

The background is an abstract composition of dynamic, flowing lines. The upper portion is dominated by deep blues and greens, while the lower portion transitions into vibrant reds and oranges. The lines create a sense of motion and depth, resembling a high-speed light trail or a stylized representation of a celestial or technological environment.

**National Instruments**

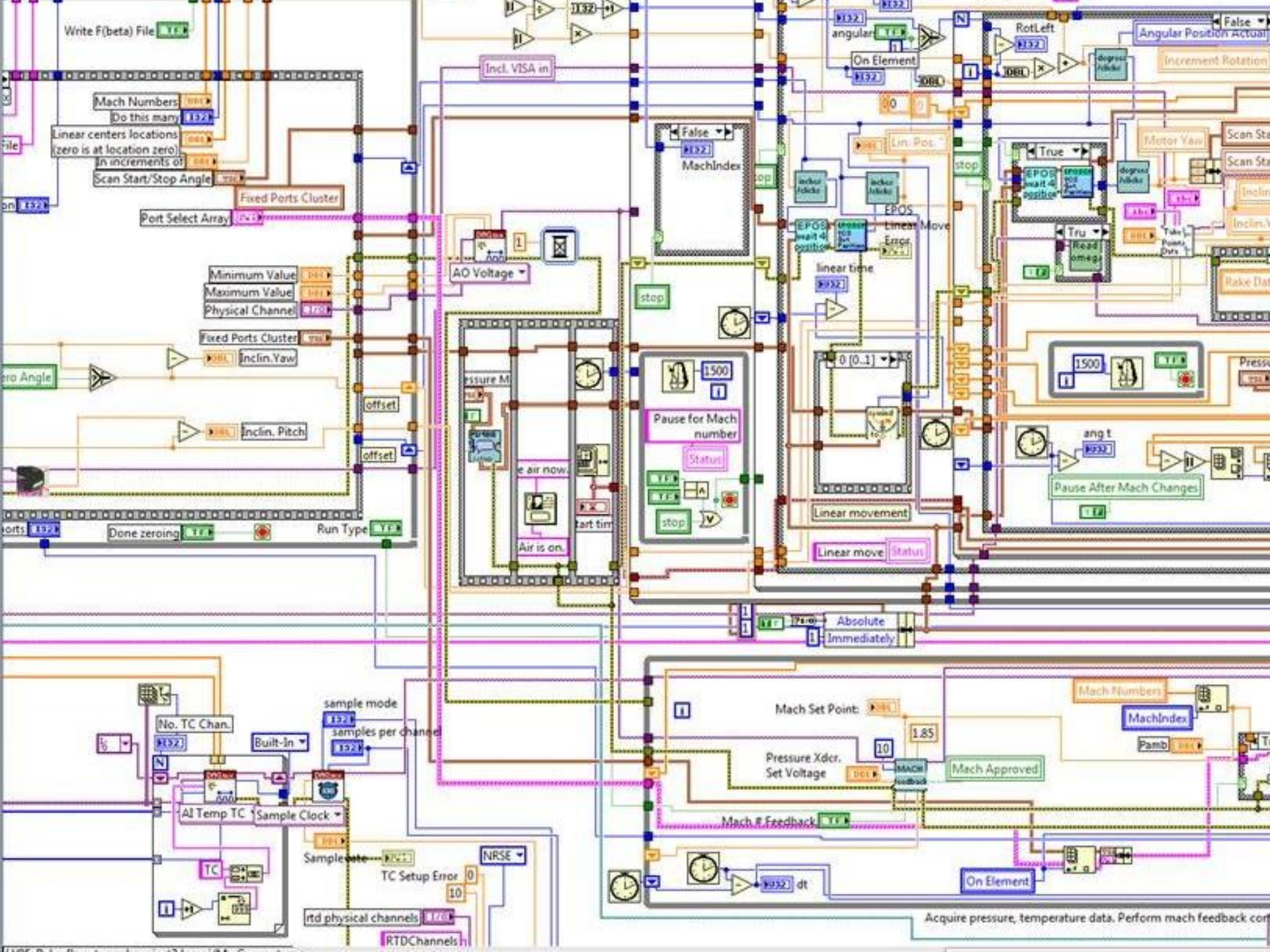
**Aerospace & Defense Day 2013**

# Técnicas avançadas para desenvolvimento de aplicações no LabVIEW

Ilton Pereira

Gerente da Engenharia de Aplicações



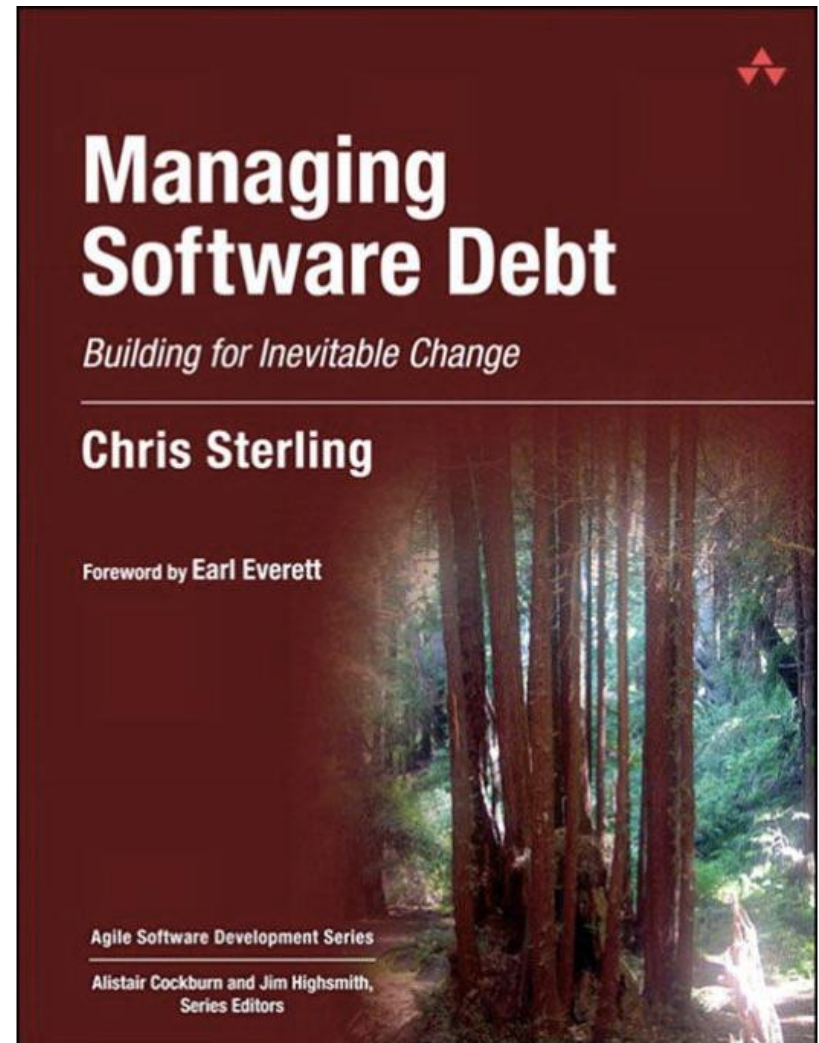


É uma questão de economizar tempo

# Tornando-se um arquiteto

“O desenvolvimento de software exige conhecimento e experiência. As pessoas não adquirem esse conhecimento e essa experiência sem passar anos desenvolvendo software... A cada dia nós nos tornamos um pouco mais proficientes em uma área funcional específica”

Chris Sterling - *Managing Software Debt*





