uso de la automatización, sistemas oleoneumáticos y mecánicos para poder realizar la operación de armado de ensambles con diferentes piezas.

***III. Objetivos***

1. **Generales**

El objetivo general de esta práctica, es automatizar una línea ensambladora, la cual debe ser programada en Codesys y simulada en Factory IO.

1. **Particulares**

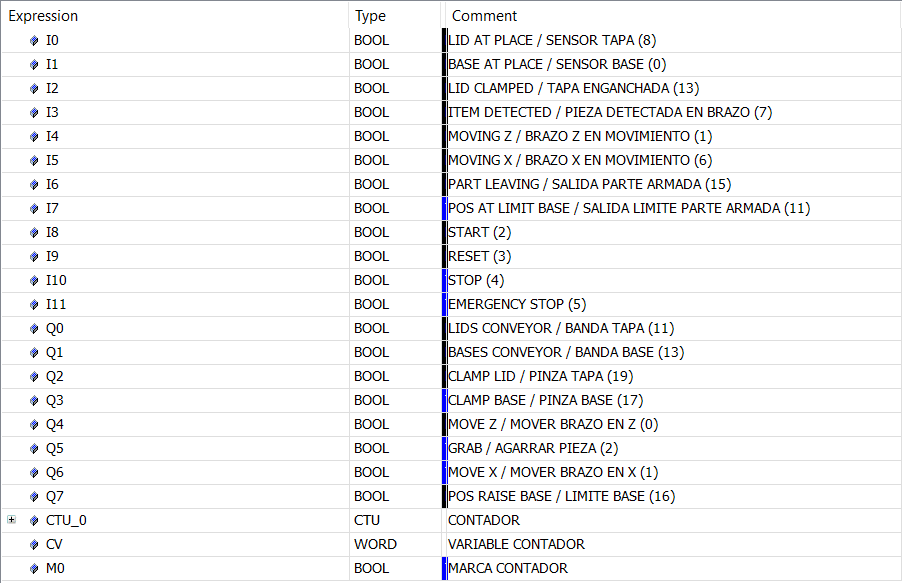
La línea de ensamble debe de contar con dos bandas las cuales transportarán piezas de dos partes diferentes del ensamble, las cuales deben de ser detenidas en una coordenada específica mediante un sensor y clamp para poder ser ensambladas y así pasar a una sola banda que tendrá un contador a la salida, la línea de ensamble debe de contar con un brazo, dos bandas transportadoras y una base de control para operar el proceso con las acciones de prendido, paro de emergencia, alto .

***IV. Materiales y métodos***

Para poder realizar la programación del proceso se utiliza Codesys el cual se conecta con Factory IO para poder mostrar físicamente la simulación del proceso.

***V. Resultados***

Para realizar la automatización de este proceso primero se tuvieron que analizar y declarar las entradas y salidas del sistema, las cuales se muestran en la Figura 1.

******

***Figura 1. Entradas y salidas de línea de ensamble***

Después de definir las variables se procede a programar las funciones en grafcet ***Figura 2***., las cuales indicarán la inicialización del proceso mediante el botón de Start, el cual dará paso a la salida de las partes del ensamble en ambas bandas transportadoras.

Para poder simular la programación ya realizada después de colocar las partes de la línea y acomodarlos correctamente en el simulador de Factory IO se verifican las variables de los actuadores y sensores que se están utilizando. (***Figura 2.1b***)

