



Aula – 3

Testes de Unidade

Disciplina: COM221 – Computação Orientada a Objetos II

Prof: Phyllipe Lima
phyllipe@unifei.edu.br

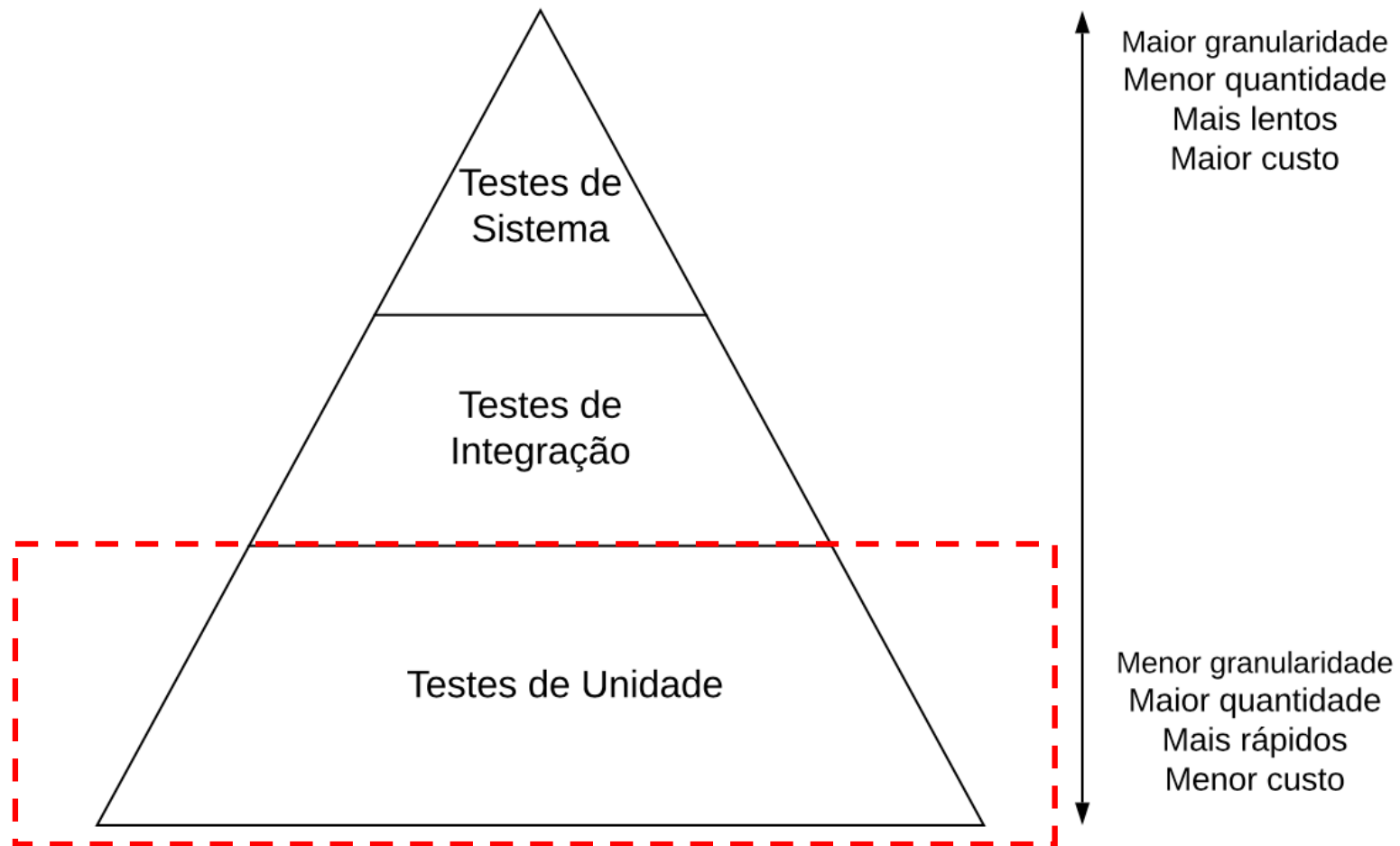
Universidade Federal de Itajubá – UNIFEI
IMC – Instituto de Matemática e Computação