# Você já ouviu isso alguma vez?

- Adicionar ou alterar suporte à hardware requer tempo e esforço significativos.
- Para adicionar uma funcionalidade similar a uma já existente, você constantemente copia trechos do código.
- Ao adicionar uma nova funcionalidade, o código começa a apresentar falhas devido a um mau projeto.



Simples – Escalável – Reutilizável

# O objetivo do projeto orientado a objeto

Organizar o software, de modo que:

- Os bugs sejam fáceis de isolar e
- Novos recursos sejam fáceis de adicionar



# Classe: Carro

Exemplos de objetos dessa classe



Propriedades desses itens:

- Consumo (km/l)
- Índice de segurança
- Câmbio manual/automático
- Motor
- Potência
- Combustível

Propriedades  
(atributos)

Esses itens podem...

- Ligar
- Dirigir
- Trocar marcha
- Acionar os *Air Bags*
- **Viajar no tempo**

Métodos



# O que é uma Classe do LabVIEW?

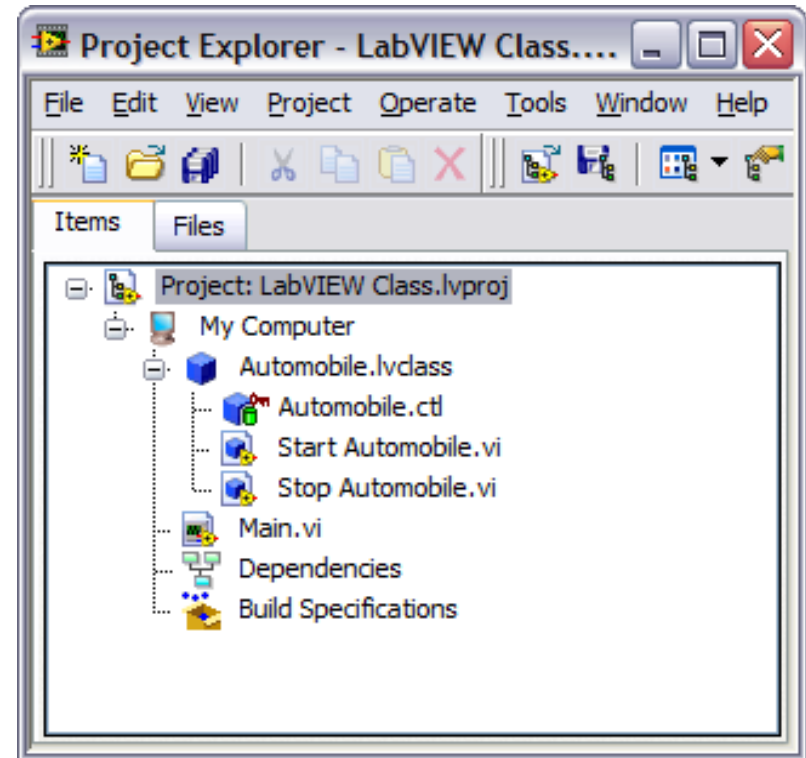
**Classe:** Um conjunto de dados e métodos que interagem com esses dados.

