

Trabalhos de Conclusão de Curso

Bacharelado em Ciência da Computação

Instituto de Matemática e Estatística

VERIFICAÇÃO & VALIDAÇÃO DE SOFTWARES DE MISSÃO CRÍTICA

2013

Camila Fernandez Achutti - nºUSP:6795610

Orientação: Ana Cristina Vieira de Melo

Cronograma

TEMAS

ESCOLHAS

IMPLEMENTAÇÃO

VALIDAÇÃO

VERIFICAÇÃO

CONSIDERAÇÕES FINAIS

VERIFICAÇÃO & VALIDAÇÃO DE SOFTWARES DE MISSÃO CRÍTICA

temas

VERIFICAÇÃO & VALIDAÇÃO DE SOFTWARES DE MISSÃO CRÍTICA

Estamos construindo o software corretamente?

Cuidado com as estruturas escolhidas, exceções, propriedades do sistema



temas

JPF (Java PathFinder)

NASA

verificador livre para programas Java

informa o "caminho" do erro

verificador de modelos

escolhas

VERIFICAÇÃO & VALIDAÇÃO DE SOFTWARES DE MISSÃO CRÍTICA

Estamos construindo o software correto?

O software deve atender
às necessidades dos usuários.
Obedecer seus requisitos.

Eclemma
JUnit

Eclemma & JUnit

facilidade de escrita de teste automatizado

ferramenta bastante visual para análise de cobertura de código

VERIFICAÇÃO & VALIDAÇÃO DE SOFTWARES DE MISSÃO CRÍTICA

Falhar não é uma opção!

Software Piloto Embarcado no Payload Data Handling Computer

SWPDC

real

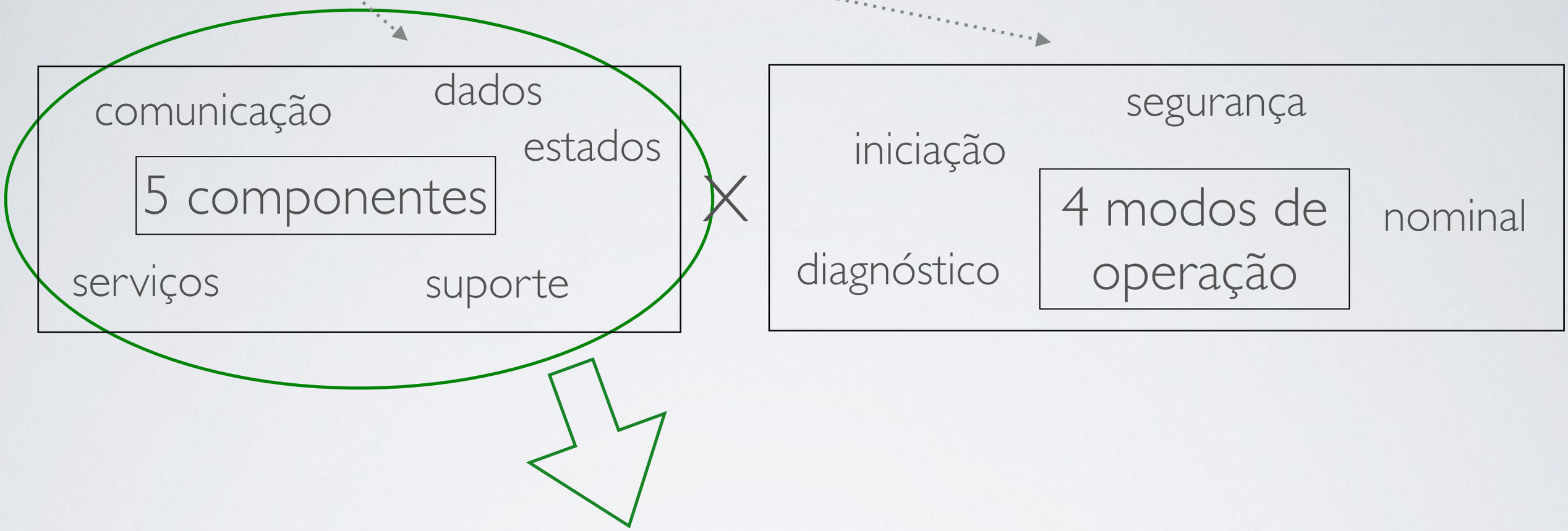
INPE
(QSEE)

