**Documentação Jantar dos Filósofos e problema do Papai Noel**

**Versão 1.0**

### Autores: Lucas Reolon e Vinícius Daniel

**2018**

**ÍNDICE**

[RESUMO 3](#_TOC_250046)

1. [INTRODUÇÃO AO DOCUMENTO 3](#_TOC_250045)
   1. [TEMA 3](#_TOC_250044)
   2. [OBJETIVO DO PROJETO 3](#_TOC_250043)
   3. [DELIMITAÇÃO DO PROBLEMA 3](#_TOC_250042)
   4. [MÉTODO DE TRABALHO 3](#_TOC_250040)
   5. [GLOSSÁRIO 3](#_TOC_250038)
2. [DESCRIÇÃO GERAL 4](#_TOC_250037)
   1. [DESCRIÇÃO DO PROBLEMA 4](#_TOC_250036)
3. ANÁLISE E DESIGN 5
   1. [DIAGRAMA DE CLASSES 5](#_TOC_250025)
   2. [DIAGRAMA DE ESTADOS 6](#_TOC_250023)
   3. [AMBIENTE DE DESENVOLVIMENTO 7](#_TOC_250017)
4. EXPLICAÇÃO DOS CÓDIGOS 7
   1. [PROBLEMA DO PAPAI NOEL 7](#_TOC_250017)
   2. [JANTAR DOS FILÓSOFOS 7](#_TOC_250017)

[BIBLIOGRAFIA 8](#_TOC_250006)

## Resumo

Esse documento contém um resumo sobre cada tecnologia utilizada para desenvolver os algoritmos do jantar dos filósofos e do problema do Papai Noel, incluindo diagramas de classe e de estado de cada um dos códigos.

## Introdução ao Documento

### Tema

O projeto tem como tema, o jantar dos filósofos e o problema do Papai Noel (Santa Claus problem). Que são problemas clássicos da computação concorrente.

### Objetivo do Projeto

O objetivo desse projeto, é aprofundar nossos conhecimentos em *Threads*, *deadlocks*, *starvation* e concorrência entre processos, utilizando a tecnologia Java.

### Delimitação do Problema

O problema se delimita ao uso de *Threads* para o gerenciamento de processos, que competem por recursos, sendo esses limitados pelas regras de cada caso.

### Método de Trabalho

Neste projeto são aplicadas tecnologias proprietárias do Java, como *Threads* e *Synchronized*, paradigma de programação orientado a objetos, NetBeans IDE, usamos também a *Unified Modeling Language ou* UML para descrever os diagramas de classe e estado.

### Glossário

Thread:

É o termo em inglês para Linha ou Encadeamento de Execução, *Thread* é um pequeno programa que trabalha como um subsistema, sendo uma forma de um processo se autodividir em duas ou mais tarefas.

Java:

Java é uma linguagem de programação orientada a objetos desenvolvida por uma equipe de programadores na empresa Sun Microsystems. A grande vantagem do Java são suas ferramentas prontas para trabalhar com *Threads*, conceito que já nasceu com a linguagem.

NetBeans:

O NetBeans IDE é um ambiente de desenvolvimento integrado gratuito e de código aberto para desenvolvedores de software.

Deadlock:

É a situação em que dois ou mais processos ficam impossibilitados de executarem suas funções, ou seja, ficam aguardando uns pelos outros.

## Descrição Geral do Sistema

### Problema do Papai Noel

O problema do Papai Noel é um problema clássico na computação, foi proposto por

Trono [1994], tendo como paradigma a programação concorrente. O problema é

basicamente composto por dez elfos, nove renas e pelo Papai Noel. A vida do Papai

Noel se resume a dormir até que ele seja acordado pelas nove renas ou por um grupo

de três elfos. Caso seja acordado pelas renas, o Papai Noel as amarra ao trenó, distribui

brinquedos, desamarra-as e volta a dormir. Caso seja acordado pelos elfos, o Papai

Noel discute projetos de brinquedos e volta a dormir. A vida de uma rena se resume a

entregar presentes e tirar férias até o próximo ano. A vida de um elfo se resume a

fabricar brinquedos e se reunir com o Papai Noel. Papai Noel dorme em sua casa no

Polo Norte, e só pode ser despertado por 9 renas (todas), quando estão voltando de

suas férias de alguma ilha tropical, ou por alguns elfos, que estão tendo dificuldades em

fazer os brinquedos. O problema de um elfo nunca é grave o suficiente para acordar o

Papai Noel (caso contrário, ele pode nunca conseguir dormir), então, os elfos visitam o

Papai Noel em um grupo de três. Quando os três elfos tem seus problemas resolvidos,

qualquer outro elfo que deseje visitar o Papai Noel deve esperar que os elfos retornem.

