

# Sistemas para Internet: Fundamentos de Engenharia de Software

## Teste de Software

Professor: Rafael Martins Alves

Cuiabá-MT/2021-2

# Agenda

- Questões da prova
- Seminário
- Projeto
- Teste de Software

## Prova

- Ver avaliação;
  - Média acerto: 94,8%
  - Maior acerto: 100,00%
  - Menor acerto: 50,00%

## Dias de Aula

Número de aulas

**29/10** - Projeto & Conteúdos

**05/11** - Projeto & Conteúdos

**12/11** - Projeto & Conteúdos

**19/11** - Projeto & Conteúdos

**26/11** - Seminário

**03/12** - Seminário

**10/12** - Pendências

# Seminário

- Em dupla;
- Mínimo 10 minutos e máximo 15 minutos;
- Pode ser gravado vídeo e disponibilizado no YouTube;
  - Interessante ser em inglês
- Sugestões de conteúdos:
  - Visão geral medição de software: definição, conceitos e técnicas
  - Princípios e conceitos de análise de software orientada a objetivos: atividades e tarefas: **Gabriel Honda e Carlos Henrique**
  - Gestão de configuração e mudança: objetivo, conceitos atividades e tarefas
  - Segurança de inteligência artificial
- Blockchain prático: **Hellen e Francilene**
- Coisas autônomas
- Nuvens distribuídas: **Thiago e Nayla**
- Edge Computing (Computação de Borda)
- Transparência e rastreabilidade
- Aprimoramento humano
- Democratização da expertise
- Multiexperiência
- Hiperautomação: **Gabriel e Guilherme**
- Chatbots inteligentes: **Cezarino e Jocínia**
- Inteligência artificial em todos os lugares
- Privacidade e segurança
- Big Data: **Jackeline e Elias**
- Sistemas Conversacionais
- IoT: Internet of Things: **Tales**
- Cloud Computer
- Ciência de dados
- Realidade aumentada: **Monyke e Michele**

## Projeto

- Projeto em grupo de 3 a 4 pessoas
- Assuntos:
  - Projeto/programa **da disciplina de Fundamentos de Programação e Programação Web Front-End;**
  - Projetos pessoais
- Apresentar a proposta de projeto no documento compartilhado
- Pasta compartilhada com os relatórios

## Projeto

- Projeto em grupo de 3 a 4 pessoas
- Assuntos:
  - Projeto/programa **da disciplina de Fundamentos de Programação e Programação Web Front-End;**
  - Projetos pessoais
- Meta da semana:
  - **Escolher o grupo**
  - **Escolher o tema**
  - **Fazer o objetivo geral**
- Apresentar a [proposta de projeto](#) no documento compartilhado
- Data de entrega do relatório: 03/12

## Projeto

- Carlos Henrique, Gabriel Honda e Tales Iago: **Chatbot para gestão de grupos de disciplina;**
- Nayla, Thiago e Gabriel: **Sistema para Academia;**
- Cezarino, Elizeu e Jocinia: **Sistema de PetShop (App e Web);**
- Monyk, Francilene e Hellen: XXX;
- Jackeline, Elias e Alcides: **App para controle de medicamento contínuo;**



## Motivação

### Foguete Ariane 5

- O foguete **perdeu o controle** de altitude e ativou a autodestruição
- Motivo: um programa que **converteria** um número real para um número inteiro de 16 bits recebeu como entrada um valor fora da faixa permitida;
- Ocorreu um **erro out of range** que desnorteara os sistemas.





# Motivação

## Falha nas linhas telefônicas da AT&T (1990)

Custo: 75M de chamadas e 200K reservas de voos perdidas.