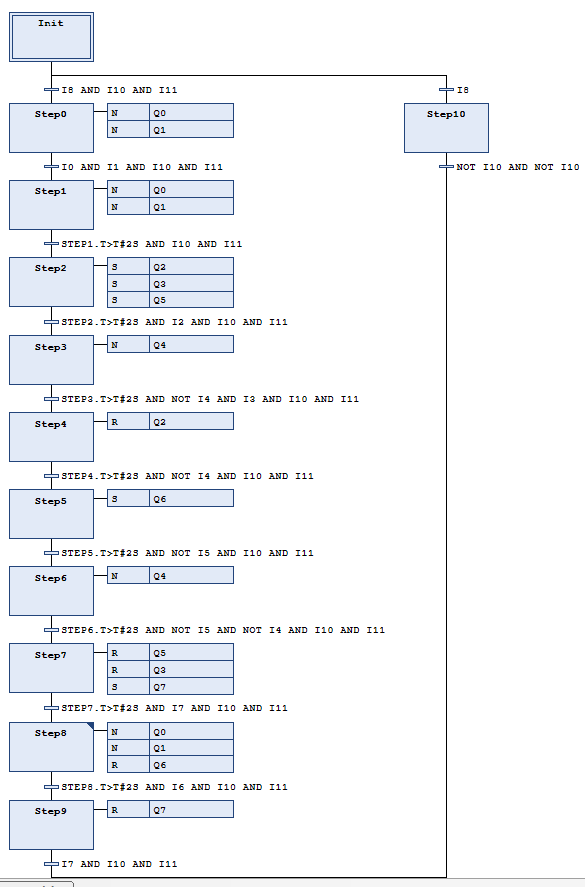


Figura 2. Programación en Grafcet de línea de ensamble

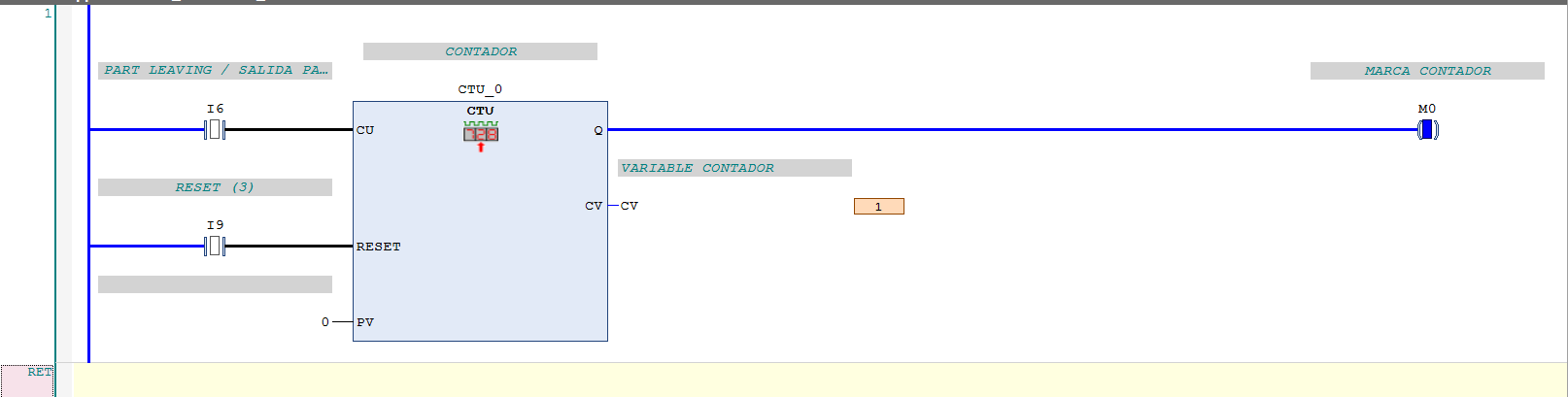


Figura 2.1a subprograma “contador”