Agenda

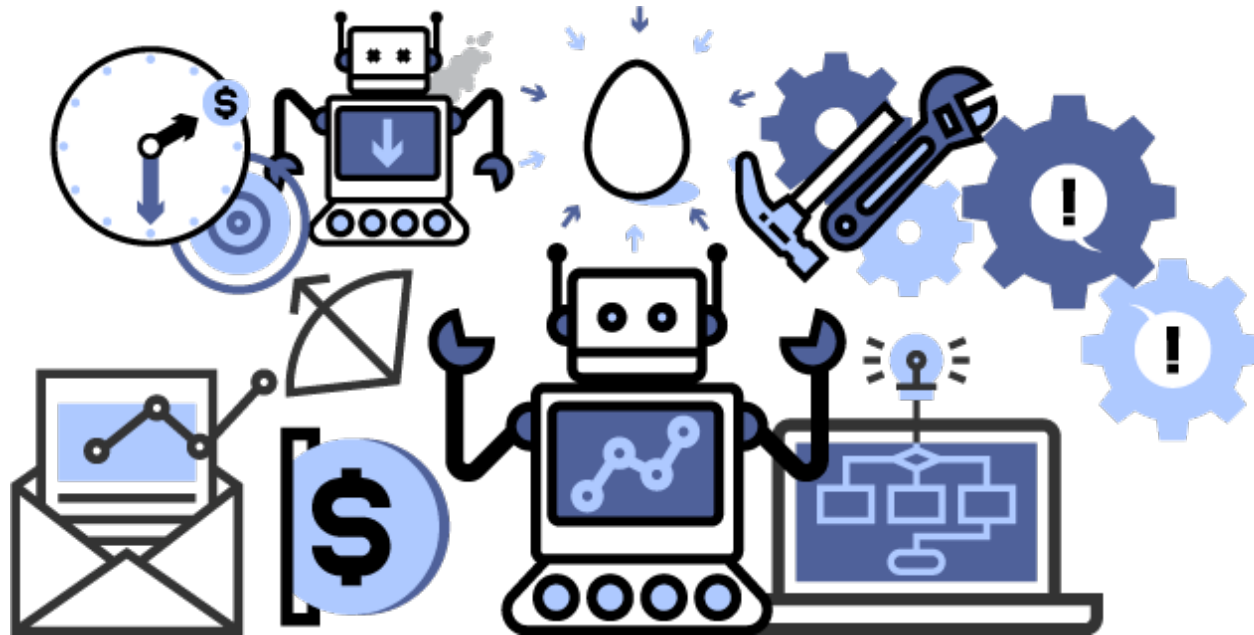


- ☐ Conhecer testes de unidade
- ☐ Junit
 - ☐ Assertivas
- ☐ Usar testes de unidade para guiar o desenvolvimento e modelar o sistema

❑ Base da pirâmide



- ❑ São testes ***automatizados*** de pequenas ***unidades*** de código, que são testadas de forma isolada.



❑ É um programa!

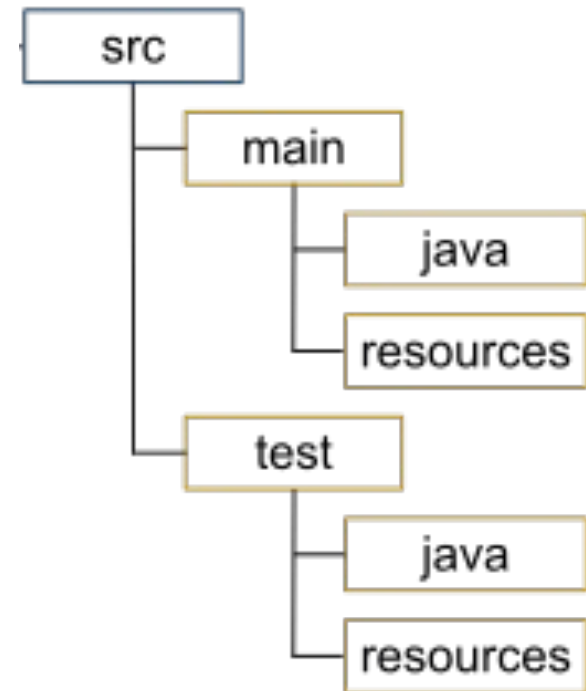
❑ Teste de unidade são escritos por programadores!



❑ Normalmente as classes ficam divididas em dois pacotes:

❑ Funcionalidade (main)

❑ Teste (test)



- ❑ São implementados por *frameworks* de teste

JUnit



pytest

JUnit.net

Por que testar?



- ❑ Por que pode ser interessante guiarmos o desenvolvimento do sistema através de testes?