**Objeto:** Uma instância de uma classe.



Uma classe do LabVIEW é...

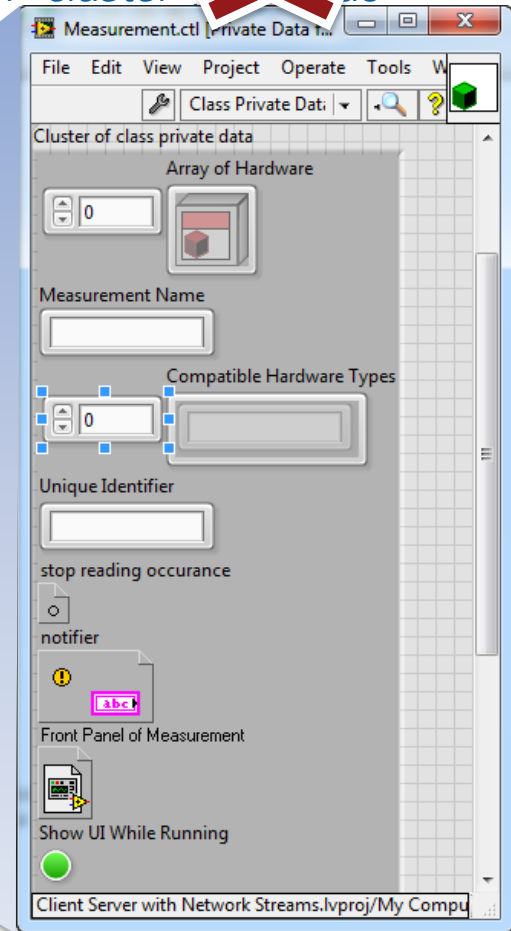
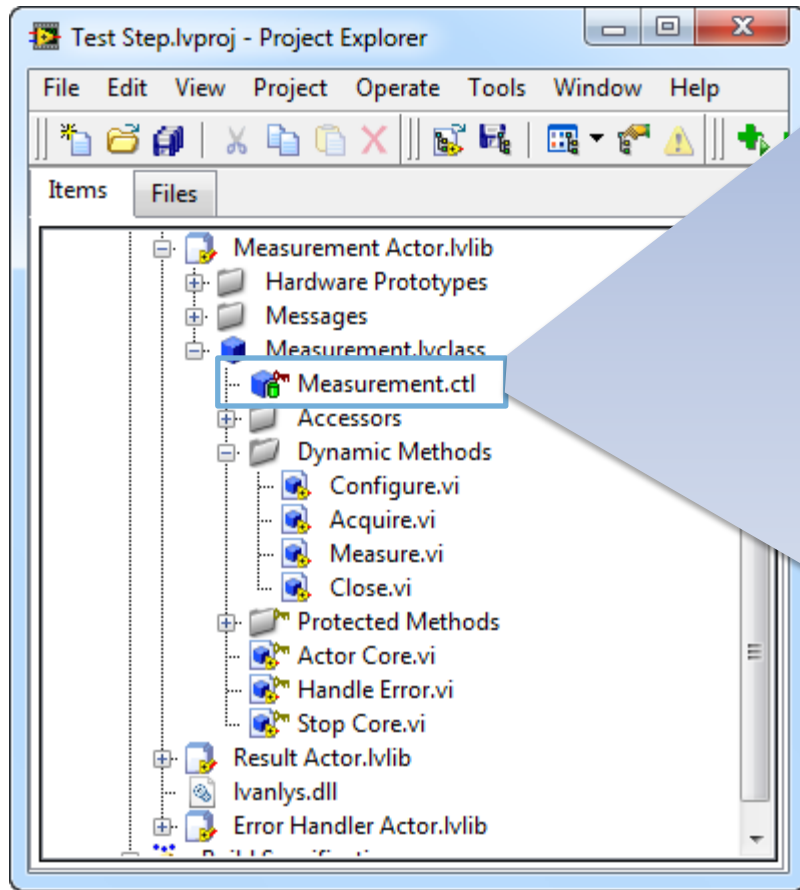
- **Cluster "Chuck Norris"**
- Um tipo de dado definido pelo usuário
- Um tipo de biblioteca de projeto