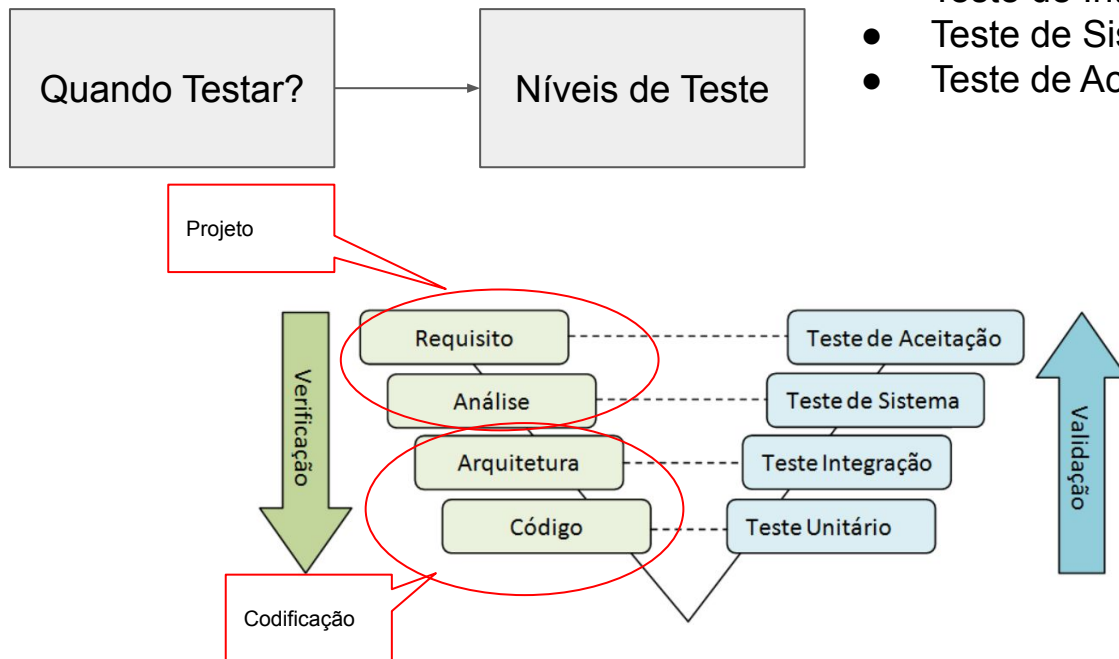
Causa: Uma **única** linha de código com bug, após uma atualização de software de um switch

```
1 while (ring receive buffer not empty
    and side buffer not empty) DO
2   Initialize pointer to first message in side buffer
    or ring receive buffer
3   get copy of buffer
4   switch (message)
5     case (incoming_message):
6       if (sending switch is out of service) DO
7         if (ring write buffer is empty) DO
8           send "in service" to status map
9         else
10          break
        END IF
11     process incoming message, set up pointers to
        optional parameters
12     break
    END SWITCH
13 do optional parameter work
```

→ Custou \$60 milhões!

# Quanto testar?

- Teste de Unidade
- Teste de Integração
- Teste de Sistema
- Teste de Aceitação



# Teste Unitário

- Implementação das classes
  - Atributos
  - Métodos (funções)

Os teste unitário, deve testar unidades de trabalhos **isolados**, a fim de mostrar que funcionam individualmente.

Existem ferramentas “*framework*” que auxilia o teste. No java temos o Junit. Mostrar funcionando.

# Teste Unitário

- Exemplo, calcular média notas:

## FrontEnd



The FrontEnd interface is a web form titled "Nota Final 2ºEM ou de acordo com Manual". It contains two input fields for "Português\*" and "Matemática\*", both ranging from 0.00 to 10.00. Below these is a note: "\*Informe a Nota Final destas matérias obtidas no 2ºEM. O conceito utilizado para classificação do candidato será o numérico, com dois dígitos após o ponto (0.00 a 10.00). USE PONTO(.) AO INVÉS DE VÍRGULA(,)." There is a section "Sistema de Pontuação Acrescida" with checkboxes for "Afrodescendência" and "Escolaridade Pública". A red "Calcular" button is present. At the bottom, there is a green box for "Estimativa da Nota" showing "0" and an orange button for "Escolher outra instituição".

