

Informações Gerais sobre a Certificação

O programa de certificação em LabVIEW da National Instruments consiste dos seguintes três níveis de certificação:

- Certified LabVIEW Associate Developer (CLAD)
- Certified LabVIEW Developer (CLD)
- Certified LabVIEW Architect (CLA)

Cada nível é um pré-requisito para o nível de certificação seguinte.

O profissional que possui a certificação CLAD demonstra entendimento amplo e completo das mais importantes características e funcionalidades disponíveis no LabVIEW Full Development System. Possui a habilidade de aplicar esse conhecimento para desenvolver, depurar e manter pequenos módulos em LabVIEW. O nível de experiência típico de um CLAD é de aproximadamente 6 a 9 meses de uso do LabVIEW Full Development System.

Um profissional com a certificação CLD demonstra experiência em desenvolvimento, depuração, organização e manutenção de médias e grandes aplicações em LabVIEW. Um CLD é um profissional com uma experiência cumulativa aproximada de 12 a 18 meses desenvolvendo médias e grandes aplicações no LabVIEW.

O profissional com a certificação CLA demonstra pleno conhecimento na arquitetura de aplicações LabVIEW para projetos envolvendo múltiplos desenvolvedores. Um CLA não somente possui a capacidade técnica e experiência em desenvolvimento de software, necessária para quebrar uma especificação de projeto em componentes do LabVIEW, como a experiência exigida para compreender o projeto em sua totalidade através do uso de ferramentas de gerenciamento de projeto e configurações. Um CLA é um profissional com uma experiência cumulativa aproximada de 24 meses desenvolvendo médias e grandes aplicações em LabVIEW.



Nota – A certificação CLAD é um pré-requisito para prestar o exame CLD. A certificação CLD é um pré-requisito para prestar o exame CLA. Não há exceções para este requisito em cada exame.

Informações Gerais sobre o Exame CLAD

Produto: LabVIEW Full Development System versão 8.0 para Windows. Acesse

ni.com/labview/how_to_buy.htm para obter detalhes das funcionalidades disponíveis no LabVIEW Full Development System.

Duração do Exame: 1 hora

Número de questões: 40

Estilo das Questões: múltipla escolha

Nota de aprovação: 70%

O exame valida conhecimento em aplicação e não a habilidade de rever passos de menu ou nomes de VIs e componentes.

O uso do LabVIEW ou quaisquer outros recursos externos é proibido durante o exame. Para assistência e quando apropriado, telas do *LabVIEW Help* são fornecidas durante o exame.

Para manter a integridade do exame, você não pode copiar ou reproduzir qualquer seção do mesmo. Falha no cumprimento resultará em reprovação. Em localidades onde o teste é aplicado como um exame em papel, tirar o grampo que une as folhas resultará em reprovação sem avaliação.

Logística do Exame

Estados Unidos e Europa: O exame CLAD pode ser prestado nos centros de teste da Pearson Vue. O exame é baseado em computador e os resultados ficam disponíveis imediatamente após o término do exame. Vá a www.pearsonvue.com para mais detalhes bem como para agendar o teste.

Ásia: O exame é em papel para o qual as avaliações e resultados levam aproximadamente quatro semanas. Por gentileza, contate seu escritório local da National Instruments para detalhes e agendamento de data.

Para questões gerais ou comentários, envie um e-mail para: certification@ni.com

Tópicos do Exame

1. Conceitos de programação em LabVIEW
2. Ambiente LabVIEW
3. Construção de software em LabVIEW
4. Programar VIs e funções
5. VIs e funções de Comunicação de Dados e sincronização
6. VIs e funções do VI Server
7. VIs e funções de gerenciamento de erro
8. Padrões de projeto
9. Projeto de SubVI
10. Ferramentas e técnicas de depuração
11. Práticas de projeto de VI e documentação (estilo)
12. Memória, desempenho e determinismo

Tópicos do Exame (Detalhado)