# Controle de dados privados: encapsulamento!

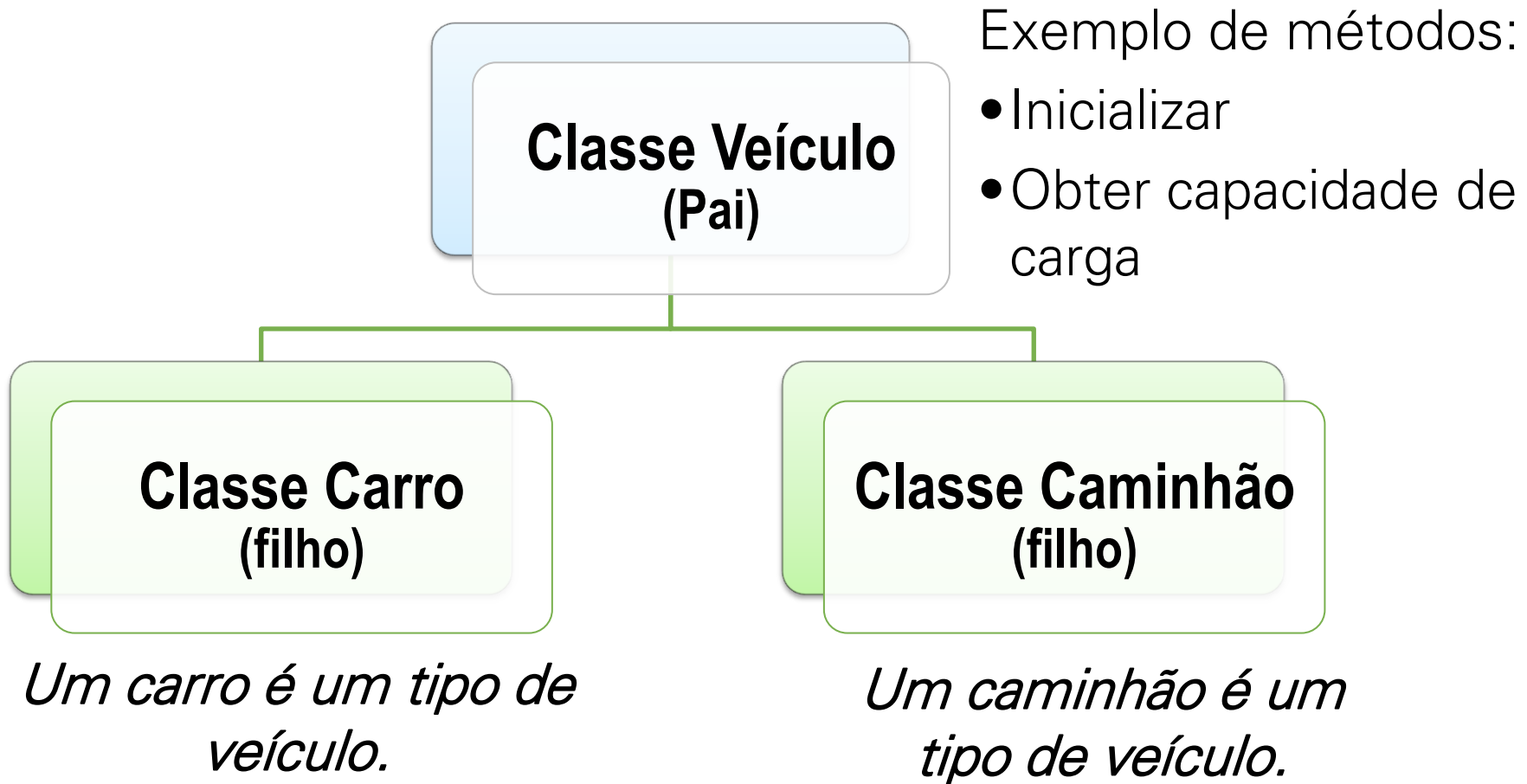
O motivo pelo qual algumas pessoas chamam classes de um "*cluster* ~~privado~~"

## Cluster "Chuck Norris"



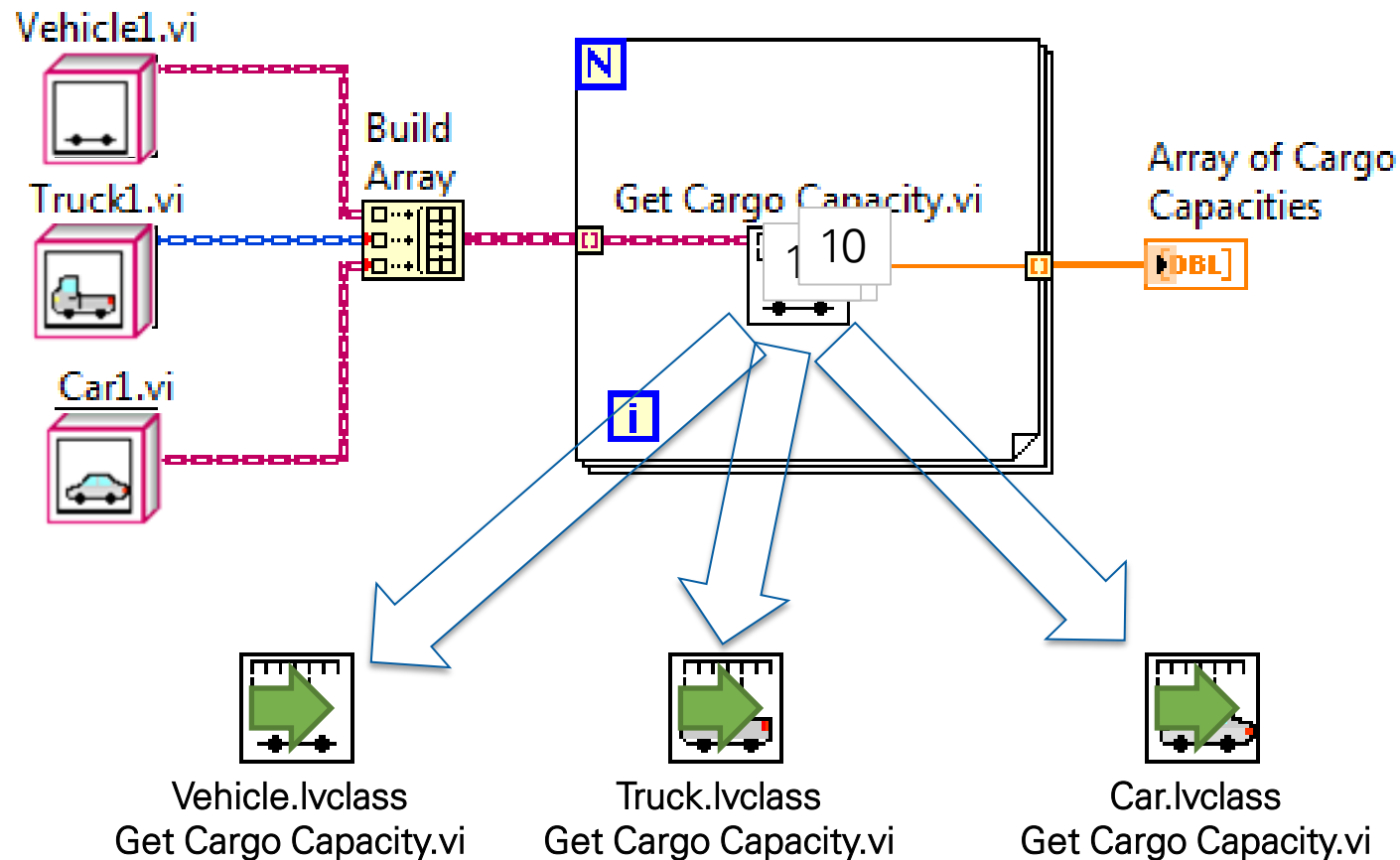
Somente VIs pertencentes à classe podem acessar ou modificar estes dados.

