

# Certified LabVIEW Associate Developer (CLAD)

## Resumen de certificación y el examen

### Descripción de la certificación en LabVIEW:

El programa de certificación de National Instruments consta de los siguientes tres niveles de certificación:

- CLAD: Desarrollador Asociado Certificado de LabVIEW (*Certified LabVIEW Associate Developer*)
- CLD: Desarrollador Certificado de LabVIEW (*Certified LabVIEW Developer*).
- CLA: Arquitecto Certificado de LabVIEW (*Certified LabVIEW Architect*).

Cada nivel constituye un requisito previo para el siguiente nivel de certificación.

El nivel **CLAD** demuestra un amplio y completo conocimiento de las principales características y funcionalidades que están disponibles en el entorno de desarrollo LabVIEW Full Development System así como la capacidad de aplicar ese conocimiento para desarrollar, depurar y mantener pequeños módulos de LabVIEW. La experiencia típica para el nivel CLAD es de aproximadamente 6 a 9 meses en la utilización de *LabVIEW Full Development System*.

El nivel **CLD** acredita experiencia en el desarrollo, depuración, distribución y mantenimiento de aplicaciones LabVIEW de media a gran escala. El nivel CLD indica que su poseedor es un profesional con una experiencia acumulada aproximada de 12 a 18 meses en el desarrollo de medianas y grandes aplicaciones LabVIEW.

El nivel **CLA** demuestra completo dominio de las arquitecturas de aplicaciones LabVIEW para un entorno de múltiples desarrolladores. Un profesional con certificación CLA no sólo posee los conocimientos técnicos y experiencia de desarrollo de software para descomponer las especificaciones de un proyecto en componentes manejables en LabVIEW, sino que además dispone de experiencia para monitorizar el proyecto a través de la utilización eficaz de herramientas de gestión de proyectos y configuraciones. El poseedor de una certificación CLA es un profesional con una experiencia acumulada aproximada de 24 meses en el desarrollo de aplicaciones medianas y grandes de LabVIEW.

**Nota.** La certificación de nivel CLAD es un requisito previo para poder examinarse del nivel CLD. El nivel de certificación de CLD es un requisito previo para poder examinarse del nivel CLA. No hay excepciones a este requisito para cada examen.

# **Certified LabVIEW Associate Developer (CLAD)**

## **Resumen de certificación y el examen**

### **Descripción del examen**

Producto: de LabVIEW Full Development System para Windows versión 8.0. Para obtener más detalles sobre las funciones disponibles en LabVIEW Full Development System consulte en [www.ni.com/LabVIEW/how\\_to\\_buy.htm](http://www.ni.com/LabVIEW/how_to_buy.htm).

Duración del examen: 1 hora.

Número de preguntas: 40

Estilo de preguntas: selección múltiple

Nota para aprobado: 70%

El examen valida el conocimiento de la aplicación, pero no la capacidad para recordar los pasos del menú o los nombres de los VIS y de los componentes.

El uso de LabVIEW o de cualquier otro recurso externo está prohibido durante el examen. Con el objetivo de proporcionar ayuda en el examen, allí donde sea necesario, se proporcionan capturas de la pantalla de la ayuda de LabVIEW.

Para mantener la integridad del examen, esta prohibido copiar o reproducir cualquier sección del examen. El incumplimiento dará lugar al suspenso. Si el examen se realiza en versión impresa, la retirada de la grapa del documento supondrá el suspenso sin evaluación.

### **Logística del examen**

Estados Unidos y Europa: El examen del nivel de CLAD se puede realizar en los centros de prueba de *Pearson Vue*. El examen está basado en PC y los resultados están disponibles inmediatamente después de concluir el examen. Para obtener más detalles y conocer la programación de los exámenes, consúltelo en [www.pearsonvue.com](http://www.pearsonvue.com).