Tópico	Sub Tópico
1. Princípios de Programação em LabVIEW	<ul style="list-style-type: none"> a. Fluxo de Dados b. Polimorfismo
2. Ambiente LabVIEW	<ul style="list-style-type: none"> a. Janela de painel frontal, diagrama de blocos, connector pane b. Menus e paletas c. Opções de configuração
3. Construção de Software em LabVIEW	<ul style="list-style-type: none"> a. Janela do painel frontal e objetos do diagrama de blocos <ul style="list-style-type: none"> i. Controles, indicadores, controles de ES, e referências ii. Terminais, constantes e nós iii. Paletas, modos de atualização e legendas de gráficos iv. Ação mecânica de objetos Booleanos v. Property nodes b. Tipos de dados e estruturas de dados <ul style="list-style-type: none"> i. Tipos de dado numérico, string, Booleano e Path ii. Tipos de dado de array e cluster iii. Tipos de dado de waveform e timestamp iv. Tipos de dados variant c. Trabalhando com objetos e tipos de dados na janela do painel frontal <ul style="list-style-type: none"> i. Faixas, formatos, representação e escalas ii. Personalizando controles iii. Type definitions e strict type definitions d. Programar estruturas de controle e armazenamento de dados

	<ul style="list-style-type: none">i. Estruturas de looping (Loops For e Loops While)<ul style="list-style-type: none">a. Indexando em bordas de loopsb. Shift registersii. Estruturas de Case e Sequence<ul style="list-style-type: none">a. Estruturas Sequence Flat e Stackedb. Valores de seletor de case e tipos de dadosc. Passando dados – túneis e sequence localsiii. Estruturas Event<ul style="list-style-type: none">a. Eventos notify e filter (interface de usuário)b. Propriedade Value (Signaling) de controlesc. Eventos dinâmicos e User Eventsiv. Formula Nodev. Estruturas Conditional Disable e Diagram Disablevi. Estruturas Temporizadasvii. Variáveis locais, globais e compartilhadas
4. Programando VIs e funções	<ul style="list-style-type: none">a. Numérico, Booleano, string, Path e Variantb. Conversão, comparação e manipulaçãoc. Arrays e clustersd. Temporização<ul style="list-style-type: none">i. Funções Wait, Tick Count (ms), e Date/Timeii. Funções de temporização relacionadas com estruturas temporizadasd. Formatos de arquivo ASCII, binário, datalog,

	<p>storage(.tdm), waveform, XML, e configuration</p> <p>e. em formato E/S</p> <p>f. Waveform e E/S de arquivo waveform</p> <p>g. Eventos Dinâmico e User Events</p>
5. Comunicação de dados, Sincronização	<p>a. Variáveis locais, globais e compartilhadas</p> <p>b. DataSocket</p> <p>c. TCP e UDP</p> <p>d. Sincronização</p> <p>i. Notifiers</p> <p>ii. Queues</p> <p>iii. Semaphores</p>
6. VI Server	<p>a. Configurando o VI Server</p> <p>b. Hierarquia de classes, referências, Property Nodes, Invoke Nodes</p> <p>c. Carregando VIs dinamicamente</p>
7. VIs e funções de gerenciamento de erro	<p>a. Clusters de erro</p> <p>b. VIs de Diálogo e Interface de Usuário</p> <p>c. Códigos de erro personalizados</p>
8. Padrões de projeto	<p>a. Máquinas de estados simples</p> <p>b. User Interface Event Handler</p> <p>c. Queued message handler</p> <p>d. produtor/consumidor (dados) e produtor/consumidor (eventos)</p> <p>e. Variáveis globais funcionais</p>
9. Projeto de SubVIs	<p>a. Métodos de criação de SubVI</p>

