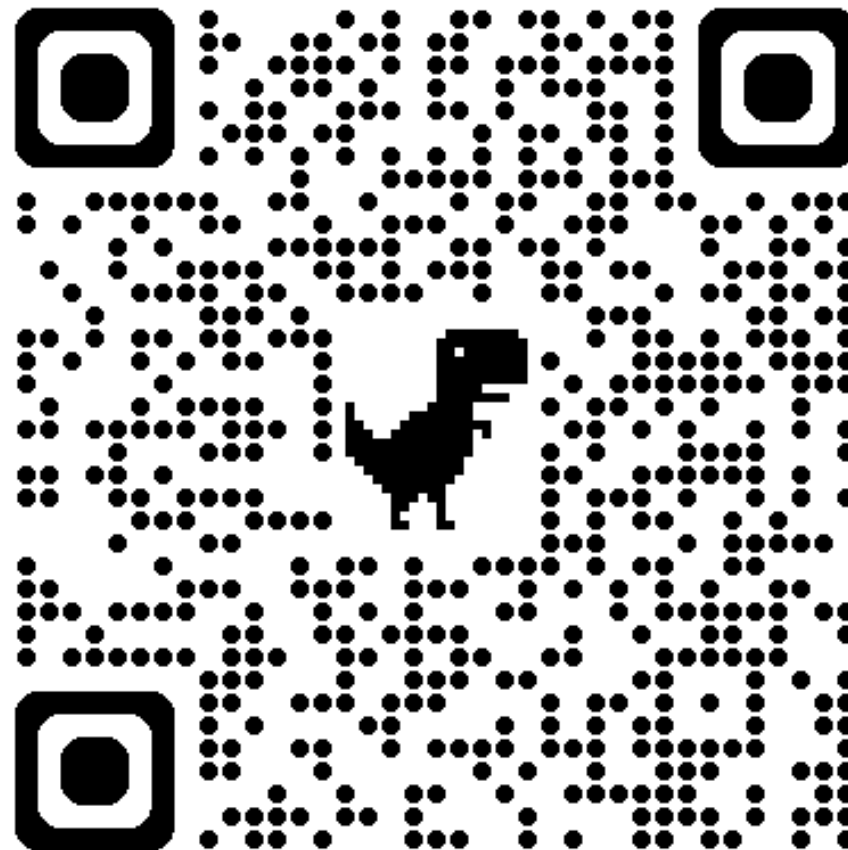


PROGRAMAÇÃO ORIENTADA A OBJETOS

João Choma Neto

joao.choma@unicesumar.edu.br

Unicesumar – Maringá



O QUE É TESTE DE SOFTWARE?

TESTE

Tudo o que consumimos no dia a dia é testado antes de chegar às prateleiras

Seja para garantir nossa segurança, como no caso de produtos físicos, seja para garantir nossa saúde, como no caso de alimentos

Programas de computador também são testados

TESTE DE SOFTWARE

Definição técnica:

“Teste de software é todo e qualquer **procedimento** que ajuda a determinar se o programa **atinge** as **expectativas** para as quais **foi criado** (BRAGA, 2016 apud NETO, 2010).”

Exemplo: Editor de texto

Expectativa: criar ou editar textos.

TESTE DE SOFTWARE

Testar um software consiste em:

- Verificar se ele atende às expectativas
- Se seu funcionamento é limpo, amigável e correto
- Se ele se enquadra no ambiente para o qual foi projetado

TESTE DE SOFTWARE

O teste tenta garantir que o programa funcione conforme o esperado, seja seguro, confiável e atenda às necessidades dos usuários

DEFEITO

Defeito (bug ou fault): É um problema ou imperfeição no código ou na lógica do software que causa um comportamento indesejado

DEFEITO

Resultado de um código mal escrito

Implementação equivocada

Causa anomalia no funcionamento do sistema

O usuário final normalmente não vê o defeito propriamente dito

Os defeitos podem permanecer no software até que sejam descobertos e corrigidos por meio de atividades de teste