# O que é herança?



# Dynamic Dispatch (Polimorfismo)

A ligação dinâmica pode ser imaginada como um “polimorfismo no tempo de execução”. O objeto que você passa em um terminal de ligação dinâmica determina o método específico a ser invocado no tempo de execução.





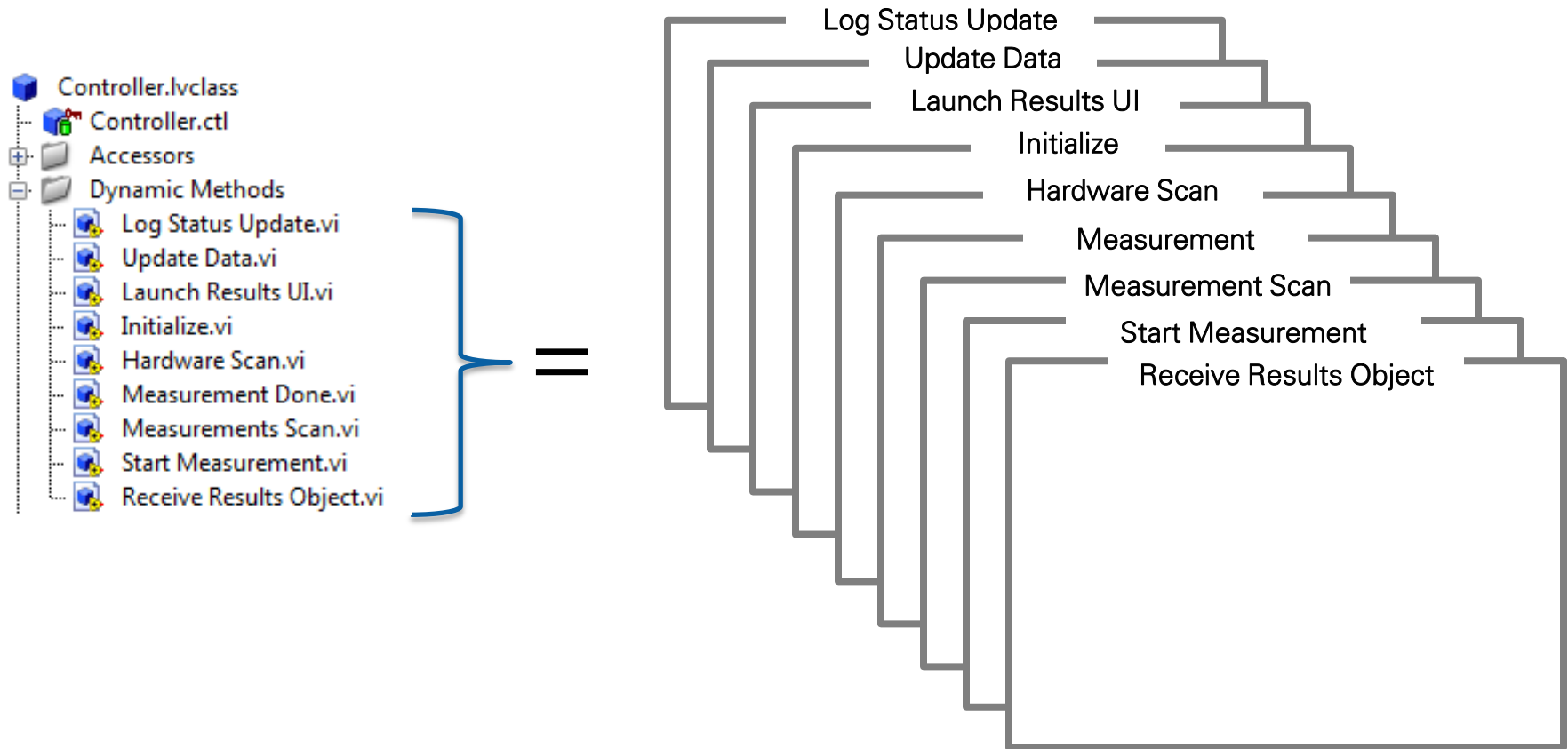
# Demonstração 1



# Para fazer o projeto orientado ao objeto

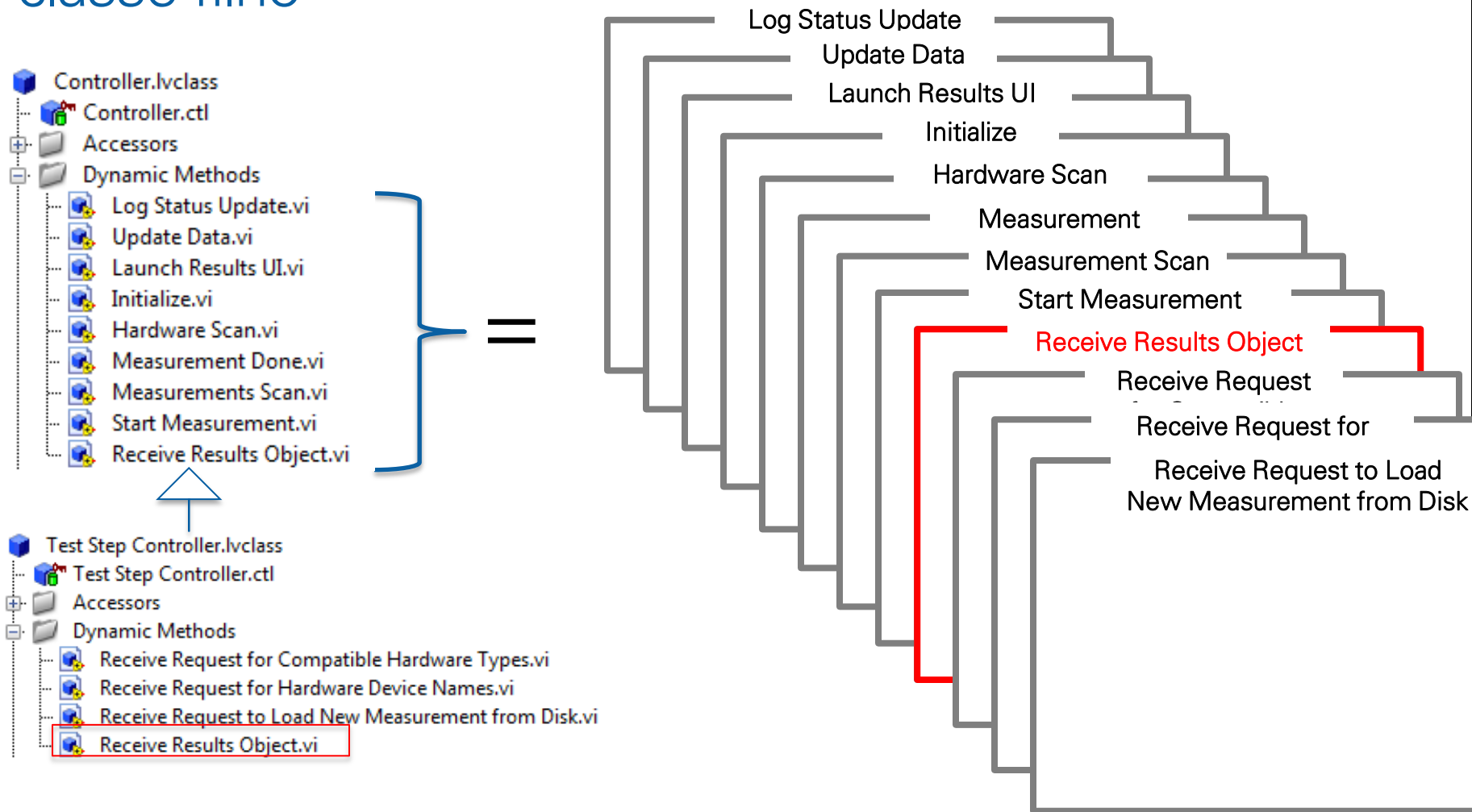
1. Descreva as classes (atores, substantivos)
2. Descreva os métodos (ações/solicitações, verbos)
3. Crie arquivos .lvclass para as classes
4. Crie VIs para os métodos (incluindo o connector pane)
5. Atribua o escopo para os VIs (público/protegido/privado)
6. Grave e TESTE o código para a classe mais simples primeiro, então, prossiga para as classes mais complexas