En Asia: el examen es en papel, por lo que las evaluaciones y los resultados tardan aproximadamente 4 semanas. Por favor, póngase en contacto con su oficina local de National Instruments para obtener más detalles y conocer la programación de los exámenes.

Para hacer preguntas o comentarios generales, utilice este correo electrónico: [certification@ni.com](mailto:certification@ni.com).

# **Certified LabVIEW Associate Developer (CLAD)**

## **Resumen de certificación y el examen**

### **Temas del examen**

1. Principios de programación LabVIEW
2. Entorno LabVIEW
3. Estructuras software en LabVIEW
4. VIs y funciones de programación
5. VIs y funciones de comunicación y sincronización de datos
6. VIs y funciones del Servidor de VIs
7. VIs y funciones de gestión de errores
8. Patrones de diseño
9. Diseño de SubVIs
10. Herramientas y técnicas de depuración
11. Prácticas (estilos) de diseño y documentación de VIs
12. Memoria, rendimiento y determinismo

# Certified LabVIEW Associate Developer (CLAD)

## Resumen de certificación y el examen

### Temas del examen (Descripción):

Tópico	Sub-Tópico
1. Principios de programación LabVIEW	<ul style="list-style-type: none"> <li>a. Flujo de datos</li> <li>b. Polimorfismo</li> </ul>
2. Entorno LabVIEW.	<ul style="list-style-type: none"> <li>a. Ventana del panel frontal, diagrama de bloques y panel del conector</li> <li>b. Menús y paletas</li> <li>c. Opciones de configuración</li> </ul>
3. Estructuras software en LabVIEW	<ul style="list-style-type: none"> <li>a. Ventana del panel frontal y objetos del diagrama de bloques <ul style="list-style-type: none"> <li>i. Controles, indicadores, controles de E/S y referencias (refnums)</li> <li>ii. Terminales, constantes y nodos</li> <li>iii. Paletas, modos de actualización y leyendas de gráficos tipo “graph” y tipo “chart”</li> <li>iv. Acción mecánica de objetos booleanos</li> <li>v. Nodos de propiedad</li> </ul> </li> <li>b. Tipos y estructuras de datos <ul style="list-style-type: none"> <li>i. Tipos de dato numérico, cadena de caracteres, booleano y ruta</li> <li>ii. Tipos de dato cluster y array</li> <li>iii. Tipos de dato forma de onda y estampado de tiempos</li> <li>iv. Tipos de dato “variant”</li> </ul> </li> <li>c. Trabajo con objetos y tipos de dato en la ventana del panel frontal <ul style="list-style-type: none"> <li>i. Rangos, formatos, representación y escalas</li> <li>ii. Controles personalizados</li> <li>iii. Definiciones de tipos y definiciones de tipos estrictos</li> </ul> </li> <li>d. Estructuras de control de programa y almacenamiento de datos <ul style="list-style-type: none"> <li>i. Estructuras de bucles For y While <ul style="list-style-type: none"> <li>a. Indexado en los límites del bucle</li> <li>b. Registros de desplazamiento</li> </ul> </li> <li>ii. Estructuras de Casos y Secuencias <ul style="list-style-type: none"> <li>a. Estructuras de secuencias planas y apiladas</li> <li>b. Valores de selección de casos y tipos de datos</li> <li>c. Paso de datos —túneles y secuencias locales</li> </ul> </li> <li>iii. Estructuras de eventos <ul style="list-style-type: none"> <li>a. Eventos de notificación y eventos de filtrado (interface de usuario)</li> <li>b. Valores de propiedades de controles</li> <li>c. Eventos dinámicos y eventos de usuarios</li> </ul> </li> <li>iv. Nodo de formulas</li> <li>v. Estructuras de deshabilitación de condiciones y de diagrama</li> <li>vi. Estructuras temporizadas</li> <li>vii. Variables locales, globales y compartidas</li> </ul> </li> </ul>