	<ul style="list-style-type: none"> b. Connector panes e tipos de conexão c. SubVIs polimórficos d. Opções relacionadas com SubVIs e. Gerenciamento de erro
10. Ferramentas e técnicas de depuração	<ul style="list-style-type: none"> a. Ferramentas de Depuração <ul style="list-style-type: none"> i. janela Error list ii. execução destacada iii breakpoints e execução passo-a-passo iv. probes genéricos e personalizados b. Práticas e técnicas de depuração para diferentes situações
11. projeto e documentação de VI	<ul style="list-style-type: none"> a. Vá ao tópico <i>LabVIEW Style Checklist</i> do <i>LabVIEW Help</i> para informação sobre os seguintes itens: <ul style="list-style-type: none"> i. projeto da interface de usuário e layout do diagrama de blocos ii. projeto modular e hierárquico iii. Ícones de SubVI e layout de connector pane (padrão) iv. propriedades do VI v. documentando VIs
12. Memória, Desempenho e Determinismo	<ul style="list-style-type: none"> a. Ferramentas para identificar problemas de memória e desempenho <ul style="list-style-type: none"> i. Profile, Performance and Memory ii. Show Buffer allocations iii. VI metrics b. Práticas de Programação <ul style="list-style-type: none"> i. Garantindo o fluxo de dados ii. atualização da interface de usuário e resposta a

Certified LabVIEW Associate Developer (CLAD)

Informações sobre o Exame e a Certificação

	<p>controles da interface de usuário</p> <p>iii. seleção de tipos de dados, coerção, e alocações de buffer</p> <p>iv. Operações com array, string e operações de loop</p> <p>v. Variáveis locais e globais, Property Nodes e referências</p>
--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

CLAD – Detalhes dos Tópicos

1. Princípios de Programação em LabVIEW

a. Fluxo de Dados

1. Defina fluxo de dados
2. Identifique a importância do fluxo de dados no LabVIEW
3. Identifique práticas de programação que impõem fluxo de dados ao diagrama de blocos, VIs e subVIs

b. Polimorfismo

1. Defina polimorfismo
2. Identifique os benefícios do polimorfismo
3. Determine os valores de saída ou intermediários dos elementos de dados em um VI que utiliza entradas polimórficas

2. O Ambiente LabVIEW

a. Janela do Painel Frontal, diagrama de blocos e connector pane

1. Identifique a relação entre a janela do painel frontal e os objetos do diagrama de blocos em um VI bem como suas conexões através do connector pane
2. Identifique que tipos de VIs não tem um diagrama de blocos
3. Identifique o propósito do connector pane e do ícone

b. Paletas

1. Identifique o tipo de paleta e suas funções

c. Opções de Configuração

1. identifique o impacto das opções de configuração para os seguintes itens:
 - a) janela de painel frontal
 - b) diagrama de blocos
 - c) ambiente

3. Janela do Painel Frontal e objetos do diagrama de blocos, tipos de dados, variáveis e construções de software:

a. Janela do painel frontal e objetos do diagrama de blocos

1. Selecione o objeto mais apropriado para a janela do painel frontal de uma aplicação
2. Descreva a conexão entre um objeto na janela do painel frontal e seus terminais
3. Escolha entre um objeto na janela do painel frontal ou uma constante no diagrama de blocos.
4. Selecione a paleta(s) mais apropriada(s), legenda(s) e modo de atualização para gráficos
5. Determine a ação mecânica mais apropriada para controles Booleanos
6. Determine a necessidade de uso de Property Nodes e selecione a propriedade adequada.

b. Tipos de Dados e estruturas de dados

1. Selecione os tipos de dados mais apropriados para a janela do painel frontal e objetos do diagrama de blocos
2. Selecione o método mais apropriado para agrupar itens de dados relacionados
3. Descreva o tipo de dado waveform e use-o para exibir dados em gráficos
4. Descreva o tipo de dado timestamp e use-o para definir a escala de tempo de dados de medição
5. Identifique as aplicações nas quais o tipo de dado Variant é a escolha mais apropriada.