ERRO

Falha humana e produz resultado incorreto

Exemplo: a falha na escrita de um código específico

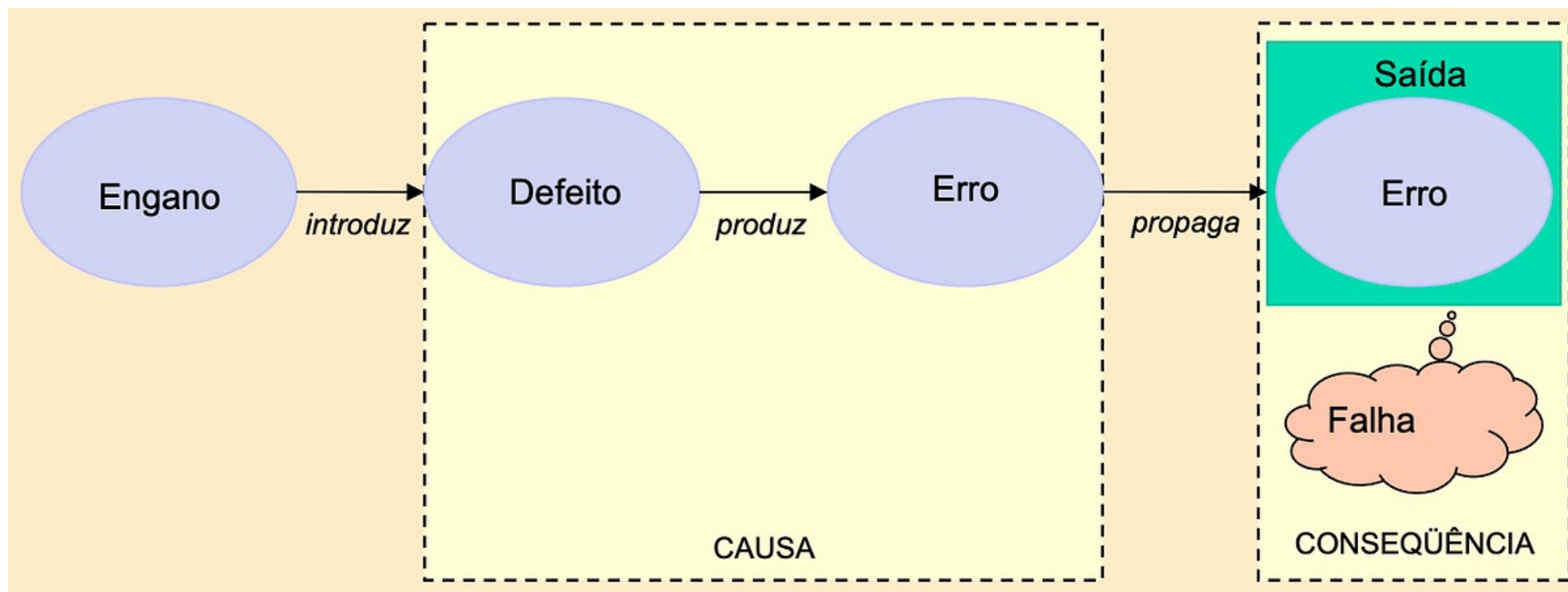
É o ato humano de cometer um engano ou incorreção durante o desenvolvimento do software

FALHA

É a ocorrência indesejada de um defeito durante a execução do software em um ambiente de produção ou em condições reais de uso