## Certified LabVIEW Associate Developer (CLAD)

### Resumen de certificación y el examen

4. VIs y funciones de programación	<ul style="list-style-type: none"> <li>a. Numéricos, booleanos, cadenas de caracteres, rutas y <i>variants</i></li> <li>b. Conversión, comparación y manipulación</li> <li>c. Arrays y clusters</li> <li>d. Temporización <ul style="list-style-type: none"> <li>i. Funciones: Temporizadores <i>Wait</i>, <i>Tick Count (ms)</i> y <i>Date/Time</i></li> <li>ii. Funciones de temporización relacionadas con estructuras temporizadas</li> </ul> </li> <li>e. Formatos de ficheros: ASCII, binario, datalog, almacenamiento (.tdm), forma de onda, XML y configuración</li> <li>f. Ficheros de E/S de tipo forma de onda</li> <li>g. Eventos dinámicos y de usuario</li> </ul>
5. Comunicación y sincronización de datos	<ul style="list-style-type: none"> <li>a. Variables locales, globales y compartidas</li> <li>b. Datasocket</li> <li>c. TCP y UDP</li> <li>d. Sincronización <ul style="list-style-type: none"> <li>i. Notificadores</li> <li>ii. Colas</li> <li>iii. Semáforos</li> </ul> </li> </ul>
6. Servidor de VIs.	<ul style="list-style-type: none"> <li>a. Configuración del Servidor de Vis</li> <li>b. Jerarquía de clases, referencias, Nodos de Propiedad y Nodos de Invocación</li> <li>c. VIs de carga dinámica</li> </ul>
7. Funciones y Vis para gestión de errores	<ul style="list-style-type: none"> <li>a. Clusters de error</li> <li>b. VIs de dialogo y de interface de usuario</li> <li>c. Códigos de error personalizados</li> </ul>
8. Modelos de diseño	<ul style="list-style-type: none"> <li>a. Máquina de estados simple</li> <li>b. Gestor de eventos de interface de usuario</li> <li>c. Gestor de mensajes en cola</li> <li>d. Productor/consumidor (de datos) y productor/consumidor (de eventos)</li> <li>e. Variables funcionales globales</li> </ul>
9. Diseño SubVI	<ul style="list-style-type: none"> <li>a. Métodos de creación de SubVI</li> <li>b. Cuadro de conexiones y tipos de conexiones</li> <li>c. SubVIs polimórficos</li> <li>d. Opciones relacionados con subVIs</li> <li>e. Gestor de errores</li> </ul>
10. Herramientas y técnicas para depuración Debugging	<ul style="list-style-type: none"> <li>a. Herramientas de depuración (Debugging) <ul style="list-style-type: none"> <li>i. Ventana de lista de errores</li> <li>ii. Ejecución highlighting</li> <li>iii. Puntos de ruptura y ejecución paso a paso</li> <li>iv. Sondas genéricas y personalizadas</li> </ul> </li> <li>b. Prácticas y técnicas de depuración para diversas situaciones</li> </ul>
11. Diseño y documentación de VIs	<ul style="list-style-type: none"> <li>a. Consulte la entrada <i>LabVIEW Style Checklist</i> de la ayuda de LabVIEW para obtener información sobre los siguientes puntos. <ul style="list-style-type: none"> <li>i. Diseño de la interfaz del usuario y del diagrama de bloques.</li> <li>ii. Diseño modular y jerárquico.</li> <li>iii. Diseño de los iconos de SubVIs y del panel de conexiones (estándar).</li> <li>iv. Propiedades de VIs</li> <li>v. Documentación de VIs</li> </ul> </li> </ul>
12. Memoria, rendimiento y determinismo	<ul style="list-style-type: none"> <li>a. Herramientas para identificar los problemas de memoria y de rendimiento.</li> </ul>