especificação
extensa e
consistente

escolhas

IMPLEMENTAÇÃO 3

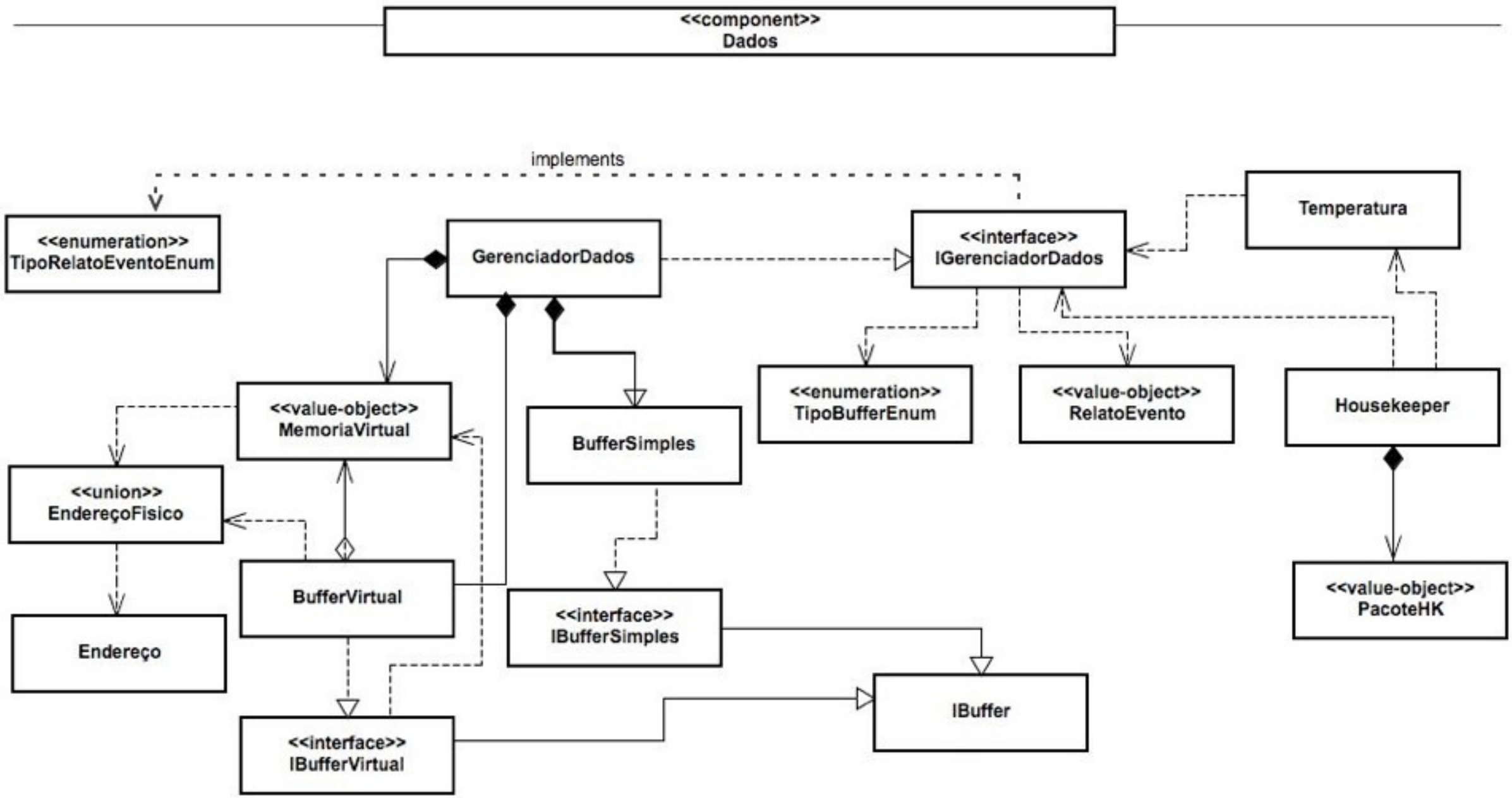
SWPDC



Componente: Dados

principal função: "gerenciamento" de temperaturas.