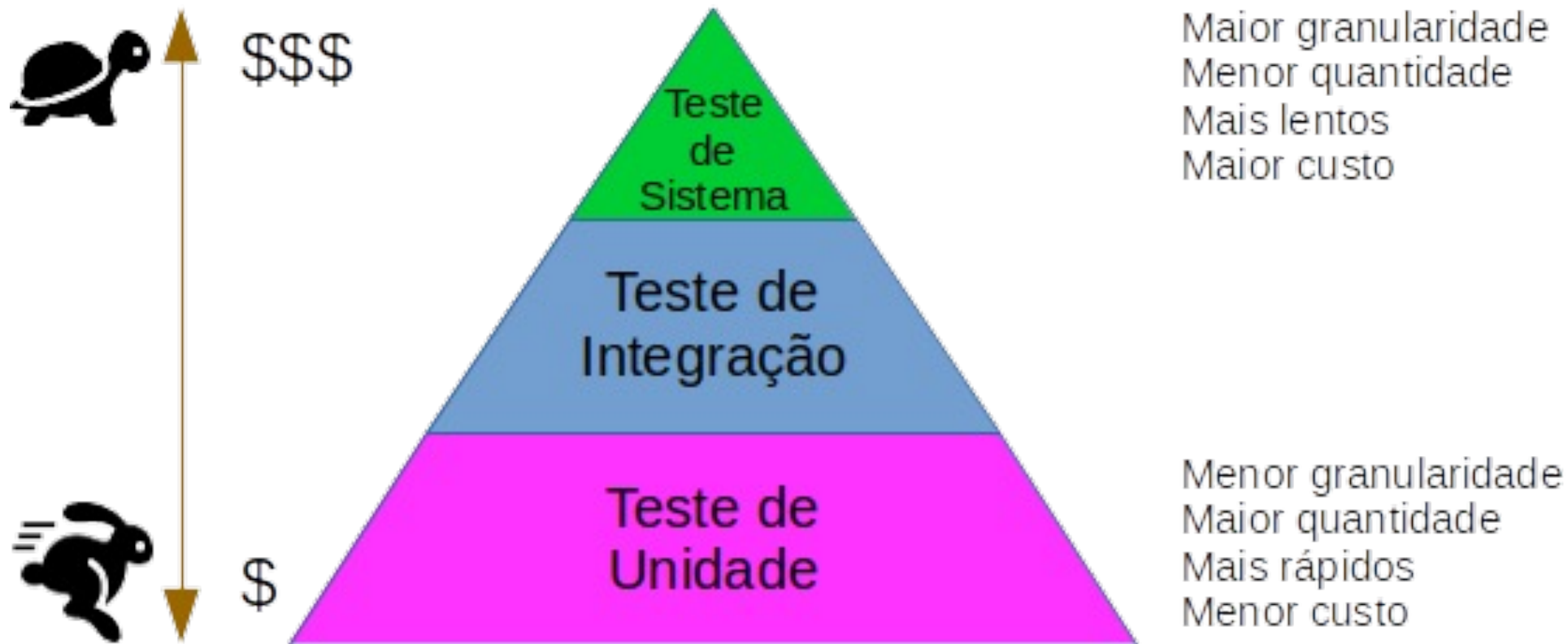
FALHA

Quando um defeito é ativado em tempo de execução, ele pode levar a falhas no software, resultando em comportamentos inesperados, travamentos, resultados incorretos ou outros problemas que afetam a funcionalidade do programa



TESTE DE UNIDADE





TESTE UNITÁRIO

- Teste unitário é uma prática de teste de software na qual cada componente individual ou "unidade"

TESTE UNITÁRIO

- Uma unidade pode ser a menor parte testável de um aplicativo
- Função
- Método
- Classe
- Módulo

TESTE UNITÁRIO

- O objetivo principal dos testes unitários é identificar defeitos nas unidades de código antes que elas sejam integradas ao restante do sistema

TESTE UNITÁRIO

- Isolar as unidades de código e testá-las independentemente
- Os desenvolvedores podem verificar se cada unidade executa as tarefas designadas de acordo com suas especificações

TESTE UNITÁRIO

- Identificar problemas cedo no processo de desenvolvimento
- Tornar mais fácil e mais econômico corrigir erros

TESTE UNITÁRIO

- Cada teste unitário deve ser automatizado para que possa ser executado repetidamente e integrado a fluxos de trabalho de desenvolvimento contínuo

TESTE UNITÁRIO

- Processo estruturado para garantir que cada componente individual do código seja testado de maneira isolada

UNIDADE

- **Identifique as Unidades de Código**
- Determine as unidades de código que você deseja testar

FRAMEWORK

- **Escolha um Framework de Testes**
- Utilize um framework de testes apropriado para a linguagem de programação que você está usando

FRAMEWORK

- JUnit (Java)
- NUnit (.NET)
- pytest (Python)
- Jasmine (JavaScript)

CASOS DE TESTE

- **Crie Casos de Teste**
- Para cada unidade de código, escreva casos de teste que cubram diferentes cenários
- Incluir casos em que a unidade deve produzir resultados corretos
- Incluir casos em que deve lidar com entradas inválidas ou inesperadas

TESTES

- **Escreva os Testes**
- Escreva o código dos testes usando o framework escolhido

TESTES

- **Dados de Teste**

- Inserir dados de teste relevantes para os casos de teste
- Criação de objetos fictícios
- Valores de entrada simulados
- Cenários de teste específicos

EXECUTAR

- **Execute os Testes**
- **Verifique os Resultados**
- O framework de testes geralmente fornecerá relatórios indicando quais testes passaram e quais falharam

REPETIR

- Repetir o processo de teste para todas as unidades de código
- Cada nova modificação recursos ao código repetir os testes
- Cada adição novos recursos, criar novos testes de unidade e repetir os testes

PROGRAMAÇÃO ORIENTADA A OBJETOS

João Choma Neto

joao.choma@unicesumar.edu.br

Unicesumar – Maringá