## Certified LabVIEW Associate Developer (CLAD)

### Resumen de certificación y el examen

	<ul style="list-style-type: none"><li>i. Perfiles de memoria y rendimiento</li><li>ii. Visualizar la ubicación de buffers</li><li>iii. Métricas del VI</li><li>b. Prácticas de programación<ul style="list-style-type: none"><li>i. Forzar el flujo de datos</li><li>ii. Actualización del interface de usuario y respuesta a los controles del interface de usuario</li><li>iii. Selección del tipo de datos, coerción, y ubicación de buffers</li><li>iv. Operaciones de vectores, cadena de caracteres y bucles</li><li>v. Variables locales y globales, nodos de propiedades y referencias</li></ul></li></ul>
--	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

# **Certified LabVIEW Associate Developer (CLAD)**

## **Resumen de certificación y el examen**

### **Detalles de los Temas del nivel CLAD**

#### **1. Principios de programación LabVIEW:**

- a. Flujo de datos.
  - 1. Definición del flujo de datos.
  - 2. Identificación de la importancia del flujo de datos en LabVIEW.
  - 3. Identificación de las prácticas de programación que refuerzan el flujo de datos en el diagrama de bloques, VIs y subVIs.
  - 4. Identificación de las prácticas de programación que rompen el flujo de datos.
- b. Polimorfismo.
  - 1. Definición de polimorfismo.
  - 2. Identificación de los beneficios del polimorfismo.
  - 3. Determinación de la salida o de los valores intermedios de los elementos de datos en los VIs que utilizan entradas polimórficas.

#### **2. Entorno LabVIEW:**

- a. Ventana del panel frontal, diagrama de bloques y panel de conexiones.
  - 1. Identificación de la relación entre la ventana del panel frontal y los objetos del diagrama de bloques en un VI y sus conexiones a través del panel de conexiones.
  - 2. Identificación de los tipos de VIs que no tienen un diagrama de bloques.
  - 3. Identificación del propósito del panel de conexiones y del icono.
- b. Paletas.
  - 1. Identificación del tipo de paletas y sus funciones.
- c. Opciones de configuración.
  - 1. Identificación del impacto de las opciones de configuración para los siguientes elementos.
    - a) Ventana del panel frontal.
    - b) Diagrama de bloques.
    - c) Entorno.

#### **3. Ventana del panel frontal y objetos del diagrama de bloques, tipos de datos, variables y estructuras software:**

- a. Ventana del panel frontal y objetos del diagrama de bloques:
  - 1. Selección del objeto más adecuado para la ventana del panel frontal de una aplicación.
  - 2. Descripción de la relación entre un objeto de la ventana del panel frontal y sus terminales.
  - 3. Selección entre un objeto de la ventana del panel frontal o una constante del diagrama de bloques.
  - 4. Selección de la paleta (s) más adecuada (s), del (de las) leyenda (s) y del modo de actualización de los gráficos tipo "graph" y tipo "chart"
  - 5. Determinación de la acción mecánica más apropiada para los controles booleanos.
  - 6. Determinación de la idoneidad de los Nodos de Propiedad y de la selección de la propiedad.
- b. Tipos y estructuras de datos.
  - 1. Selección del tipo de dato más apropiado para la ventana del panel frontal y los objetos del diagrama de bloques.
  - 2. Selección del método más adecuado para agrupar datos relacionados.
  - 3. Descripción del tipo de dato forma de onda y su utilización para visualizar los datos en gráficos tipo *graph* y *chart*.
  - 4. Descripción del tipo de dato "timestamp" y su utilización en los datos de medida que requieren estampado de tiempos.

