

# Programação Modular

## Qualidade de software: uma introdução

Prof. Dr Johnatan Oliveira

PUC Minas

Bacharelado em Engenharia de Software

# Engenharia de software

2

- ▣ Objetivo da Engenharia de Software:
  - ▣ Produção sistemática de software de boa qualidade.



# Engenharia de software

3

## ▣ Produção sistemática:

### ▣ Processo de desenvolvimento de software:

- Especificação;
- Desenvolvimento;
- Teste/manutenção;
- Evolução.

# Engenharia de software

4

## ▣ Boa qualidade:

- ▣ Fatores externos: aqueles percebidos pelos usuários ou com os quais os usuários interagem.
- ▣ Fatores internos: aqueles relacionados com o projeto, com os quais os programadores interagem.

# Fatores externos

5

- Correção
- Robustez
- Extensibilidade
- Reusabilidade
- Compatibilidade
- Portabilidade
- Eficiência
- Facilidade de uso
- Integridade
- Verificabilidade

# Fatores internos

6

- Fatores externos são os percebidos pelos usuários.
  - *Muito importante!!*
- Fatores externos são atingidos por meio dos fatores internos:
  - Legibilidade ( + Inteligibilidade);
  - Manutenibilidade;
  - Modularidade.

# Fatores internos

7

## □ Legibilidade:

- Facilidade de identificar os elementos que compõem o código-fonte.
  - Nomeação de componentes, organização do código.

## □ Inteligibilidade:

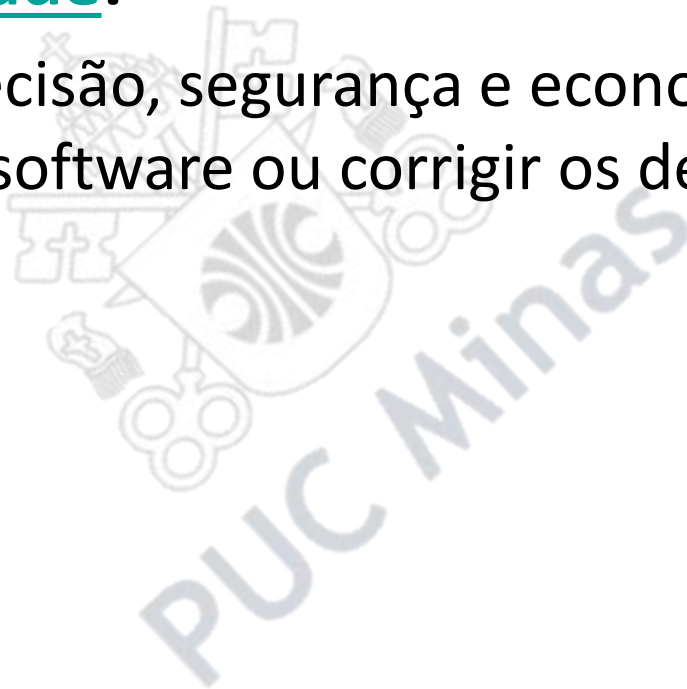
- Facilidade de compreender a coerência de um software em um nível superior do que a legibilidade proporciona.

# Fatores internos

8

## ▣ Manutenibilidade:

- ▣ Facilidade, precisão, segurança e economia para modificar um software ou corrigir os defeitos detectados.





# Fatores internos

9

## ▣ Modularidade:

- ▣ *“Mecanismo para aumentar a flexibilidade e compreensibilidade de um sistema, ao mesmo tempo em que permite a redução do seu tempo de desenvolvimento.”*

*(Parnas, David L. On the Criteria To Be Used in Decomposing Systems into Modules. Communications of the ACM, Vol. 15, No. 12, pp. 1053 – 1058, 1972. Tradução livre do autor.)*