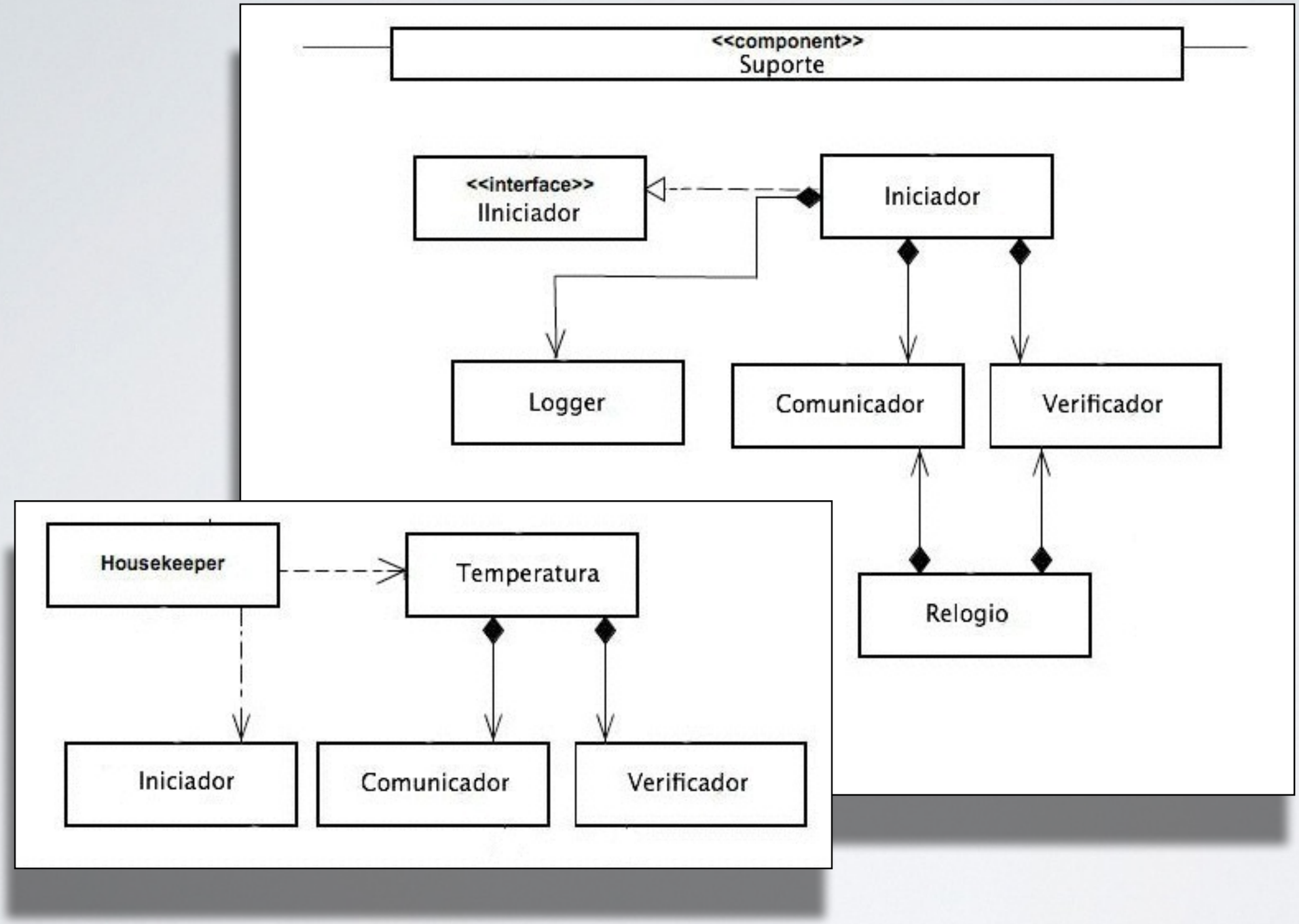
implementação



…mas estava faltando algo…

implementação

Escopo TEVE que aumentar



implementação

VALIDAÇÃO 3

Cenário

Conjunto de requisitos essenciais para o programa

1

Adaptação

adequação para a nossa realidade e escopo

2

Caso de Teste

quais são os casos de teste necessários para garantir tal requisito?

validação

Quando o SWPDC requisitar temperatura para o PDC. Transmissão encerrada com DATAEnd e bufferizada.

1

A transmissão de 12 amostras deve ser realizada a cada requisição do SWPDC e corretamente bufferizada.

2

5 casos de teste:

SRS005-1: AQUISIÇÃO BEM SUCEDIDA
SRS005-2: AQUISIÇÃO MAL SUCEDIDA POR ATRASO DO PDC
SRS005-3: AQUISIÇÃO MAL SUCEDIDA AMOSTRAGEM VAZIA
SRS005-4: AQUISIÇÃO MAL SUCEDIDA POR FALHA NO ARQ. DE AMOSTRAS
SRS005-5: AQUISIÇÃO MAL SUCEDIDA NUMERO ERRADO DE AMOSTRAS

validação

Detalhamento de caso de teste:

