

### Introducción



[LabVIEW](#) de National Instruments es un Entorno de desarrollo integrado especializado en informática industrial y científica. Su particularidad es que se basa en el lenguaje G (G por Gráfico), creada por Nacional Instrumentos que es enteramente gráfica. Permite el desarrollo de programas informáticos complejos facilitando al mismo tiempo la programación y en consecuencia disminuir los plazos de desarrollo. Gracias a sus librerías de funciones dedicadas a la adquisición de datos, la instrumentación, al análisis matemático de las mediciones y la visualización, [LabVIEW](#) se dedica especialmente a los bancos de pruebas y mediciones.

### ¿ [LabVIEW](#) por qué hacer?

Como lo vimos en introducción, [LabVIEW](#) es especialmente conveniente a la informática industrial y científica. Podrá pues utilizarlo para el desarrollo de:

- Software para Windows, UNIX/Linux o Mac, Windows Mobile o Palm OS,
- Librerías (DLL, Activo X, .NET),
- Controles de instrumentos,
- componentes embarcados,
- componentes tiempo real,
- tarjetas FPGA.

### [LabVIEW](#) y el hardware

Lo vimos, [LabVIEW](#) permite programar sobre mucho orienta diferentes. Lo mismo sucede para el material y sobre todo la instrumentación. En efecto, con [LabVIEW](#) y gracias a sus numerosas librerías, podrá intercomunicarse y encargar las siguientes tarjetas y aparatos:

- [VXI](#) , [PXI](#) , Compacto PCI,
- PCI,
- USB, FireWire,
- Serie.

Comunicando con los siguientes protocolos:

- [VXI](#) , [PXI](#) , Compact PCI,
- PCI express, [PXI](#) express,
- PCI,
- USB, FireWire,
- serie RS 232, 422, 485...
- TCP/IP, UDP
- Modbus RTU, Modbus TCP, Profibus, otros protocolos industriales...
- [Bluetooth](#) , WIFI.