# Fatores internos

10

## ▣ Modularidade:

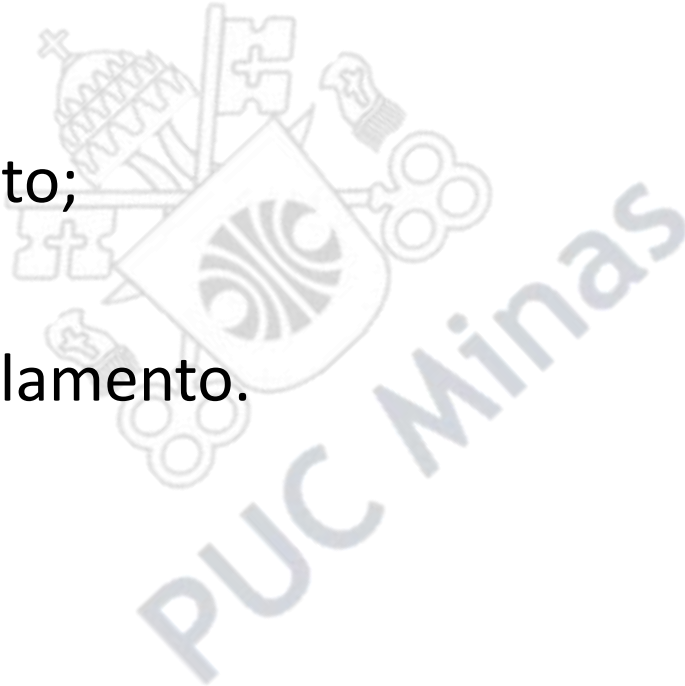
- ▣ Divisão do sistema em partes independentes: módulos.
- ▣ Ideia fundamental: quebrando o problema em partes menores, facilita-se sua solução e a reutilização das partes.

# Fatores internos

11

## ▣ Modularidade:

- ▣ Encapsulamento;
- ▣ Abstração;
- ▣ Coesão e acoplamento.



# Fatores internos

12

## ▣ Modularidade:

- ▣ Isolamento para testes e manutenção;
- ▣ Interface pública uniforme;
- ▣ Criação de bibliotecas reutilizáveis;
- ▣ Economia de memória.

# Qualidade de software

13

## ❑ Programando descuidadamente:

- ❑ Ignorar as boas práticas;
- ❑ Codificar antes de pensar;
- ❑ Nomenclatura mal feita;
- ❑ Código duplicado;
- ❑ Programar sem testar;
- ❑ Reaproveitar sem entender;

# Qualidade de software

14

- Programando descuidadamente:
  - Não usar ou usar incorretamente o controle de versões;
  - Ignorar avisos e erros;
  - Não tratar entrada de dados;
  - Pouca documentação;
  - (Muita) documentação substituindo refatoração;
  - Alto acoplamento/dependência (projeto espaguete).

# Começando: e o seu código?

15

- Observe e comente no exercício de revisão:

Correção

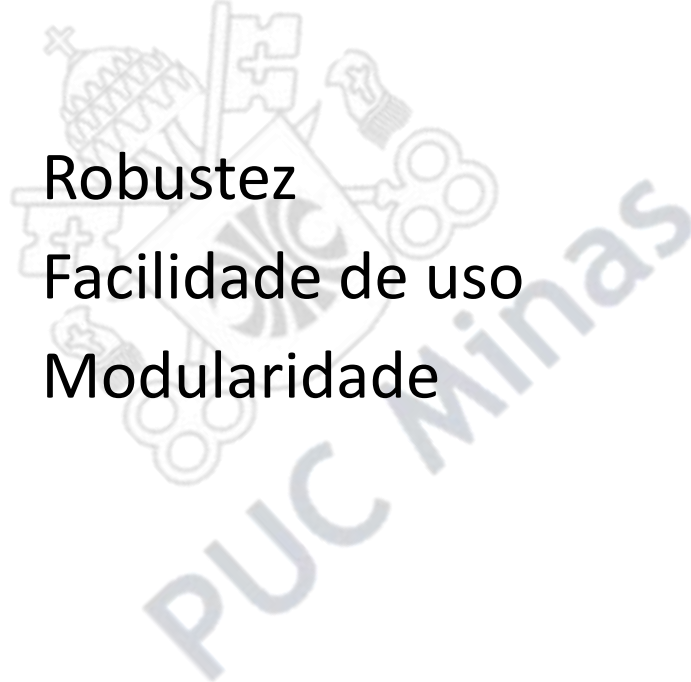
Eficiência

Legibilidade

Robustez

Facilidade de uso

Modularidade



# Um recado do Fowler

16

*“Any fool can write code that a computer can understand... But only good programmers write code that humans can understand.”*

*(Martin Fowler)*