## **Certified LabVIEW Associate Developer (CLAD)**

### **Resumen de certificación y el examen**

5. Identificación de las aplicaciones en las que los tipos de datos “variant” son la opción más apropiada.
- c. Trabajo con los objetos de la ventana del panel frontal y los tipos de datos.
1. Determinación de la representación más adecuada, del rango, del formato, de la precisión y de la escala para la representación de cada elemento de los datos.
  2. Identificación y descripción de los escenarios en los que se necesitará personalizar un control.
  3. Distinción entre la definición de tipo y la definición estricta de tipo.
  4. Identificación y descripción de las aplicaciones que se beneficiarán de la utilización de la definición de tipo y de la definición estricta de tipo.
  5. Determinación de si es necesaria una definición de tipo o de una definición estricta de tipo para representar un elemento de los datos.
- d. Estructuras de control del programa y almacenamiento de datos.
1. Selección y aplicación de la estructura de control del programa más adecuada.
  2. Selección e implementación de un mecanismo de almacenamiento de datos para la estructura de control del programa.
  3. Identificación y descripción de la función de los componentes de una estructura de tipo bucle.
  4. Selección de bucles de tipo “While Loop” o “For Loop” como la estructura más adecuada de bucle.
  5. Descripción de la auto-indexación y determinación de los efectos de la activación o desactivación de la indexación en cada tipo de estructura de bucle.
  6. Determinación de los valores de los datos en un bucle que utiliza la auto-indexación, una vez realizado un cierto número de iteraciones o en el momento de terminación del bucle.
  7. Descripción del uso y de la inicialización del registro de desplazamiento como elemento de almacenamiento de datos.
  8. Determinación de los valores de los datos en el registro o registros de desplazamiento una vez realizado un cierto número de iteraciones o en el momento de terminación del bucle.
  9. Identificación de los pros y los contras y elección entre una estructura de tipo “Sequence” o una estructura de tipo “Case”.
  10. Identificación de las ventajas y desventajas entre las estructuras “Flat” y “Stacked Sequence” con respecto al flujo de datos y la transferencia de datos.
  11. Selección del tipo de dato más apropiado para cablear al terminal del selector de una estructura de tipo “Case”.
  12. Identificación de los dos tipos de túneles de salida en una estructura de tipo “Case” e identificación de las ventajas y desventajas de cada tipo.
  13. Identificación de las ventajas de las estructuras de tipo “Event” para la programación controlada por eventos.
  14. Identificación de los componentes de una estructura de tipo “Event”.
  15. Identificación de las diferentes formas en que se puede generar un evento.
  16. Identificación de los distintos eventos que puede manejar una estructura de tipo “Event”.
  17. Identificación de los dos tipos de eventos de la interfaz del usuario.
  18. Reconocimiento del impacto del bloqueo de la ventana del panel frontal sobre los eventos de la interfaz del usuario.
  19. Identificación y explicación de la necesidad de eventos dinámicos por parte de la aplicación.
  20. Aplicación de técnicas para registrar y anular el registro de eventos dinámicos.
  21. Identificación y explicación de la necesidad por parte de la aplicación de eventos de usuario.
  22. Aplicación de las técnicas para registrar, generar y destruir los eventos de usuario.
  23. Determinación del mecanismo de eventos más apropiado para la aplicación.
  24. Identificación de los componentes de un nodo-fórmula (“Formula Node”) y la relación entre las variables del “script” y los terminales de entrada y salida.
  25. Determinación de la salida de una aplicación que utiliza un nodo-fórmula.



# Certified LabVIEW Associate Developer (CLAD)

## Resumen de certificación y el examen

26. Identificación de la diferencia (s) entre estructuras “Conditional Disable” y “Diagram Disable”
27. Determinación de la salida de una aplicación que utiliza una estructura “Conditional Disable” o “Diagram Disable”.
28. Determinación de si una estructura “Conditional Disable” o “Diagram Disable” es la más apropiada para una aplicación.
29. Selección de la estructura de temporización más adecuada.
30. Identificación y configuración de las entradas y salidas de los componentes de las diferentes estructuras temporizadas.
31. Configuración de la prioridad de las estructuras temporizadas.
32. Determinación de la salida de una aplicación que consta de varias estructuras temporizadas con establecimiento de prioridades.
33. Identificación de la diferencia (s) entre variables compartidas, locales y globales.

