

# **Especificação de Requisitos de Software**

**para o**

# **Simulador de Escalonamento de Tarefas**

**Versão 1.0**

**Realizado por:**

**Igor Martire**

**Kelly Maria Tavares**

**Maria Júlia Amorim**

**Mayara Tesch**

**Victor Olimpio**

**26/06/2015**

# Índice

## Índice

### Histórico de Revisões

#### 1. Introdução

- 1.1 Propósito
- 1.2 Convenções do Documento
- 1.3 Público-Alvo e sugestões de leitura
- 1.4 Escopo do Produto

#### 2. Descrição Geral

- 2.1 Perspectiva do Produto
- 2.2 Funções do Produto
- 2.3 Classes de Usuário e Características
- 2.4 Ambiente de Execução
- 2.5 Restrições de Design e Implementação
- 2.6 Documentação do Usuário
- 2.7 Suposições e Dependências

#### 3. Requisitos de Interface Externa

- 3.1 Interfaces com o usuário

#### 4. Funcionalidade do Sistema

- 4.1 Descrição
- 4.2 Sequência de Estímulo/Resposta
- 4.3 Requisitos Funcionais

#### 5. Outros Requisitos Não-Funcionais

- 5.1 Atributos de Qualidade de Software
- 5.2 Regras de Trabalho

#### Apêndice A: Glossário

#### Apêndice B: Modelos de Análise

## Histórico de Revisões

Nome	Data	Razão para mudanças	Versão
IM	20/03/2015	Documento inicial.	0.1
IM, KMT, MJA, MT	01/04/2015	Preenchimento dos tópicos 1 ao 4.	0.2
KMT, MT	15/05/2015	Preenchimento do tópico 5, exclusão dos tópicos 3.2, 3.3 e 3.4, modificação do tópico 3.1.	0.3
IM, MJA, VO	22/05/2015	Pequenas alterações nos tópicos 1.2, 1.3, 2.5 e 4.1.	0.4
MJA	23/06/2015	Retificação e conclusão do processo de execução do escalonador no item 4.3. Inclusão e detalhamento de dois novos algoritmos no item 5.2.	0.5
IM	26/06/2015	Revisão final. Atualização dos tópicos 2.2, 3.1, 4.1, 4.2 e 4.3 para incluir a especificação do uso de parâmetros. Corrigido o diretório onde é necessária a permissão de escrita, no tópico 2.4. No tópico 3.1, maior clarificação sobre o formato do arquivo de saída. Adicionado o atributo Legibilidade no tópico 5.1. Editadas as regras de trabalho no tópico 5.2	1.0

# **1. Introdução**

## **1.1 Propósito**

O propósito deste documento é o de apresentar uma descrição do produto que deve simular o processo de escalonamento de tarefas, sendo este descrito de forma integral, e não somente em parte. Aqui serão descritas as funcionalidades e interfaces do sistema, bem como o resultado esperado do mesmo.

## **1.2 Convenções do Documento**

Na seção Histórico da Revisão, no campo Nome, as abreviações referem-se aos nomes dos autores deste documento, de acordo com suas respectivas iniciais.

## **1.3 Público-Alvo e Sugestões de Leitura**

Este documento é direcionado ao Dr.-Ing. Raphael Pereira de Oliveira Guerra, professor da disciplina Técnicas de Projeto e Implementação de Sistemas de Informação I, lecionada no primeiro semestre de 2015 no Instituto de Computação da Universidade Federal Fluminense, no intuito de avaliar a qualidade de confecção do mesmo.

Também é direcionado aos desenvolvedores do sistema aqui especificado, a fim de que entendam as funcionalidades que devem ser atendidas e os formatos de comunicação com o usuário acertados.

Por fim, também tem como alvo o responsável pela implementação e suporte do software na máquina do usuário. Para este, é fundamental o bom entendimento das informações apresentadas no tópico 2.4.

Recomenda-se, para a leitura, seguir a sequência natural dos tópicos. Para uma compreensão rápida e objetiva da utilização do produto, recomenda-se o início no tópico 3 ou a leitura do manual do usuário referido no tópico 2.6.

