

Práctica 1: LabVIEW

Tomás Ricardo Basile Álvarez

23 de julio de 2021

Resumen

LabVIEW es un lenguaje de programación que ofrece un método de trabajo sencillo por medio de una interfaz gráfica interactiva. En LabVIEW los programas se van construyendo en la interfaz gráfica como si fuera un diagrama de flujo, por lo que es fácil de utilizar y de depurar. LabVIEW puede ser un programa muy útil para controlar, automatizar y monitorear instrumentos experimentales y para comunicar al usuario con el experimento, ya que es compatible con muchos instrumentos distintos. En esta práctica vemos algunos conceptos básicos de cómo funciona LabVIEW y realizamos 4 programas sencillos.

1. Introducción y Teoría

LabVIEW es una plataforma de desarrollo creada por National Instruments con un lenguaje de programación con el objetivo de automatizar el instrumental de laboratorio. En un documento de LabVIEW se trabaja en dos entornos:

- **Front Panel:** Es la interfaz sobre la que colocamos todos los elementos o controles con los que interactúa el usuario para la entrada de datos y la lectura de salidas de datos.
- **Block Diagram:** Es el entorno en el que se hace la programación. Aquí se hace el diagrama de flujo del programa usando comandos que se representan por bloques.

En el Block Diagram se colocan bloques que representan las distintas funciones del programa. Existen varios tipos de bloques que usamos según lo que querramos hacer, enlisto aquí algunos tipos de bloques que usamos:

- **De entrada:** Bloques para la entrada de datos por parte del usuario, ya sean numéricos, cadenas, booleanos, etc. También se pueden tener bloques que toman como entrada valores provenientes de los instrumentos del laboratorio.
- **Funciones y flujo del programa:** En esta categoría podemos incluir bloques que se utilizan para estructuras de control como el ciclo for o while, también funciones matemáticas, operadores lógicos, etc.
- **Salida:** Son los bloques que utilizamos para representar los resultados del programa. Pueden ser por ejemplo gráficas, histogramas o la impresión de datos numéricos.

Además podemos trabajar con distintos tipos de datos, que se representan por diferentes colores:

- **Número:** Tipo de dato para almacenar números reales.
- **Cadena:** Tipo de dato que almacena una cadena de caracteres alfanuméricos.
- **Path:** Tipo de dato para guardar la ubicación de archivos en los directorios de la computadora.
- **Booleano:** Tipo de dato que sólo puede guardar valores de verdadero y falso

2. Experimento

Usaremos los tipo de datos y las funciones mencionadas en la sección anterior para crear 4 programas distintos que describimos a continuación.

1. Operaciones Básicas y Tipos de datos

Este programa consiste de 6 secciones para estudiar los distintos tipos de ingreso y manipulación de datos.

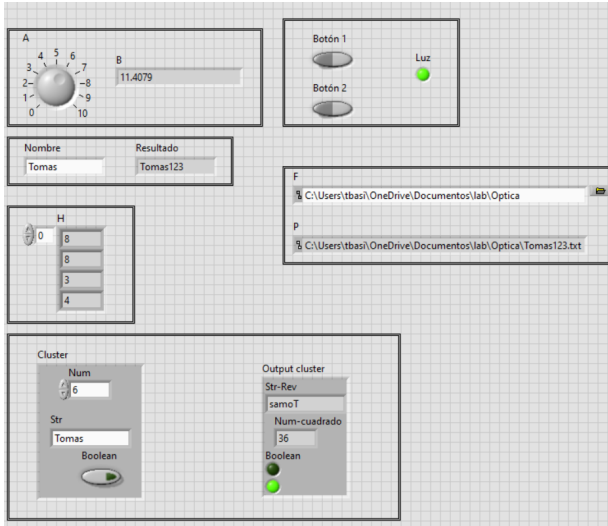


Figura 1: Front Panel del programa

una luz dependiendo del valor del booleano.