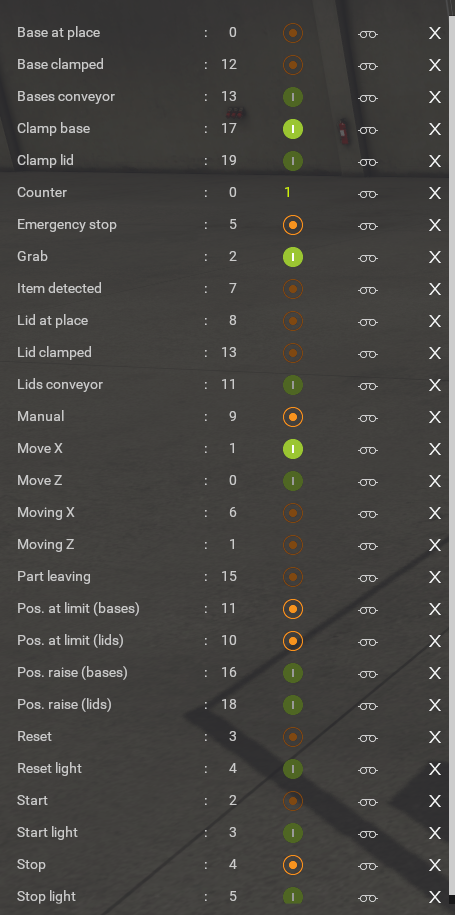


Figura 2.1b Variables declaradas en Factory IO

Teniendo esto se realiza la conexión del simulador con el PLC de codesys declarando las variables necesarias como son los actuadores y los sensores ***Figura 2.1c y 2.1d***



Figura 2.1c Actuadores



Figura 2.1 d Sensores

Posteriormente respecto al avance de ambos productos por las transportadoras al llegar al sensor y ser detectadas, se tiene un temporizador de 2 segundos para evitar interferencias y posteriormente se activan los clampers, desactivando las transportadoras que impedirán que las piezas sigan avanzando.

Ya que están detenidas por los clampers, se activa el eje z del brazo, hasta llegar al punto donde se encuentra la primera parte y se activa el proceso 1 de “grab”, el cual sostiene la pieza para después nuevamente activar el movimiento del brazo en z y después en x ***Figura 4.***, lo cual permitirá desplazarse hasta la segunda línea transportadora en donde se encuentra el complemento del ensamble, bajando nuevamente en el eje z, , reseteando el Grab para que suelte la pieza y liberando nuevamente el clamp de de tal manera que la pieza quede ensamblada ***Figura 5***.

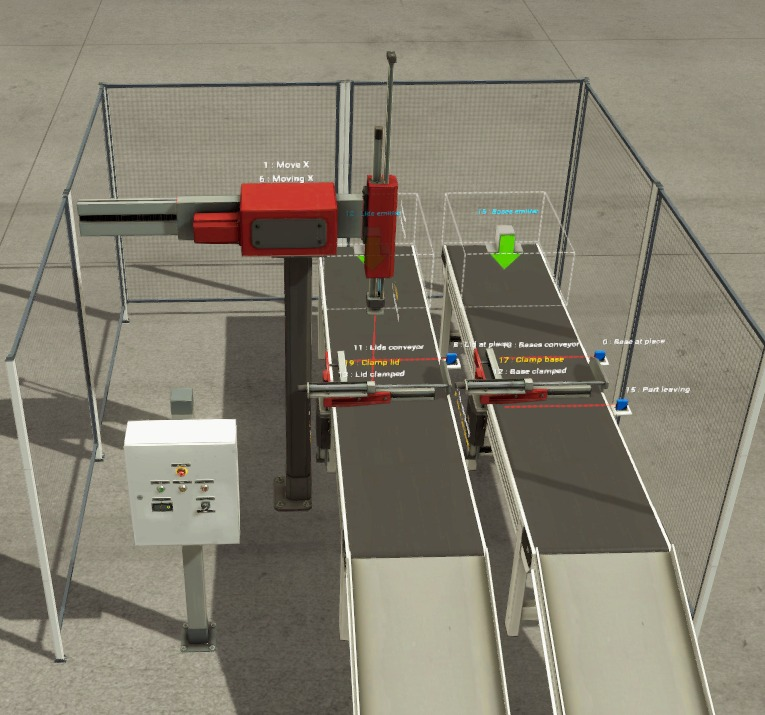


Figura 3. Inicialización del proceso de ensamblado

En el momento en el que es detectado por el sensor, el brazo regresa por eje z primero y después por el eje x para así volver a su posición original, subiendo nuevamente las barras y reactivando las líneas transportadoras para comenzar el proceso nuevamente