c. Trabalhando com objetos e tipos de dados da janela do painel frontal

1. Determine a representação, faixa, formato, precisão e escala mais apropriada para representar um item de dado
2. Identifique e descreva os cenários nos quais há necessidade de personalizar um controle
3. Diferencie um type definition de um strict type definition
4. Identifique e descreva as aplicações que poderiam beneficiar-se do uso de um type definition ou um strict type definition
5. Determine se um type definition ou um strict type definition é necessário para representar um item de dado

d. Estruturas para controle de programas e armazenamento de dados

1. Selecione e aplique a estrutura para controle de programa mais adequada
2. Selecione e implemente um mecanismo de armazenamento de dados para uma estrutura para controle de programa
3. Identifique e descreva a função de estruturas de looping
4. Selecione um Loop While ou Loop For como a estrutura de looping mais apropriada
5. Descreva auto-indexação e determine os efeitos de habilitar ou desabilitar a indexação com cada tipo de estrutura de looping
6. Determine os valores de dados em um loop que utilize auto-indexação, após um número definido de iterações ocorrer ou após a finalização de um loop
7. Descreva o uso e a inicialização de shift registers como elementos de armazenamento de dados
8. Determine os valores de dados no shift register(s) após um número definido de iterações ocorrer ou após a finalização do loop
9. Identifique os prós e contras e selecione entre uma estrutura Sequence ou estrutura Case
10. Identifique os prós e contras entre estruturas Sequence Flat e Stacked em relação ao fluxo de dados e passagem de informações dentro do programa
11. Selecione os tipos de dados mais apropriados para conectar ao terminal seletor de uma estrutura Case
12. Identifique dois tipos de túneis de saída em uma estrutura Case e identifique os prós e contras de cada tipo
13. Identifique as vantagens das estruturas de Eventos para programação orientada a eventos
14. Identifique os componentes de uma Event Structure
15. Identifique as diferentes formas nas quais um evento pode ser gerado
16. Identifique os diferentes eventos que uma estrutura de eventos pode gerenciar
17. Identifique os dois tipos de eventos de interface de usuário
18. Reconheça o impacto de travar a janela do painel frontal para eventos de interface de usuário
19. Identifique e explique a necessidade de uma aplicação usar eventos dinâmicos
20. Aplique as técnicas para registrar e desregistrar eventos dinâmicos
21. Identifique e explique a necessidade de uma aplicação usar User Events

22. Aplique as técnicas para registrar, gerar, e destruir User Events
23. Determine o mecanismo de evento mais apropriado para uma aplicação
24. Identifique os componentes de um Formula Node e a relação entre as variáveis declaradas no script e os terminais de entrada e saída.
25. Determine a saída de uma aplicação que utiliza um Formula Node
26. Identifique as diferenças entre estruturas Conditional Disable e Diagram Disable
27. Determine a saída de uma aplicação que utiliza uma estrutura Conditional Disable e Diagram Disable
28. Determine se a estrutura Conditional Disable ou a Diagram Disable é mais apropriada para uma aplicação
29. Selecione a estrutura Temporizada (Time Structures) mais apropriada
30. Identifique e configure a entrada e saídas dos componentes de diferentes estruturas Temporizadas
31. Configure a prioridade de estruturas Temporizadas
32. Determine a saída de uma aplicação que consiste de múltiplas estruturas Temporizadas com prioridades definidas
33. Identifique a(s) diferença(s) entre variáveis compartilhadas, locais e globais

4. Programando VIs, funções e propriedades

a. VIs e VIs Expressos

1. Determine os valores de saída ou intermediário de elementos de dados em uma aplicação que utiliza VIs e funções da lista a seguir
2. Utilizando VIs e funções da lista a seguir, determine o(s) VI(s) e função (s) apropriadas para implementar uma função específica

Lista dos VIs e funções desta seção:

- a) Numérico – paletas **Numeric, Conversion, Data Manipulation e Comparison**
- b) Booleano – paleta **Boolean**
- c) String – paletas **String, String / Number Conversion, e String / Array / Path Conversion**
- d) Diretório – Funções de Path na paleta **File I/O**

- e) Variants – Funções de Variants na paleta **Cluster & Variant**
- f) Array – paleta **Array**
- g) Cluster – Funções de Cluster na paleta **Cluster & Variant**
- h) Temporização – paletas **Timing** e **Timed Structures**
- i) E/S de Arquivo – paletas **File I/O** e **XML**
- j) Waveform – paleta **Waveform**
- k) Eventos – paleta **Events**