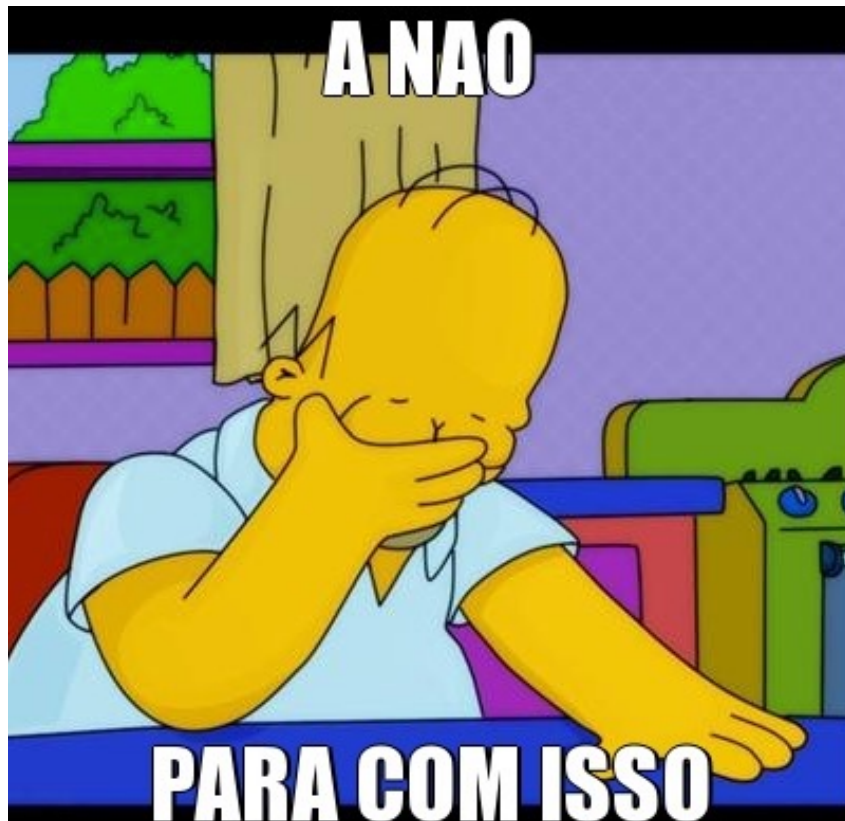




Thousands of
useful debugging
tools and options



`printf("it works");`



- ☐ Ao invés de imprimir mensagens na tela, escreva testes de unidade.
- ☐ Mensagens na tela precisam ser validadas manualmente e depois serão apagadas.
- ☐ Os testes permanecerão na suíte

❑ Encontrar bugs no início de desenvolvimento.....



❑antes de entrar em produção

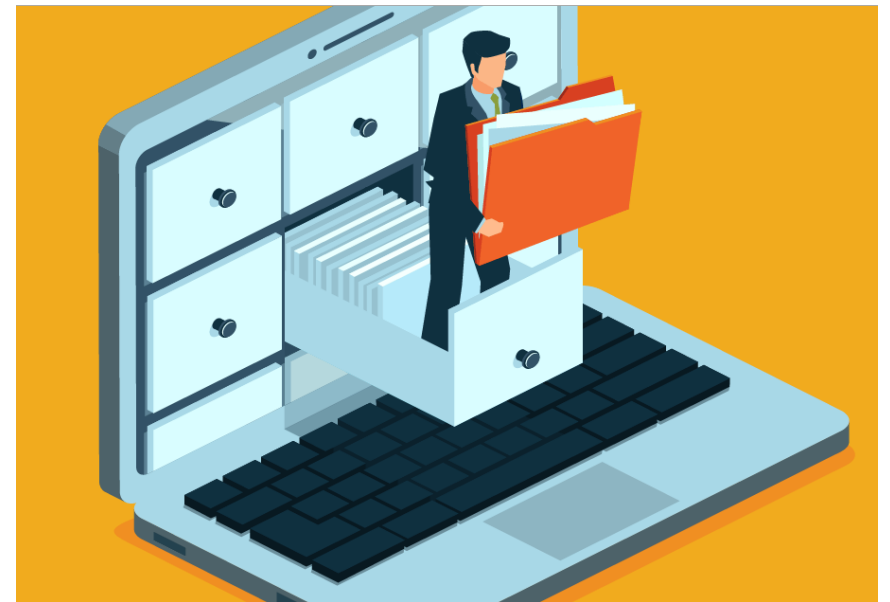
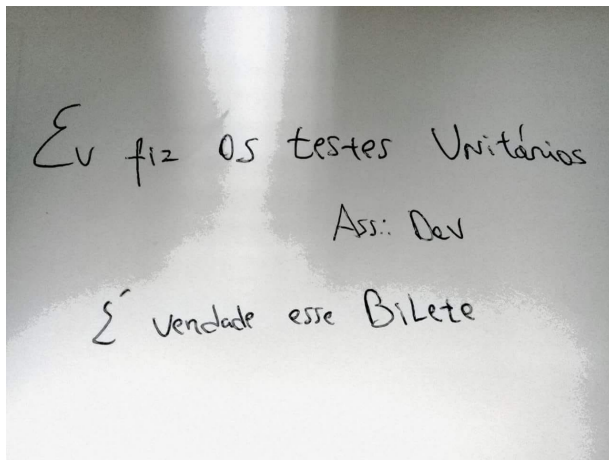


☐ Protege contra *regressão* de código

- ☐ Regressão ocorre se um *bug* for introduzido no código ao:
 - ☐ Refatorar
 - ☐ Corrigir outro *bug*
 - ☐ Nova funcionalidade
- ☐ Isto é, foi introduzido um erro em uma parte do código que *já funcionava*.
- ☐ Regrediu!