## BackEnd

Calcula Nota

Calculadora  
-Soma  
-Divisão

Teste  
Unitário:  
Soma e  
Divisão

## Vantagens? Desvantagens?

# Teste Integração

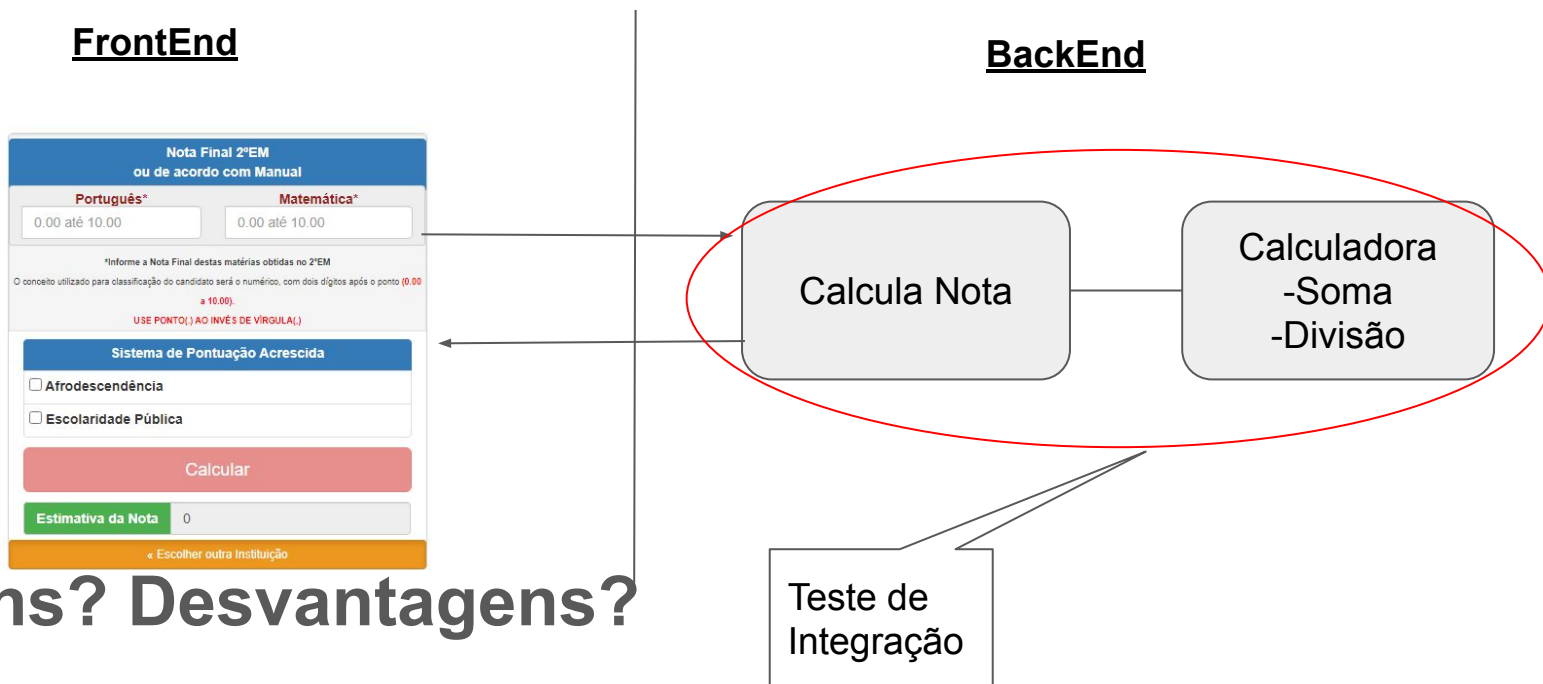
Validar a “ligação” entre os componentes de um sistema;

Testes de Integração  
Comece a pensar no todo!



# Teste Integração

- Exemplo, calcular média notas:



Vantagens? Desvantagens?

# Teste Sistema e Aceitação

Sistema do ponto de vista do **usuário final**

- **Teste de sistema:** planejado pela equipe de teste (sem os desenvolvedores)
- **Teste de aceitação:** executado pelo usuário após o teste de sistema





# Teste Sistema e Aceitação

- Exemplo, calcular média notas:

Teste de  
Sistema e  
Aceitação

FrontEnd



Nota Final 2ªEM  
ou de acordo com Manual

Português\* Matemática\*

0.00 até 10.00 0.00 até 10.00

\*Informe a Nota Final destas matérias obtidas no 2ªEM  
O conceito utilizado para classificação do candidato será o numérico, com dois dígitos após o ponto (0.00 a 10.00).  
USE PONTO(.) AO INVÉS DE VÍRGULA(,)

Sistema de Pontuação Acrescida

☐ Afrodescendência  
☐ Escolaridade Pública

Calcular

Estimativa da Nota 0

« Escolher outra instituição

BackEnd

Calcula Nota

Calculadora  
-Soma  
-Divisão

Vantagens? Desvantagens?

## Atividades

1) Com relação ao teste de aceitação, assinale a alternativa correta.

Escolha uma opção:

- a. O teste de aceitação verifica as permissões de uso do sistema e identifica se o sistema aceita usuários não autenticados.
- b. O teste de aceitação é utilizado para definir quando o sistema está pronto para ser lançado em produção.
- c. O teste de aceitação verifica os conjuntos de dado de entrada que o sistema aceita.
- d. O teste de aceitação avalia a usabilidade do sistema e a aceitação dos usuários.
- e. O teste de aceitação deve ser executado para verificar se os módulos do sistema são aceitos para serem integrados.

## Atividades

2) Em relação ao teste de software, assinale a alternativa correta sobre o teste de integração.