2. Generación de N números aleatorios entre 0 y 1 y graficación

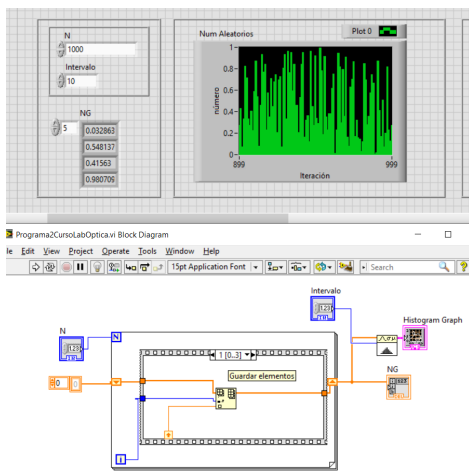


Figura 2: Front Panel y Block Diagram del programa 2

3. Generar una secuencia de ceros y unos y revisar su aleatoriedad

Este programa toma como entrada un número N y genera una lista aleatoria de ceros y unos de longitud N . Luego, guarda esta lista en un archivo .txt y le realiza un análisis de Borel para

La primera sección consiste de una perilla para tomar la entrada de un dato numérico y un indicador donde se imprime dicho número.

La segunda sección consiste de dos botones que se usan como entradas de datos booleanos. Con estos datos se usa una compuerta tipo and para prender una luz si ambos botones están encendidos.

La tercera sección toma como entrada una cadena, le agrega los caracteres '123' y la imprime. La cuarta sección consiste en la construcción de una lista de números.

La quinta sección toma como entrada un path y luego le agrega el resultado de la sección 2 al path.

La última sección es un cluster de datos, contiene un dato numérico, una cadena y un booleano como entradas. Luego lo que hace es voltear la cadena, elevar al cuadrado el número y prender

Como se ve en la imagen del Front panel, este programa toma como entrada un número N y un número llamado 'intervalo'. Luego, se generan N números aleatorios entre 0 y 1, se guardan en una lista y se grafican (como se ve en la gráfica en la parte superior de la figura 2). Además, se genera un histograma con la cantidad de intervalos establecida (no se muestra en la figura).

En el Block Diagram podemos ver que el programa toma como entrada el número N y luego realiza un ciclo for, dentro del cual se hace una secuencia de pasos que consiste en crear un número aleatorio, agregarlo a la lista y luego agregarlo a la gráfica.

comprobar que sea aleatoria.

El análisis de Borel que realizamos consiste en primero revisar que la proporción de ceros y unos en la secuencia. Luego, se analizan todas las posibles secuencias de longitud dos dentro de la lista y se comprueba cuántas hay de cada tipo posible (00, 01, 10, 11). Cada una de ellas debe de aparecer aproximadamente 25 % de veces.

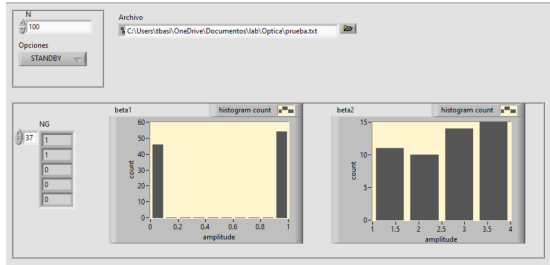


Figura 3: Front Panel del programa 3

te programa se utilizó una función importante de LabVIEW, que consiste en guardar un archivo completo que tiene ciertas entradas y salidas como un bloque que luego se utiliza en otros archivos.

4. Enviar y recibir datos de un instrumento

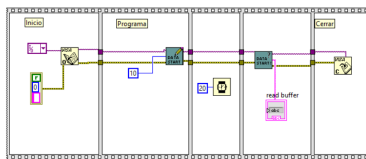


Figura 4: Block Diagram del programa 4

de LabVIEW.