# Os métodos de um ator são executados quando uma mensagem é recebida



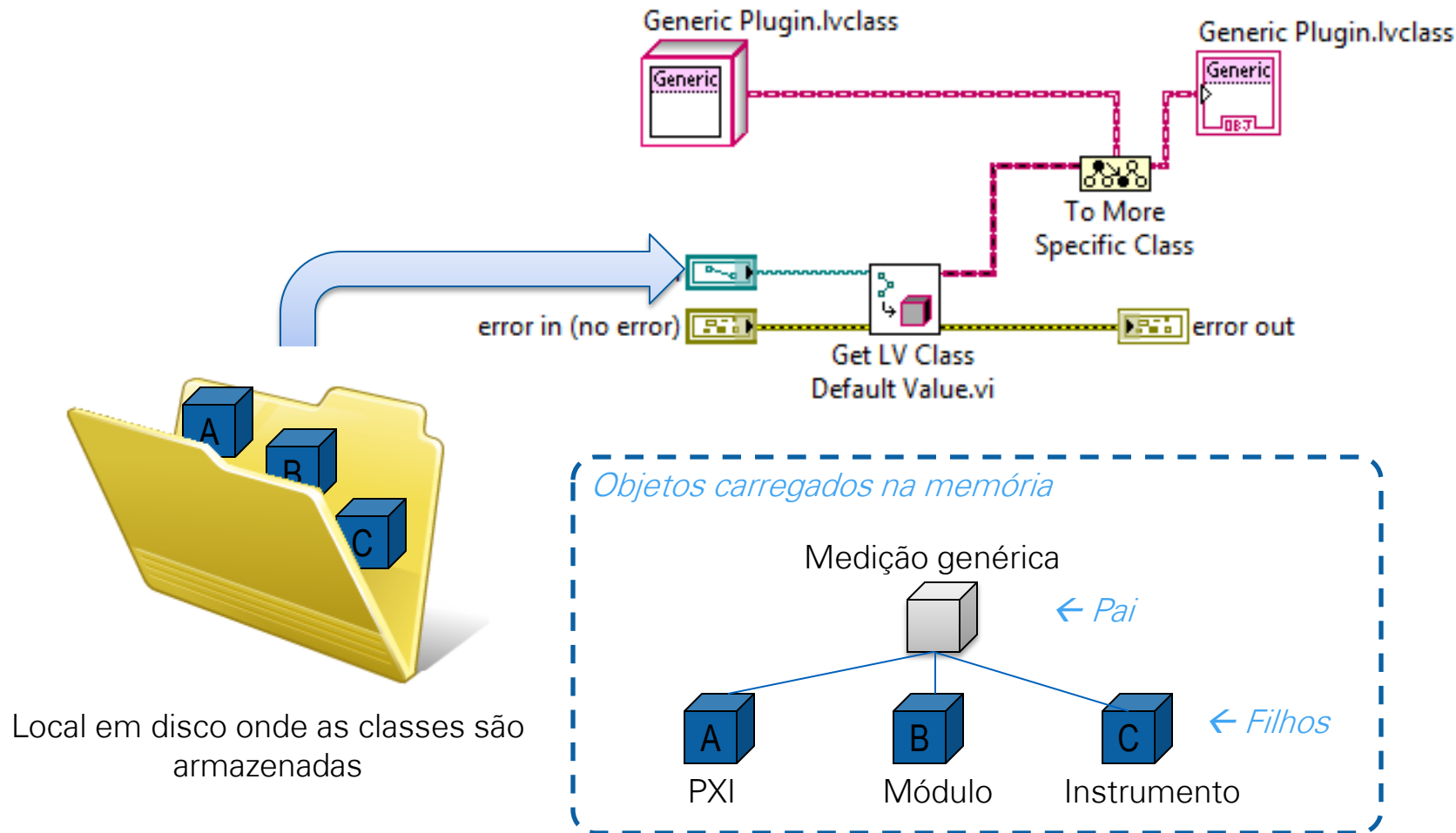
Métodos do Ator definem as mensagens que podem ser consumidas por ele mesmo

# Métodos podem ser alteradas ou adicionadas na classe filho





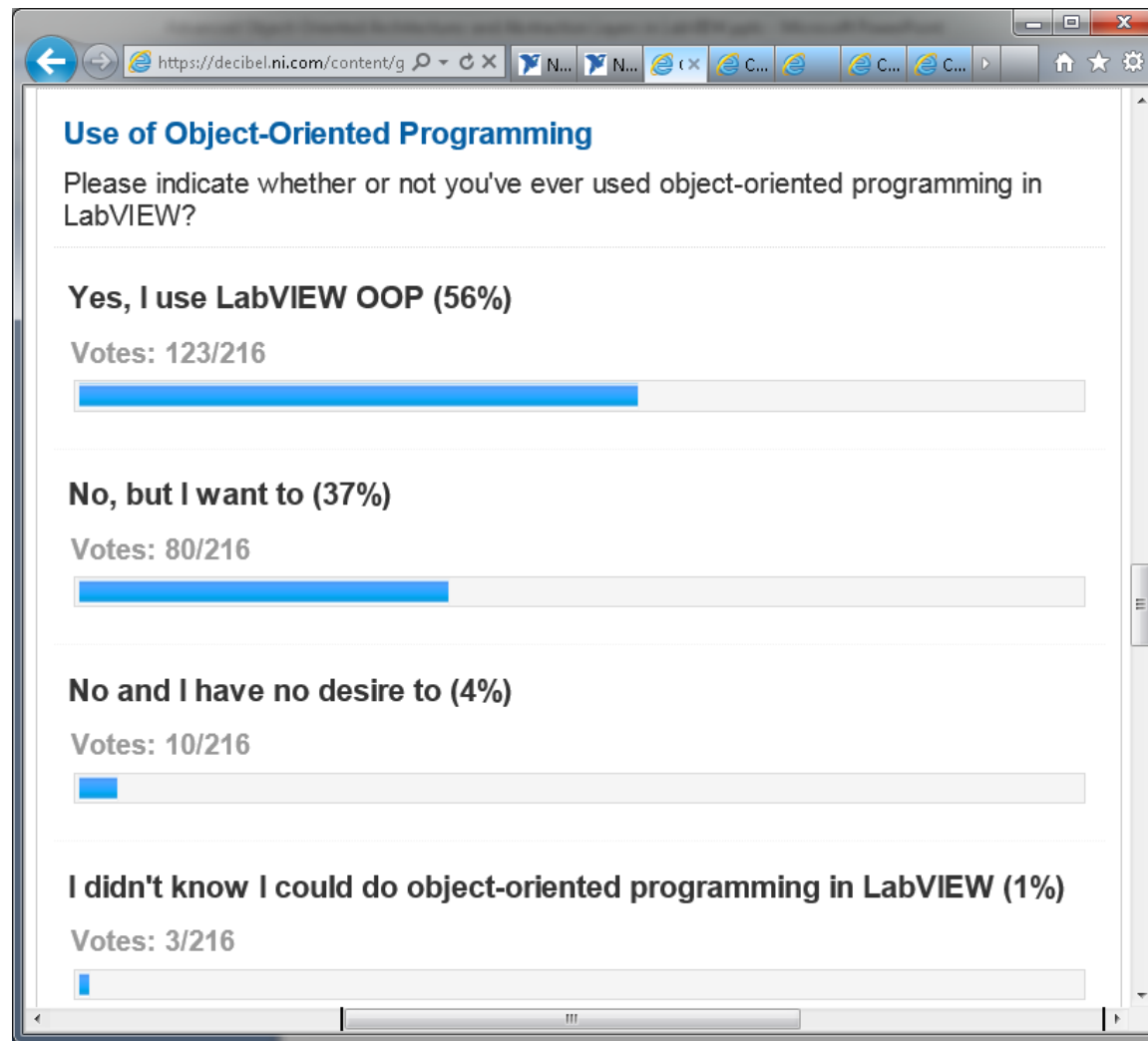
# Compartilhamento de classes e padrões de projeto