Escolha uma opção:

- a. Visa testar as falhas decorrentes da integração dos módulos do sistema.
- b. Tem a mesma aplicação do teste de aceitação.
- c. Teste realizado pelos usuários finais do software.
- d. Via descobrir falhas por meio de utilização do mesmo.
- e. Tem como objetivo explorar a menor unidade de um projeto.

## Atividades

3) O teste de aceitação normalmente é realizado utilizando-se a interface final do sistema. Sobre esse tipo de teste pode-se afirmar que

Escolha uma opção:

- a. assim como o teste de sistema, faz a verificação de defeitos do sistema.
- b. tem como objetivo principal a validação do software quanto aos requisitos.
- c. os sistemas feitos sob medida (tailored) sempre devem ser testados por testes de aceitação alfa e beta.
- d. é realizado pelo programador para testar componentes individuais do sistema.
- e. é realizado pela equipe de desenvolvimento.

# Atividades

4) Numere a segunda coluna de acordo com a primeira, associando os Níveis de Teste de Software às suas respectivas características.

- (1) Teste de Unidade
- (2) Teste de Integração
- (3) Teste de Sistema
- (4) Teste de Aceitação

( ) Avalia o software com respeito ao projeto de seus subsistemas e detecta suposições errôneas sobre pré e pós-condições para execução de um componente, falhas nas interfaces de comunicação dos componentes do software.

( ) Avalia o software com respeito aos seus requisitos e detecta falhas nos requisitos e na interface com o usuário.

( ) Avalia o software com respeito a sua implementação detalhada e detecta falhas de codificação, algoritmos ou estruturas de dados incorretos ou mal implementados.

( ) Avalia o software com respeito ao seu projeto arquitetural e detecta falhas de especificação, desempenho, robustez e segurança.

A sequência correta de preenchimento dos parênteses, de cima para baixo, é

Escolha uma opção:

- a. 1 – 4 – 2 – 3.
- b. 1 – 3 – 4 – 2
- c. 4 – 2 – 3 – 1.
- d. 2 – 4 – 1 – 3.
- e. 2 – 1 – 3 – 4.

## Atividades

5) \_\_\_\_\_ é o teste que tem como foco as menores unidades de um programa, que podem ser funções, procedimentos, métodos ou classes. Neste contexto, espera-se que sejam identificados erros relacionados a algoritmos incorretos ou mal implementados, estruturas de dados incorretas ou simples erros de programação. Como cada unidade é testada separadamente, este teste pode ser aplicado à medida que ocorre a implementação e pelo próprio desenvolvedor, sem a necessidade de dispor-se do sistema totalmente finalizado.

Assinale a alternativa que preenche corretamente a lacuna do texto acima.

Escolha uma opção:

- a. Teste de regressão
- b. Teste de sistema
- c. Teste de integração
- d. Teste de aceitação
- e. Teste de unidade

# Atividades

6) \_\_\_\_\_ é uma verificação de consistência entre o sistema de software e sua especificação e, portanto, é uma atividade de verificação feita depois que se tem o sistema completo, com todas suas partes integradas para verificar se as funcionalidades especificadas nos documentos de requisitos estão todas corretamente implementadas. Este tipo de teste é focado principalmente na descoberta de falhas e executado pelo grupo de desenvolvimento de testes, tendo também um papel importante para avaliar se o produto pode ser liberado para os consumidores, o que é diferente do seu papel de expor falhas que são removidas para melhorar o produto.

Assinale a alternativa que preenche corretamente a lacuna do texto acima.

Escolha uma opção:

- a. Teste de regressão
- b. Teste de unidade
- c. Teste de sistema
- d. Teste de integração
- e. Inspeção

## Referência

QUEIROZ PINTO, Bruno. **Engenharia de Software I**. [S. l.], 25 abr. 2018. Disponível em:

<https://sites.google.com/a/iftm.edu.br/profbruno/semestres-anteriores/2017-01/engenharia-e-software-i>. Acesso em: 8 out. 2021.

SOMMERVILLE, Lan. **Engenharia de software**. 9ª ed. São Paulo: Pearson Addison Wesley, 2011.

ENGENHARIA de Software I. [S. l.], 18 ago. 2021. Disponível em:

<https://sites.google.com/a/iftm.edu.br/profbruno/semestres-anteriores/2017-01/engenharia-e-software-i>. Acesso em: 18 ago. 2021.

ENGENHARIA de Software Moderna. [S. l.], 2020. Disponível em:

<https://engsoftmoderna.info/slides.html>. Acesso em: 18 ago. 2021.