El programa primero genera los N dígitos aleatorios (para ello crea números aleatorios en $[0, 1]$ y luego se usa un comparador para convertir los resultados menores a 0,5 en ceros y los mayores a 0,5 en unos). El programa crea un histograma con la cantidad de ceros y unos (primer histograma de la figura 3) Luego, el programa va revisando cada pareja de dígitos en el array y cuenta cuántos patrones de la forma 00, 10, 01, 11 hay, para finalmente crear un histograma con la cantidad de parejas de cada tipo (segundo histograma de la figura). Para este

En este programa se revisa una de las funciones más importantes de LabVIEW. Consiste en conectarse con instrumentos y tomar datos de él. Dado que no tenemos un instrumento que conectar como prueba, sólo vemos la estructura general del programa, que consiste en abrir el instrumento, mandar instrucciones por medio de comandos y leer información. Todo esto se realiza con el protocolo VISA que forma parte

3. Conclusiones

LabVIEW es una herramienta muy útil en el trabajo experimental por su capacidad de conectarse con instrumentos experimentales, mandar instrucciones y recolectar distintos tipos de datos. Además, como se observa en los ejemplos realizados, LabVIEW permite crear programas usando una interfaz gráfica intuitiva en la que se escribe el programa como un diagrama de flujo. LabVIEW tiene varias funcionalidades distintas para tomar entradas de datos, manipularlos, operar con ellos y finalmente analizarlos y graficarlos, por lo que se pueden crear programas de todo tipo con esta herramienta. Habiendo dicho todo esto, queda muy claro la gran utilidad de LabVIEW como herramienta de trabajo para apoyar durante el trabajo experimental.

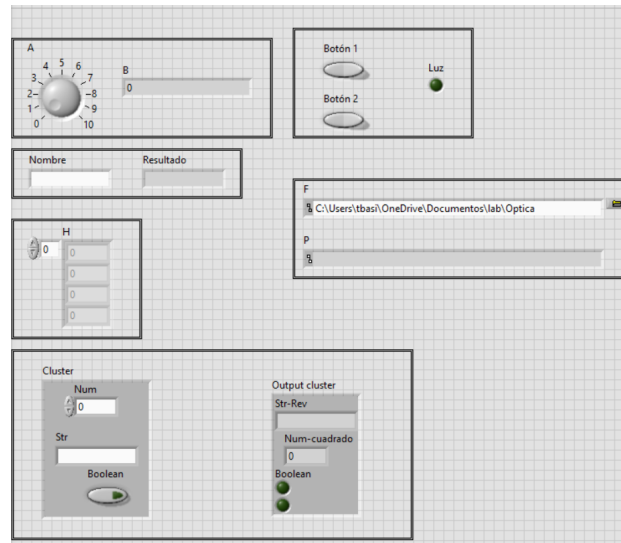
4. Referencias

- 1 Ramírez, H. C. (2017). *LabVIEW I: una breve introducción*. Recuperado 20 de Julio de 2021 de <http://www.paginaspersonales.unam.mx/app/webroot/files/5268/Labview I.pdf>
- 2 Ramírez, H.C (2021). *Laboratorio de óptica 2021-2: Lecciones 01-03*. Videos en la plataforma de *Google Classroom*

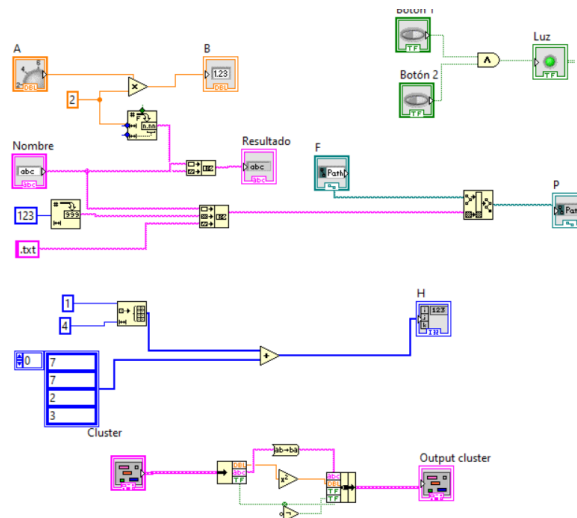
Anexo

Anexo algunas imágenes de los programas.