Se o Papai Noel acordar e encontrar três elfos esperando na porta de sua loja, junto

com a última rena voltando dos trópicos, o Papai Noel decide que os elfos podem

esperar até depois do Natal, porque é mais importante preparar seu trenó assim que

que possível. Supõe-se que as renas não querem deixar as férias, portanto, ficam lá até

o último momento possível.

### Jantar dos filósofos

Consiste em cinco filósofos que passam suas vidas meditando e comendo. Uma mesa

redonda é compartilhada entre os filósofos contendo cinco cadeiras. Cada Filósofo tem

um prato de espaguete. Para que o filósofo consiga comer ele precisa de 2 garfos. A

vida de cada filósofo consiste em períodos nos quais ele alternadamente come e pensa.

Quando ele medita ele não interage com seus colegas. Quando um filósofo sente fome

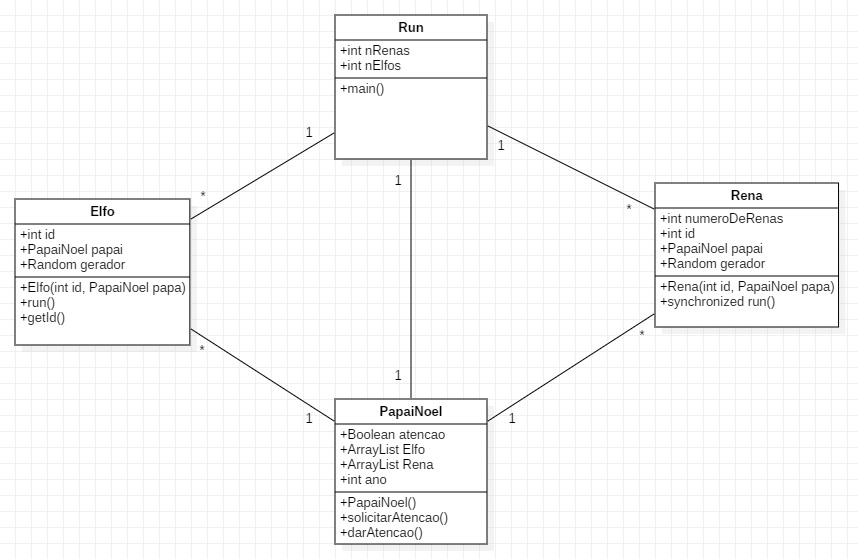
ele tenta pegar os dois garfos mais próximos de si (direita e esquerda do seu próprio

prato). Um filósofo pode pegar apenas um garfo de cada vez (a ordem não importa) e,

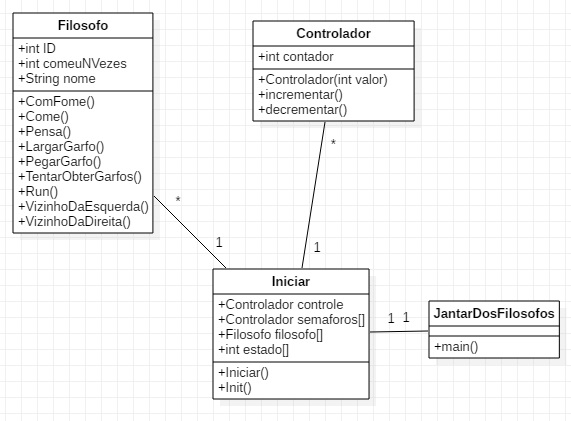
é claro, não poderá pegar nenhum garfo que esteja na mão de outro filósofo.