### 4. Programación de VIS, funciones y propiedades

#### a. VIs y VIs Express.

1. Determinación de la salida o de los valores intermedios de los elementos de datos en una aplicación que utilice VIs y funciones de la lista de más abajo.
2. Utilización de VIS y de las funciones de la lista de más abajo, determinación del (de los) VI (s) o de la (s) función (es) más apropiadas para completar una determinada funcionalidad.

Lista de los VIs y de las funciones que son apropiadas para esta sección:

- a) Numéricos – Paletas **“Numeric”**, **“Conversion”**, **“Data Manipulation”** y **“Comparisson”**.
- b) Booleanos - Paleta **“Boolean”**.
- c) Cadenas de caracteres – Paletas **“String”**, **“String / Number Conversion”**, y **“String /Array /Path Conversion”**.
- d) Ruta - Funciones de rutas en la paleta **“File I/O”**.
- e) Variant - Funciones de variant en las paletas **“Cluster & Variant”**.
- f) Arrays - Paleta **“Array”**.
- g) Cluster - Funciones de cluster en la paleta de **“Cluster & Variant”**.
- h) Temporización - Paletas **“Timing”** y **“Timed Structures”**.
- i) E/S de fichero - Paletas **“File I/O”** y **“XML”**.
- j) Formas de onda - Paleta **“Waveform”**.
- k) Eventos - Paleta **“Events”**.

### 5. Comunicación y sincronización de datos de VIs y funciones

#### a. Funciones, VIs y VIs Express.

1. Selección del método más apropiado para la comunicación de los datos entre aplicaciones.
2. Identificación de las ventajas y desventajas de la utilización de variables locales, globales y compartidas en la comunicación de datos.
3. Identificación y explicación de los diferentes métodos de comunicación a través de TCP y UDP.
4. Identificación y explicación de la (s) diferencia (s) entre los notificadores y las colas.
5. Selección del método más apropiado para comunicar datos entre las múltiples secciones del diagrama de bloques.
6. Identificación de las aplicaciones que utilizan semáforos para la sincronización y protección de datos.
7. Determinación de la salida o de los valores intermedios de los elementos de datos en una aplicación que utiliza VIs y funciones de la lista de más abajo.
8. Utilización de los VIs y de las funciones de la lista de más abajo para determinar el (los) VI (s) o la (s) función (es) más apropiadas que se necesitan para completar una determinada funcionalidad.

Lista de VIs y funciones que se aplican en esta sección:

- a) Datasocket – Paleta **“Datasocket”**

# Certified LabVIEW Associate Developer (CLAD)

## Resumen de certificación y el examen

- b) TCP y UDP - Paletas “TCP y UDP”
- c) Notificadores - Paleta “Notifier Operations”
- d) Colas – Paleta “Queue Operations”
- e) Semáforos - Paleta “Semaphore”

### 6. Servidor de VIs

- a. Configuración.
  - 1. Aplicación de los ajustes apropiados para configurar el Servidor de VIs.
- b. Jerarquía de clases, referencias, Nodos de Propiedades, Nodos de Invocación y VIs de carga dinámica.
  - 1. Identificación de los diferentes métodos de carga y ejecución dinámica de VIs, el tipo de referencia, las propiedades y los métodos utilizados para soportar el método.
  - 2. Dada una jerarquía de clase, reconocimiento de la herencia de propiedades y métodos y uso de los VIs de conversión de tipo de referencia con el fin de obtener una referencia para una clase superior o inferior en la jerarquía.
  - 3. Determinación del método más adecuado para la carga dinámica y el funcionamiento de VIs.