4.VIs e funções de Comunicação de Dados e Sincronização

a. Funções, VIs e VIs Expressos

1. Selecione o método mais apropriado para passar dados entre aplicações
2. Identifique os prós e contras do uso de variáveis locais, globais ou compartilhadas para comunicação de dados
3. Identifique e explique os diferentes métodos de comunicação usando TCP e UDP
4. Identifique e explique as diferenças entre notifiers e queues
5. Selecione o método mais apropriado para comunicar dados entre seções de múltiplos diagramas de blocos
6. Identifique as aplicações que usam semáforos para proteção de dados e sincronização
7. Determine os valores de saída ou intermediário de elementos de dados de uma aplicação que utiliza VIs e funções da lista a seguir
8. Utilizando VIs e funções da lista a seguir, determine o VI (s) ou Função(s) mais apropriado para implementar uma função específica

Lista dos VIs e funções desta seção:

- a) DataSocket – paleta **DataSocket**
- b) TCP e UDP – paletas **TCP** e **UDP**
- c) Notificadores – paleta **Notifier Operations**
- d) Queues – paleta **Queue Operations**
- e) Semáforos – paleta **Semaphore**

6. VI Server

a. Configuração

1. Aplique definições apropriadas para configurar o VI Server

b. Hierarquia de classe, referências, Property Nodes, Invoke Nodes e carregamento dinâmico de VIs

1. Identifique os diferentes métodos de carregar e executar dinamicamente VIs, o tipo de referência, propriedades e métodos usados para suportar o método

2. Dada uma classe da hierarquia, reconheça as propriedades e métodos herdados e use VIs de conversão de tipo para obter uma referência de uma classe mais alta ou mais baixa na hierarquia.

3. Determine o método mais apropriado para carregar e executar VIs dinamicamente

7. Funções e VIs de gerenciamento de erro

a. Clusters de erro e conexões

1. Identifique os componentes de clusters de erro e terminais que aceitam a conexão de erro

2. Identifique as diferenças entre erros e alertas (warnings)

3. Projete VIs que obedecem ao tópico *LabVIEW Style Checklist* do *LabVIEW Help*. Por exemplo, utilize verificação de erro para controlar Loops While, gerencie erros com estruturas Case, e use terminais apropriados no connector pane.

b. VIs de Erro e Diálogo

1. Dado um VI ou subVI, identifique os locais mais apropriados para gerenciar erros e notificar o usuário ou um VI de mais alto nível do erro

2. Utilizando VIs e funções da paleta **Dialog & User Interface**, determine os VIs ou função(s) para gerenciar e reportar erros como especificado

c. Códigos de erro personalizados

1. Identifique a faixa numérica e métodos para definir códigos de erro personalizados e use os códigos de erro personalizados para gerar erros a partir dos VIs.

8. Padrões de Projeto

a. Selecione um padrão de projeto:

1. Identifique um padrão de projeto, explique seus prós e contras, e compare-o com outros padrões de projeto

2. Dados os requisitos de uma aplicação, selecione o padrão de projeto mais apropriado a partir dos seguintes:

- a) Máquina de estados simples
- b) User interface event handler
- c) Queued message handler
- d) Produtor/consumidor (dados)
- e) Produtor/consumidor (eventos)
- f) Variável global funcional

9. Projeto de SubVI

a. Métodos para criar subVIs

1. Identifique e explique os métodos usados para criar subVIs e os prós e contras de cada método

b. Connector Pane e tipos de conexão

1. Selecione o connector pane mais apropriado e designe terminais de acordo com as recomendações do tópico *LabVIEW Style Checklist* do *LabVIEW Help*