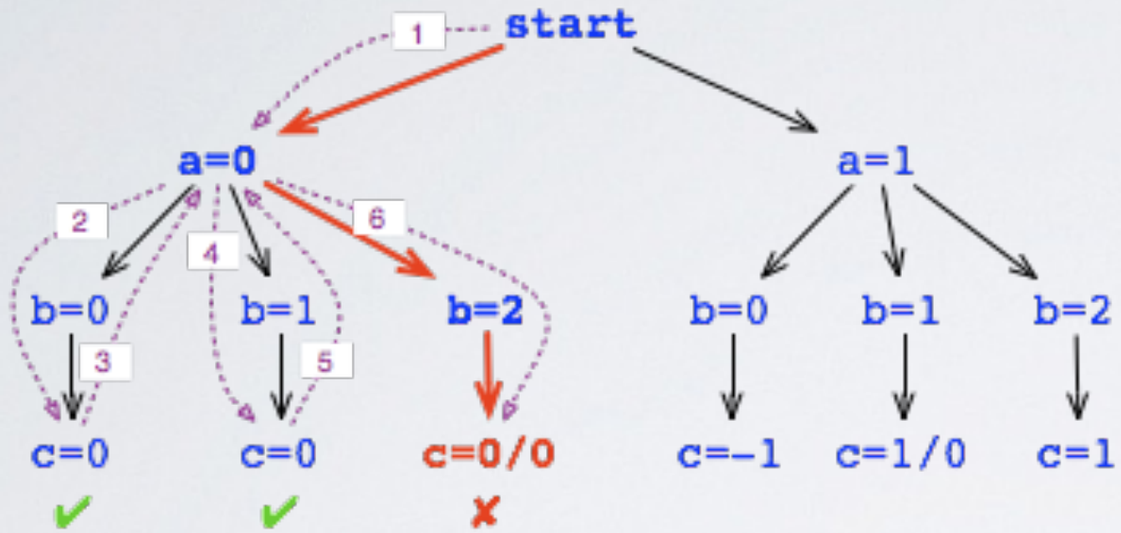
ID	SRS005-1
Nome	Aquisição de temperatura
Funicionalidade	AQUISIÇÃO BEM SUCEDIDA
Pré-condição	SWPDC em Nominal; Emissão de comando de aquisição; PDC em estado consistente.
Procedimento	Limpeza do buffer; Adquirir temperaturas; Verificar se cada amostra pertence ao intervalo; Armazenar as corretas no buffer.
Resultado Esperado	Armazenamento das amostras corretas no buffer

validação

VERIFICAÇÃO 3

JPF:

JVM que executa um programa de "todas as maneiras possíveis", verificando se há violações das propriedades estabelecidas



fonte:http://babelfish.arc.nasa.gov/trac/jpf/

BFS

DFS

Aleatório

Propriedades padrão:

```
gov.nasa.jpf.jvm.NotDeadlockedProperty
gov.nasa.jpf.jvm.NoAssertionViolatedProperty
gov.nasa.jpf.jvm.NoUncaughtExceptionsProperty
```

verificação

Saída:

```
JavaPathfinder v6.0 (rev ${version}) - (C) RIACS/NASA Ames Research Center

================================================== system under test
application: swpdc/SWPDC.java

================================================== search started: 10/24/13 1:13 PM
```

ERRO!

```
Camilas-MacBook-Air:src cachutti$ java -jar /Users/cachutti/projects/jpf-core/build/RunJPF.jar +shell.port=42
/Users/cachutti/NetBeansProjects/SWPDC/src/swpdc/SWPDC.jpf
JavaPathfinder v6.0 (rev ${version}) - (C) RIACS/NASA Ames Research Center


================================================== system under test
application: swpdc/SWPDC.java

================================================== search started: 10/24/13 12:41 PM

================================================== results
no errors detected

================================================== statistics
elapsed time:       00:00:01
states:             new=38, visited=0, backtracked=38, end=0
search:             maxDepth=38, constraints hit=0
choice generators:  thread=38 (signal=0, lock=1, shared ref=0), data=0
heap:               new=2905, released=707, max live=2245, gc-cycles=37
instructions:       191502
max memory:         81MB
loaded code:        classes=250, methods=3533

================================================== search finished: 10/24/13 12:41 PM
```

SUCESSO!

```
instructions:       191502
max memory:         81MB
loaded code:        classes=250, methods=3533

================================================== search finished: 10/24/13 1:13 PM
```

verificação

CONSIDERAÇÕES FINAIS

- **Desenvolvimento formal** de sistemas reais é, ainda, **incipiente na literatura**.

- **Avaliação do JPF** dentro da literatura de verificação formal de programas **se fazia necessária.**

- É necessário **deixar evidente que a combinação de técnicas de verificação e teste em conjunto provê ao software mais qualidade** se comparada ao uso de somente uma delas.

- **As ferramentas e técnicas existentes** de verificação e validação ainda **não são adequadas** para o uso da grande comunidade de desenvolvedores.

considerações finais

Perguntas?

Dúvidas?

Sugestões?

camila.achutti@gmail.com

Obrigada!

Referências:

[1]Java Pathfinder - http://babelfish.arc.nasa.gov/trac/jpf/

[2]Oracle: Java Technology - http://www.oracle.com/br/technologies/java/index.html. Acesso em: Jul. 2012

[3]Santiago Júnior, Valdivino Alexandre de: SOLIMVA: A Methodology for Generating Model-Based Test Cases from Natural Language Requirements and Detecting Incompleteness in Software Specications, 2011.

[4]Eclipse. http://www.eclipse.org/. Acesso em: Jul. 2012.