### 7. VIs y Funciones de gestión de errores

- a. Clusters y cables de error.
  - 1. Identificación de los componentes de los clusters y de los terminales de error que acepten cables de error.
  - 2. Identificación de las diferencias entre errores y warnings.
  - 3. Diseño de VIs que sigan los principios indicados en la sección “LabVIEW Style Checklist” de la ayuda de LabVIEW. Por ejemplo, utilización de la comprobación de errores para el control de lazos “While”, gestión de errores con estructuras “Case” y utilización de los terminales correspondientes en el panel de conexiones.
- b. VIs de errores y diálogos.
  - 1. Dado un VI o subVI, identificación de las ubicaciones más adecuadas para la gestión de errores y notificar al usuario o hacer una llamada al VI del error.
  - 2. Utilizando los VIs y las funciones de la paleta “Dialog & User Interface”, determinar el (los) VI (s) o función (es) más apropiadas para llevar a cabo la gestión de errores y la funcionalidad de presentación de informes especificadas.
- c. Códigos de error personalizados.
  - 1. Identificación del rango numérico y de los métodos para definir los códigos de error personalizados y utilización de los códigos de error personalizados para generar errores desde VIs.

### 8. Patrones de diseño

- a. Selección de un patrón de diseño:
  - 1. Identificación de un patrón de diseño, explicación de las ventajas y desventajas y comparación con otros patrones de diseño.
  - 2. Dados los requisitos de una aplicación, seleccionar el patrón de diseño más adecuado entre las siguientes opciones:
    - a) Máquina de estados sencilla.
    - b) Gestor de eventos de la interfaz de usuario.
    - c) Gestor de mensajes encolados.
    - d) Productor/consumidor (datos).
    - e) Productor/consumidor (eventos).
    - f) Variable global funcional.

### 9. Diseño de SubVIs

- a. Métodos para crear subVIs.
  - 1. Identificación y explicación de los métodos utilizados para crear subVIs y los pros y los contras de cada método.

# **Certified LabVIEW Associate Developer (CLAD)**

## **Resumen de certificación y el examen**

- b. Panel de conexiones y tipos de conexión.
  - 1. Selección del panel de conexiones más adecuado y asignación de los terminales de acuerdo a las recomendaciones de la sección “*LabVIEW Style Checklist*” de la ayuda de *LabVIEW*.
  - 2. Identificación de los terminales que son requeridos, recomendados u opcionales.
  - 3. Dado un requisito, identificación de los terminales a establecer como conexiones obligatorias, recomendadas u opcionales.
- c. SubVIs polimórficos.
  - 1. Evaluación de si el diseño de un subVI polimórfico es la opción más apropiada.
  - 2. Identificación de los pros, contras y de las restricciones en el desarrollo de un subVI polimórfico.
- d. Opciones relacionadas con SubVIs.
  - 1. Identificación de la configuración de la ventana y de la ejecución y explicación de las implicaciones de cada configuración.
  - 2. Selección y aplicación de la configuración de la ventana y de la ejecución más adecuada para uno solo o para todos los subVI.
- e. Gestión de errores.
  - 1. Aplicación del manejo de errores a un subVI de acuerdo a las recomendaciones de la sección “*LabVIEW Style Checklist*” en la ayuda de *LabVIEW*.

### **10. Técnicas y herramientas de depuración**

- a. Herramientas de depuración.
  - 1. Identificación y explicación de las implicaciones que tienen los ajustes de las propiedades de los VIs en la forma en que LabVIEW gestiona los errores y los “warnings”.
  - 2. Identificación de los errores de un VI que resultan en una ruptura del botón de ejecución “**Run**” y utilización de la ventana “**Error list**” para determinar la causa.
  - 3. Explicación del uso del resaltado de la ejecución para el rastreo del flujo de datos como herramienta de ayuda en conjunto con otras herramientas de depuración.
  - 4. Explicación y aplicación de las interrupciones, de la suspensión de ejecución y de las herramientas paso a paso para depurar subVIs y VIs.
  - 5. Utilización de la herramienta Sonda (“Probe”), de los indicadores, de las sondas genéricas, de las sondas condicionales y de las sondas personalizadas para mostrar los valores de los datos.
- b. Prácticas de depuración y técnicas para diferentes situaciones.
  - 1. Dada una situación de error, selección del método más apropiado para depurar el error.
  - 2. Determinación de si un determinado diagrama de bloques se traduciría en una condición de error.