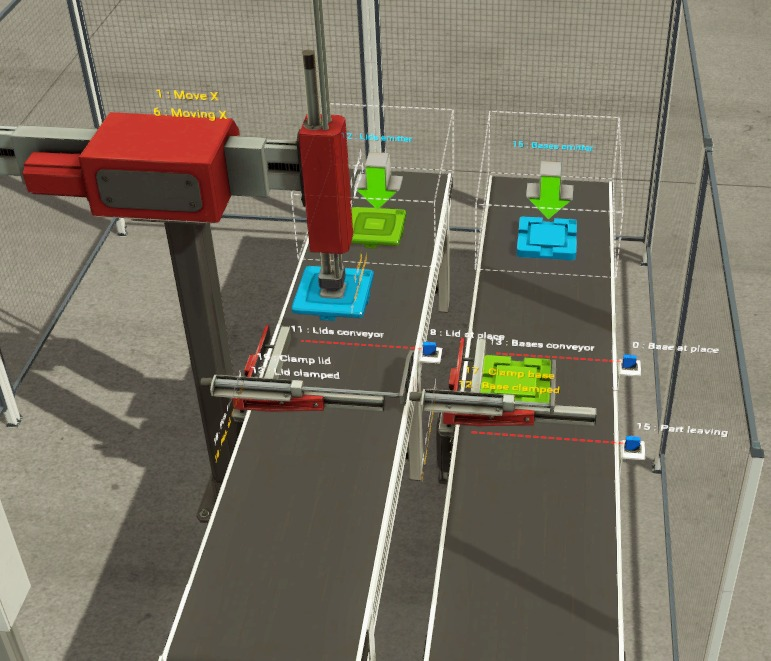


Figura 4. Movimiento de brazo para ensamble

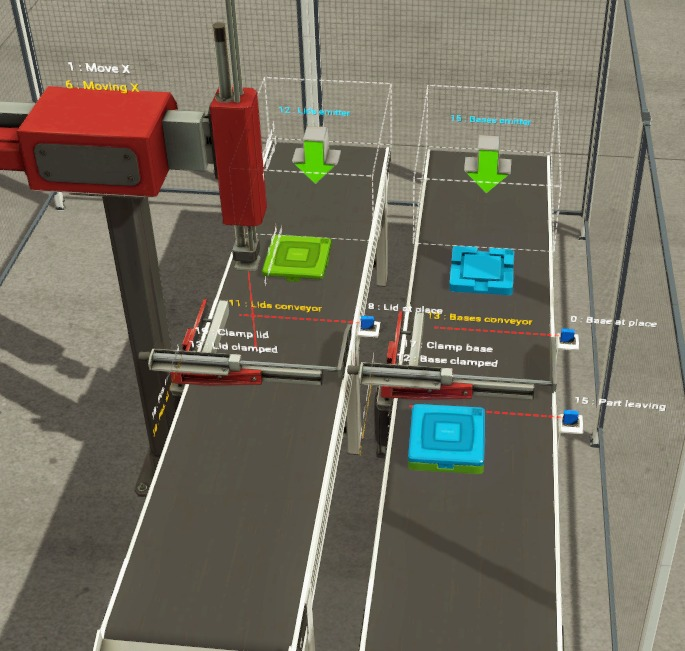


Figura 5. Ensamble listo y posición de los actuadores re-inicializados

Cuando se activan nuevamente las líneas transportadoras, permite al producto terminado seguir avanzando por ellas y un sensor las detecta para realizar un conteo como se muestra en la orden del contador en el subprograma de la Figura 2.1a., dentro de la programación de Grafcet de Codesys.