- ❑ Teste Unitário é uma ótima forma de documentação.
- ❑ Analisando as classes de teste, o desenvolvedor pode compreender o que as ***classes testadas fazem***.
- ❑ Ao se aproximar de um código novo, é boa prática que o desenvolvedor estude ***primeiro as classes de teste***!





- ❑ Testes Unitários são amplamente utilizados nas grandes empresas!



Microsoft



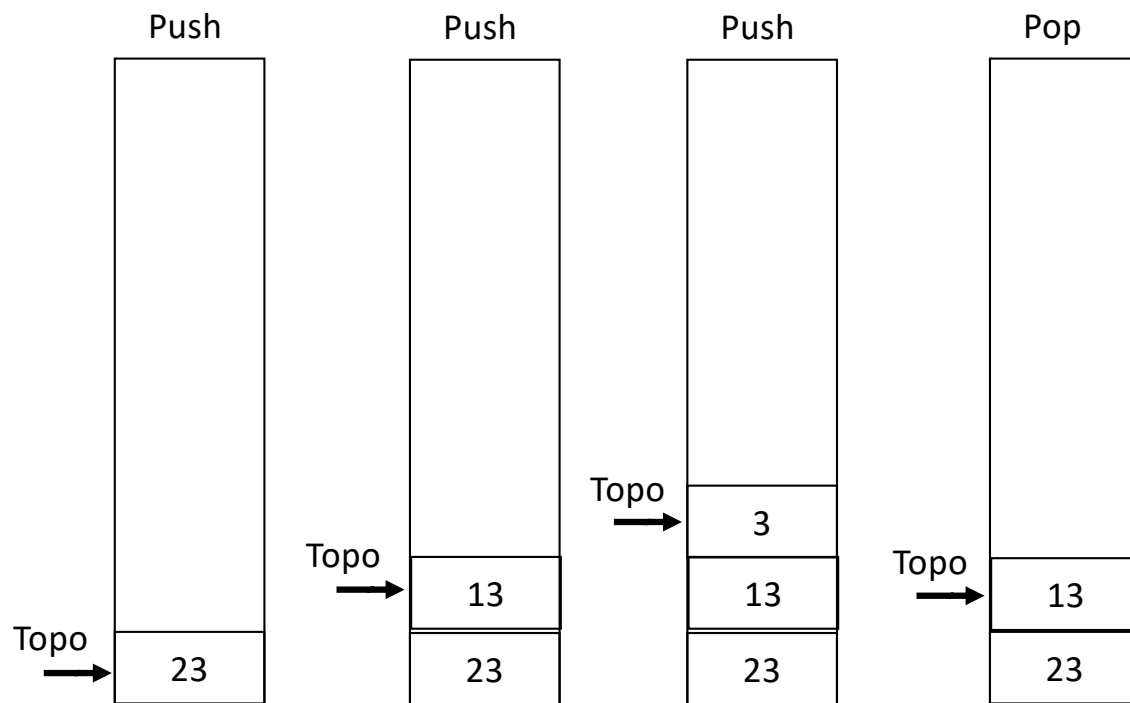
☐ Testar uma Pilha



Pilha (*Stack*) - Revisão

- ❑ A Pilha é uma estrutura de dados do tipo LIFO (***Last In First Out***). O último que entra é o primeiro a sair.
- ❑ Podemos fazer as seguintes operações na pilha:
 - ❑ Empilhar um elemento no topo(***push***)
 - ❑ Desempilhar em elemento do topo (***pop***)
 - ❑ Olhar o elemento do topo
 - ❑ Verificar se está vazia
- ❑ Observe que toda a manipulação é feita no ***topo*** da pilha.

Pilha (*Stack*) - Revisão



Resolução dos Exercícios/Exemplos

<https://github.com/phillima-unifei/COM221>





Aula – 3

Testes de Unidade

Disciplina: COM221 – Computação Orientada a Objetos II

Prof: Phyllipe Lima
phyllipe@unifei.edu.br

Universidade Federal de Itajubá – UNIFEI
IMC – Instituto de Matemática e Computação