# Demonstração 2

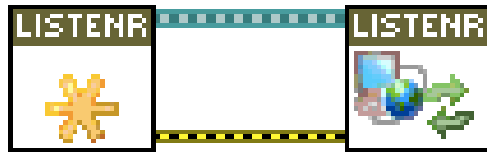


# Comunidade: LabVIEW Development Best Practices



<https://decibel.ni.com/content/groups/large-labview-application-development>

# Um último pensamento... Critique sua arquitetura



“Isso frequentemente me deixa com os pés no chão, pois eu percebo que resultado meu trabalho (o produto) é várias vezes imperfeito e pode ser melhorado”

*Veja as séries em Humildade e a Melhor Programação em [LabVIEWJournal.com](http://LabVIEWJournal.com)*



# Resumo

Orientação a objeto em LabVIEW lhe ajuda a desenvolver um código consistente, no qual:

- O escopo do projeto é conhecido com antecedência
- Os VIs tem funções bem definidas
- Cada classe pode ser testada e validada individualmente
- O código se torna modular e escalável
- Novas funcionalidades são adicionadas e depuradas sem provocar um grande impacto na aplicação final
- Reutilização e compartilhamento de código

Novo Usuário

Usuário Experiente

Usuário Avançado

LabVIEW Core 1

LabVIEW Core 2

LabVIEW Core 3

LabVIEW Connectivity

Object-Oriented Design  
and Programming in LabVIEW

LabVIEW Performance

Managing Software  
Engineering in LabVIEW

Advanced Architectures  
in LabVIEW

Certificações

Certified LV Associate  
Developer

Certified LabVIEW  
Developer

Certified LabVIEW  
Architect

Outros Treinamentos

LabVIEW Real-Time 1  
LabVIEW Real-Time 2

DAQ & Signal Conditioning  
LabVIEW FPGA

LabVIEW Machine Vision

# Baixe exemplos e apresentações

[ni.com/largeapps](http://ni.com/largeapps)



Ferramentas de engenharia de software



Práticas de desenvolvimento



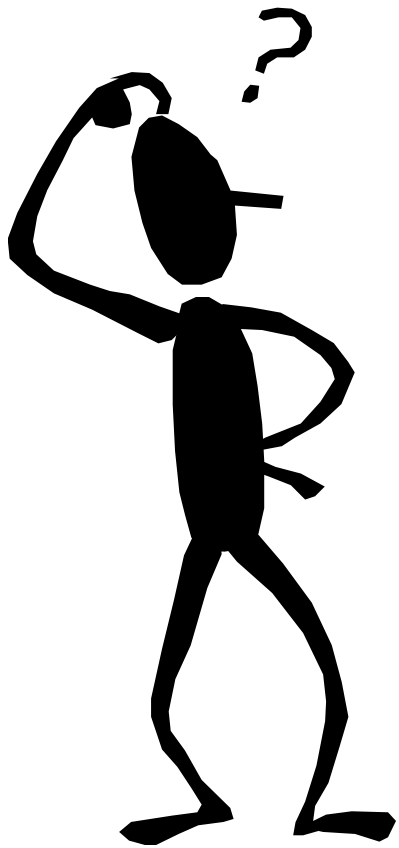
Comunidade para grandes aplicações



Comunidade para grandes aplicações



Práticas de desenvolvimento



Perguntas?



Obrigado!



# NIDays

## GRAPHICAL SYSTEM DESIGN CONFERENCE

Data: 27/03/2014

---

Local: Center Norte