De esta manera permite que el programa siga con el mismo proceso hasta que se active el botón de STOP o el botón de paro de emergencia.

Es importante destacar que para cada uno de los botones también se agregó un led el cual permite al operador observar si está activado.

***VII. Conclusiones***

Axel Arriola Fonseca.

El presente proyecto puso a prueba nuestras habilidades para resolver problemas y programación de PLCs en un situación de la industria real, así como otros ejercicios aplicados en clase, con el uso de diferentes lenguajes de programación, a veces es más fácil resolver un problema por un lenguaje que por otro, en este caso optamos por Grafcet, el cual nos ayudó a identificar mejor los pasos en esta ensambladora, ya que tenia pasos secuenciales, lo unico que fuimos modificando fueron las variables para las distintas transiciones. Al final se logró resolver la problemática, lo más importante fue el planteamiento de las ideas.

Dennis Ivan Pérez Montiel.

Podemos concluir que el uso de PLC es de muy útil al momento de automatizar, pero para esta práctica fue necesario observar el comportamiento del proceso, y hacer una planeación de cómo debería funcionar, qué sensores se activan y qué actuadores activar, los tiempos y el orden, así la programación se volvió más sencilla y ordenada, evitando errores en nuestro programa.

Scarlett A. Cisneros Aymerich

Es importante recalcar que en el mundo de la automatización es necesario poder analizar de forma lógica las variables que se requieren del sistema dependiendo de las necesidades del mismo, en este caso, fue muy útil el poder observar con el simulador el proceso de ensamble de la línea, ya que de esta manera se puede tener un panorama más amplio de en qué momento y de qué forma se activan los sensores y actuadores, para que así al momento de programar pueda llevarse el proceso con más claridad y orden.

El punto más enriquecedor de la práctica ha sido el visualizar que para poder poner en marcha la automatización de un proceso existen diferentes métodos y que esta diversidad ayuda a los programadores a tener una variedad de elección amplia dependiendo de las necesidades del sistema.

***VIII. Bibliografía***

* *What IS a PLC?» (en inglés). Consultado el 23 de enero de 2013.*
* *M. A. Laughton, D. J. Warne (ed), Electrical Engineer's Reference book, 16th edition,Newnes, 2003 Chapter 16 Programmable Controller*
* *Alison Dunn (12 de septiembre de 2008).* [*«The father of invention: Dick Morley looks back on the 40th anniversary of the PLC»*](https://web.archive.org/web/20190609030841/https://www.automationmag.com/features/the-father-of-invention-dick-morley-looks-back-on-the-40th-anniversary-of-the-plc.html) *(en inglés). Archivado desde* [*el original*](http://www.automationmag.com/features/the-father-of-invention-dick-morley-looks-back-on-the-40th-anniversary-of-the-plc.html) *el 9 de junio de 2019. Consultado el 23 de enero de 2013.*
* *PLC. (s. f.). PLC. Recuperado 8 de junio de 2021, de* [*http://www.sc.ehu.es/sbweb/webcentro/automatica/WebCQMH1/PAGINA%20PRINCIPAL/PLC/plc.htm*](http://www.sc.ehu.es/sbweb/webcentro/automatica/WebCQMH1/PAGINA%20PRINCIPAL/PLC/plc.htm)
* *Revista ElectroIndustria - Desde la lÃ3gica cableada a los Micro Automatismos. (s. f.). Revista Electro Industria. Recuperado 8 de junio de 2021, de* [*http://www.emb.cl/electroindustria/articulo.mvc?xid=32&ni=desde-la-logica-cableada-a-los-micro-automatismos*](http://www.emb.cl/electroindustria/articulo.mvc?xid=32&ni=desde-la-logica-cableada-a-los-micro-automatismos)
* *Games, R. (s. f.). Factory I/O - Documentation. Factory IO. Recuperado 20 de julio de 2021, de* [*https://docs.factoryio.com/*](https://docs.factoryio.com/)