Programa 1:

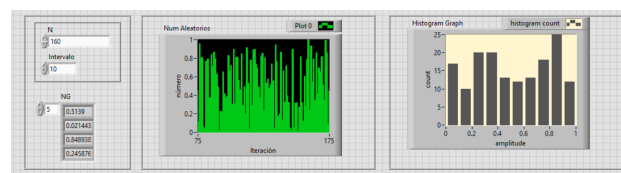


Front Panel programa 1

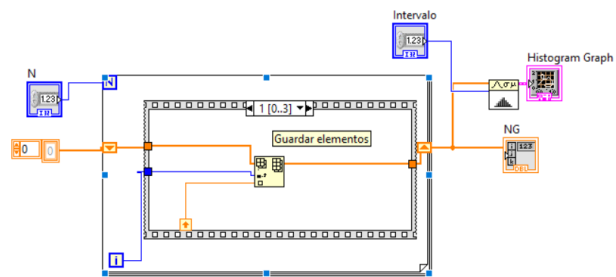


Block Diagram programa 1

Programa 2:

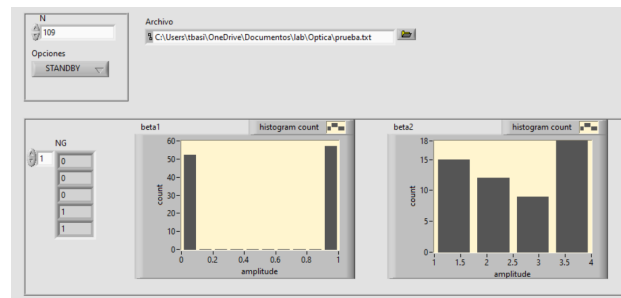


Front Panel programa 2

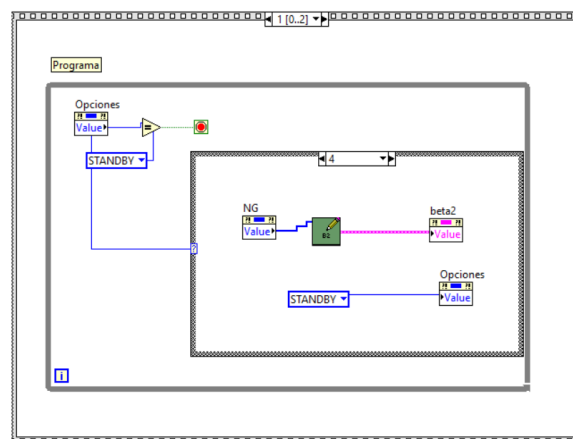


Block Diagram programa 2

Programa 3:

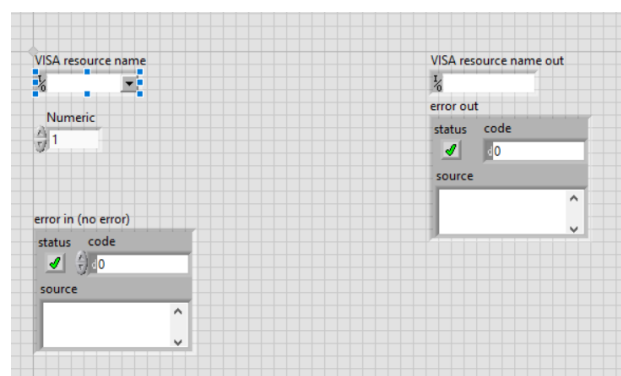


Front Panel programa 3

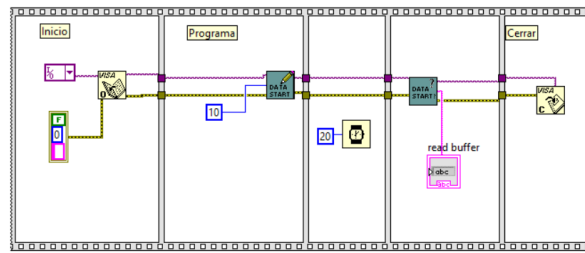


Block Diagram programa 3

Programa 4:



Front Panel programa 4



Block Diagram programa 4