2. Identifique quais terminais são Required, Recommended ou Optional

3. Dado um requisito, identifique quais terminais definir como conexões Required, Recommended ou Optional

c. SubVIs Polimórficos

1. Avalie se um projeto de subVI polimórfico é a escolha mais apropriada

2. Identifique os prós, contras e restrições no desenvolvimento de um subVI polimórfico

d. Opções relacionadas com SubVIs

1. Identifique a execução e definição de janela bem como explique a implicação de cada definição

2. Selecione e aplique as configurações de execução e de janela mais apropriada para todas ou para uma única instância de um subVI

e. Gerenciamento de Erro

1. Aplique gerenciamento de erro a um subVI como recomendado no tópico *LabVIEW Style Checklist* do *LabVIEW Help*

10. Ferramentas e técnicas de depuração

a. Ferramentas de Depuração

1. Identifique e explique as implicações das configurações do VI Properties que determinam como o LabVIEW gerencia erros e alertas
2. Identifique os erros em um VI que resultam no botão **Run** e use a janela **Error List** para determinar a causa
3. Explique o uso de execução destacada para rastrear o fluxo de dados e como uma ferramenta de auxílio em conjunto com outras ferramentas de depuração
4. Explique e utilize breakpoints, suspensão da execução, e ferramenta de execução passo-a-passo para depurar VIs e subVIs.
5. Utilize a ferramenta de Probe, indicadores, probes genéricos, probes condicionais e probes personalizados para exibir valores de dados.

b. Técnicas e Práticas de depuração para diferentes situações

1. Dada uma situação de erro, selecione o método mais apropriado para depurar o erro
2. Determine se um diagrama de blocos dado poderia resultar em uma condição de erro

11. Projeto de documentação de VIs

a. Utilize e aplique o tópico *LabVIEW Style Checklist* do *LabVIEW Help* para os seguintes:

1. Projeto da interface de usuário e layout do diagrama de blocos
2. Projeto modular e hierárquico
3. Ícones de subVI e layout de connector pane (padrão)
4. VI Properties
5. Documentando VIs

12. Memória, desempenho e determinismo

a. Ferramentas para identificar problemas de memória e desempenho

1. Selecione a(s) ferramenta(s) mais apropriada(s) para identificar problemas de memória e desempenho

b. Práticas de programação

1. Identifique o código no diagrama de blocos que quebra o fluxo de dados e técnicas para impor o fluxo de dados

2. Identifique código no diagrama de blocos que pode diminuir a velocidade da resposta e atualização da interface de usuário e identifique técnicas para melhorar a resposta
3. Selecione o tipo de dados mais apropriado para limitar coerção, alocações de buffer e otimizar velocidade e reuso de memória
4. Identifique operações de array, string e loop que limitam memória e desempenho e identifique técnicas e métodos que melhoram o desempenho
5. Identifique race conditions, problemas de memória e desempenho associados ao uso de variáveis locais e globais, Property Nodes, e referências, e use técnicas para aperfeiçoar seu uso

Recursos de Preparação do CLAD

Use os seguintes recursos para preparação para o exame:

Curso de Preparação do CLAD:

- [National Instruments CLAD Preparation Course \(Online\) webcast](#)

Exame amostra do CLAD

- [CLAD Sample Exam](#)

Treinamento / Tutoriais

- [Online LabVIEW Basics](#)
- [LabVIEW Introduction Course – 3 horas](#)
- [LabVIEW Introduction Course – 6 horas](#)
- [National Instruments LabVIEW Basics I](#) e [LabVIEW Basics II](#) cursos:
 - Ministrado por instrutor
 - Individual utilizando apostilas do curso
- [National Instruments LabVIEW Intermediate I](#) e [Intermediate II](#) cursos:
 - Ministrado por instrutor
 - Individual utilizando apostilas do curso

Recursos na Web

- [LabVIEW Development Guidelines](#)
- Prática livre [LabVIEW Fundamentals Exam](#)
- [National Instruments Academic Web](#)
- [National Instruments Developer Zone](#)
- [National Instruments Developer \(LabVIEW\) Zone](#)
- [National Instruments LabVIEW Zone](#)
- [National Instruments LabVIEW Support](#)
- [LabVIEW Manuals Online](#) (manuais atuais)
- Tutoriais Livres
 - [LabVIEW Tutorial](#) (University of Sydney)
 - [LabVIEW for Dummies](#)® (Illinois Institute of Technology)
 - [LabVIEW Tutorial](#) (University of Buffalo)
 - [LabVIEW Tutorial Series](#) (University of Western Australia)