1. **Análise e *Design***

### Diagrama de Classes

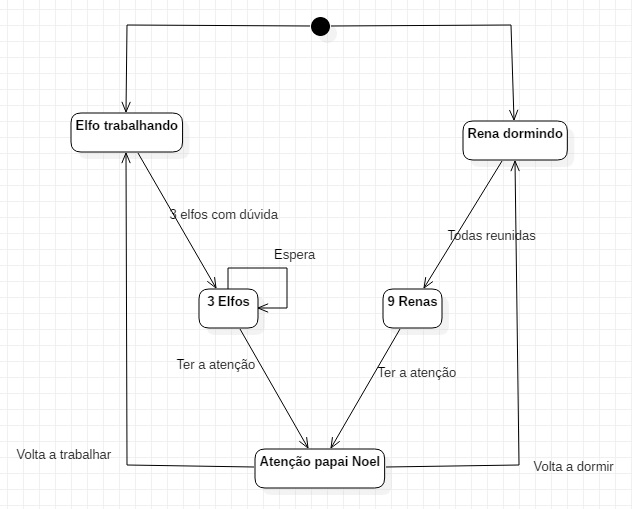
**

*Diagrama de classes problema do Papai Noel*

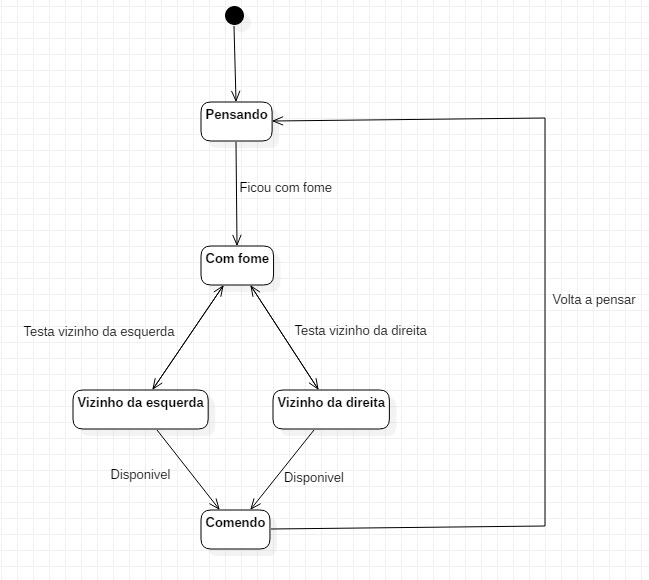
**

*Diagrama de classes jantar dos filósofos*

### Diagrama de Estados



*Diagrama de estados problema do Papai Noel*

****

*Diagrama de estados jantar dos filósofos*

### Ambiente de Desenvolvimento

Ambos algoritmos foram desenvolvidos na IDE NetBeans usando a tecnologia Java, em maquinas com Windows 10.

1. **Explicação dos códigos**
   1. **Problema do Papai Noel**

A classe Controlador controla o acesso aos recursos como a mesa e os garfos esquerdo e direito.

A classe JantarDosFilosofos chama a classe Iniciar que inicia as *threads* dos filósofos, irá definir também o nome e o ID de cada filósofo, e inicializará os controladores.

A classe Filosofo roda no *run*, aonde os estados dos filósofos iniciam pensando. Logo após o filósofo irá entrar na função de pegar o garfo, onde seu estado será alterado para “com fome”, e então ele tentará pegar o garfo esquerdo. Tendo êxito em pegar o garfo esquerdo, ele tentará pegar o garfo direito, e se assim conseguir, seu estado será alterado para “comendo”. Para pegar os garfos, o filósofo precisa checar os seus vizinhos esquerdo e direito, para ver se o vizinho não está comendo, e se não está com o garfo necessário em mãos. Após comer, o filósofo irá notificar os seus vizinhos para que se eles estiverem com fome, já possam estar tentando adquirir os garfos, e então retornará a pensar. Caso algum dos garfos não possa ser adquirido, o filósofo estará aguardando mais alguns instantes, e então tentará novamente adquirir os garfos. O filósofo estará ganhando maior prioridade para comer, caso esteja comendo menos que os outros filósofos.

* 1. **Jantar dos Filósofos**

São declarados o objeto Papai Noel, *threads* para as Renas e para os Elfos. Basicamente os elfos são iniciados quando criados e passam a vida a fabricar presentes, a cada período randômico tem dúvidas, duvidas essas que formam uma fila no objeto PapaiNoel. Quando a fila chega a três elfos suas dúvidas são solucionadas e eles voltam ao trabalho.

As renas também são iniciadas quando declaradas, passam grande parte do tempo dormindo quando chega o Natal eles vão até o PapaiNoel, o tempo do Natal é fixo, porém a um pequeno acréscimo de tempo randômico para que as renas acordem em tempos diferentes. O PapaiNoel fica sem fazer ações quando não há o número de renas ou de elfos suficientes em suas respectivas filas, seu acesso é limitado pelo método *synchronized*.

## 

## Bibliografia

Glossário - documentação SciELO Publishing Schema 1.6. Disponível em: < http://docs.scielo.org/projects/scielo-publishing-schema/pt\_BR/1.6-branch/glossary.html>. Acesso em: 18 set. 2018

UML – Wikipédia, a enciclopédia livre. Disponível em: < https://pt.wikipedia.org/wiki/UML >. Acesso em: 18 set. 2018

O que é Thread? Disponível em: < https://canaltech.com.br/produtos/o-que-e-thread/>. Acesso em: 18 set. 2018

Inanição (computação) – Wikipédia, a enciclopédia livre. Disponível em: < https://pt.wikipedia.org/wiki/Inani%C3%A7%C3%A3o\_(computa%C3%A7%C3%A3o)>Acesso em: 18 set. 2018

Introduction - StarUML documentation. Disponível em: < https://docs.staruml.io/>Acesso em: 18 set. 2018