### El lenguaje gráfico

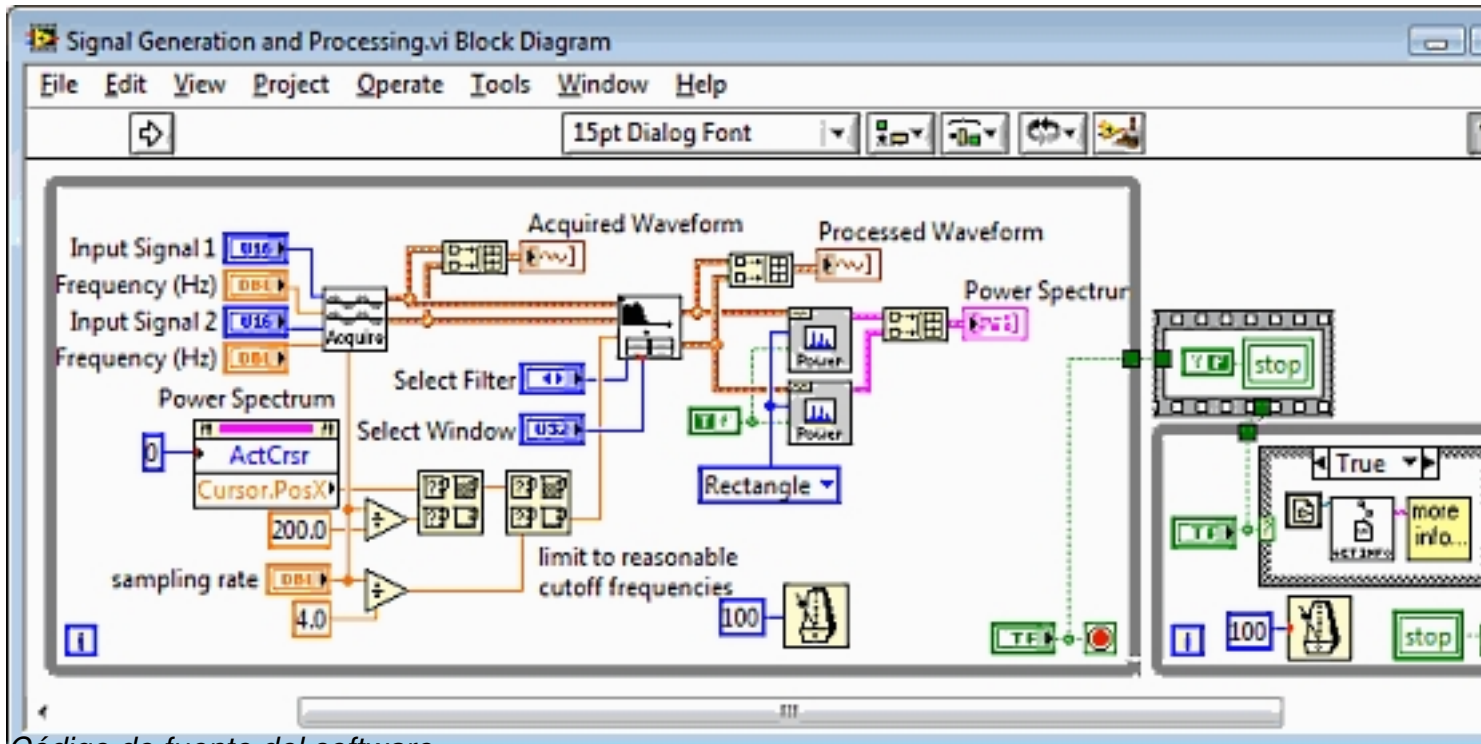
Con [LabVIEW](#) , no se programa el software a escribir líneas de códigos con una sintaxis compleja. La programación se hace con iconos los cuales representan funcionalidades, ligados entre ellos por cables quienes representan los flujos de datos (un poco a la manera de una tarjeta electrónica con sus componentes y circuitos integrados).

Esta representación muy adornada del código es cerca de la concepción como uno hace: con esquemas; eso, por supuesto, facilita mucho el trabajo que necesita la programación del concepto. Esta abstracción del lenguaje gráfico no requiere de ser un experto en programación para desarrollar software sencillos. De igual forma, para software mas complejos, el cliente "profano" en programación pero experto en su dominio, podrá leer y entender la idea y así guiar el ingeniero experto en programación.

Para ilustrar el sencillo que es el lenguaje G, mira el ejemplo que sigue: un software que genera señal y de tratamiento.

{flowplayer}http://www.ajolly.fr/images/stories/[LabVIEW](#)/signal-generation-and-processing.flv{/flowplayer}

*Vídeo del software*



Código de fuente del software

## Ejemplos de proyecto con [LabVIEW](#)

- [Inspección de bloques de fusibles](#) (Visión, industria automóvil)
- [Análisis de gas](#) (instrumentación, industria petroquímica)
- [Supervisor de control de calidad de arena](#) (comunicación con [PLC](#), materiales de construcción)
- [Validación de ECM \(Electronic Control Module\)](#) (industria automóvil)
- [Estudios clínicos de la piel](#) (instrumentación, cosmetología)

## Alternativas a [LabVIEW](#)

Otras herramientas permiten usar la mismas funcionalidades pero en otros lenguajes:

- lenguaje C, ver el artículo "[LabWindows/CVI](#)"
- lenguaje C#, Visual Basic, C++, ver el artículo "[Measurement Studio](#)"
- En línea de producción se usa bastante [LabVIEW](#) con un secuenciador como [TestStand](#)

[d](#) . Esta

pareja permite agilizar el desarrollo: con

[LabVIEW](#)

se desarrollo "ladrillos" sencillos que se integran en una secuencia de

[TestStand](#)

que se dedica a gestionar los datos, generar reportes y manejar la interfaz grafica.