### **11. Diseño y documentación de VIs**

- a. Utilización y aplicación de la sección “*LabVIEW Style Checklist*” de la ayuda de *LabVIEW* para lo siguiente:
  - 1. Diseño de la interfaz del usuario y del diagrama de bloques.
  - 2. Diseño modular y jerárquico.
  - 3. Diseño de paneles de conexión y de iconos de SubVI (estándar).
  - 4. Propiedades de VIs.
  - 5. Documentación de VIs.

### **12. Memoria, rendimiento y determinismo**

- a. Herramientas para la identificación de los problemas de memoria y de rendimiento.
  - 1. Selección de la (s) herramienta (s) más apropiada (s) para la identificación de los problemas de memoria y del rendimiento.
- b. Prácticas de programación.
  - 1. Identificación del código del diagrama de bloques que rompe el flujo de datos y aplicación de las técnicas que refuerzan el flujo de datos.

## **Certified LabVIEW Associate Developer (CLAD)**

### **Resumen de certificación y el examen**

2. Identificación del código del diagrama de bloques que puede ralentizar la respuesta de la interfaz de usuario o actualización e identificación de las técnicas para mejorar la respuesta.
3. Selección del tipo de datos más adecuado para limitar las conversiones de datos ("coerción"), las reservas de buffers y para optimizar la velocidad y la reutilización de la memoria.
4. Identificación de las operaciones de arrays, cadenas de caracteres y bucles que limitan la memoria y el rendimiento e identificación de las técnicas y métodos que optimicen el rendimiento.
5. Identificación de los problemas debidos a las condiciones de carrera, de memoria y de rendimiento asociados con el uso de variables globales y locales, Nodos de Propiedad y referencias y la utilización de técnicas para la optimización de su empleo.

# Certified LabVIEW Associate Developer (CLAD)

## Resumen de certificación y el examen

### Recursos para la preparación del examen de nivel CLAD

Utilice los siguientes recursos para la preparación del examen:

- [Seminario web de preparación para nivel CLAD de National Instruments \(en línea\)](#)
- [Examen de muestra de nivel CLAD](#)

Formación/Tutoriales:

- [Fundamentos de LabVIEW en línea](#)
- [Curso de introducción a LabVIEW – Tres horas](#)
- [Curso de introducción a LabVIEW – Seis horas](#)
- [Cursos básicos de LabVIEW I y LabVIEW I de National Instruments:](#)
  - Dirigido por un instructor
  - Autoguiado utilizando los manuales del curso
- [Cursos intermedios de LabVIEW I y II National Instruments:](#)
  - Dirigido por un instructor
  - Autoguiado utilizando los manuales del curso
- [Curso de \*LabVIEW Upgrade Primer\* \(en línea\) National Instruments](#)

Recursos en la Web:

- [Directrices de desarrollo de LabVIEW](#)
- [Práctica gratuita - Examen de fundamentos de LabVIEW](#)
- [Web universitaria de National Instruments](#)
- [Zona de desarrollo de National Instruments](#)
- [Zona del desarrollador de LabVIEW de National Instruments](#)
- [Zona de LabVIEW de National Instruments](#)
- [Soporte de LabVIEW de National Instruments](#)
- [Manuales en línea de LabVIEW \(manuales actuales\)](#)
- [Cursos guiados gratuitos:](#)
  - [Revisión de LabVIEW](#) (Colorado School of Mines)
  - [Curso guiado de LabVIEW](#) (University of Sydney)
  - [LabVIEW para Dummies®](#) (Illinois Institute of Technology)
  - [Curso guiado de LabVIEW](#) (University of Buffalo)
  - [Serie de cursos guiados de LabVIEW](#) (University of Western Australia)