## **1.4 Escopo do Produto**

O escopo do produto é servir como motivador para a construção de um documento de Especificação de Requisitos de Software, que é um dos objetivos da disciplina mencionada no tópico 1.3, além de ser o objeto do desenvolvimento de habilidades de codificação, de forma eficiente e organizada, de acordo com os ensinamentos do professor da disciplina.

## **2. Descrição Geral**

### **2.1 Perspectiva do Produto**

O software neste documento especificado é completo em si só, não sendo parte de um sistema maior. Sua origem é didática, referente à disciplina descrita no tópico 1.3 cujos objetivos estão descritos no tópico 1.4.

### **2.2 Funções do Produto**

O produto possui funcionalidades bem simples, sendo as três principais listadas a seguir:

- Leitura e processamento do arquivo e dos parâmetros de entrada.
- Simulação do escalonamento dos processos, de acordo com os parâmetros escolhidos.
- Escrita do resultado no arquivo de saída.

O fluxo dessas funcionalidades também é bem definida, como apresentado na imagem anexa no Apêndice B.

### **2.3 Classes de Usuários e Características**

Não há diferenciação entre os tipos de usuários que terão acesso ao sistema. Contudo, é necessário que todos os usuários estejam familiarizados com a forma de utilização do sistema definida no tópico 3, uma vez que o arquivo de entrada deve possuir um formato especificado para que o sistema possa interpretar corretamente os dados fornecidos pelo utilizador.

### **2.4 Ambiente de Execução**

Uma vez que o sistema será desenvolvido com o uso restrito aos recursos da linguagem Java, faz-se necessário o uso de um sistema que possua instalado o Java Runtime Enviroment em sua versão 7. Não há restrições quanto ao sistema operacional utilizado ou a plataformas de hardware específicas. É necessário também que haja a permissão de leitura ao arquivo de entrada passado para o software e que haja, também, a permissão de escrita no diretório corrente onde está sendo executado o programa.

## **2.5 Restrições de Design e Implementação**

Aos desenvolvedores é somente permitido o uso das bibliotecas e ferramentas nativas da linguagem Java em sua versão 7. Quaisquer bibliotecas não nativas à versão 7 do Java são terminantemente proibidas, uma vez que fogem ao detalhamento do ambiente de execução suficiente descrito no tópico 2.4. Além disso, é obrigatória a utilização de comentários de documentação, tal como usados pelo Javadoc.

## **2.6 Documentação do Usuário**

Junto com a disponibilização do software será fornecido um manual, em formato PDF, com as informações de uso do produto. O manual conterá as informações necessárias para que o usuário entenda como utilizar o sistema, bem como um exemplo de utilização do mesmo.

## **2.7 Suposições e Dependências**

É assumido que o software será executado em uma máquina que cumpra as necessidades do ambiente de execução descrito no tópico 2.4. Não haverá dependências externas, uma vez que, como tratado no tópico 2.5, todos os recursos utilizados serão nativos ao Java em sua versão 7.

# **3. Requisitos de Interface Externa**

## **3.1 Interface com o usuário**

Não haverá interface gráfica com o usuário. A utilização do software será feita mediante “linha de comando”. O usuário passa as informações necessárias para a aplicação através de um arquivo de texto e de parâmetros. Caso ocorra algum erro, uma mensagem será exibida no mesmo terminal em que foi iniciada a aplicação, através da saída padrão. Caso o programa seja finalizado com sucesso, a resposta do mesmo é salva em um arquivo texto.

O formato do conteúdo do arquivo de entrada é como se segue:

```
<id_processo>  
<tempo_execução_processo>  
<tempo_chegada_processo>  
<id_processo>  
<tempo_execução_processo>  
<tempo_chegada_processo>  
...
```

Nota-se que cada processo é definido em 3 linhas.

A primeira indica o id do processo, que deve ser único em todo o arquivo. São aceitos quaisquer caracteres alfanuméricos.

A segunda indica o tempo de execução do processo, em unidades de tempo, valor numérico, sem casas decimais.

A terceira indica o tempo de chega do processo, também em unidades de tempo, valor numérico, sem casas decimais.

A última linha do arquivo deve ser o tempo de chegada do último processo listado no arquivo.

Já o formato do conteúdo do arquivo de saída é apresentado a seguir:

Processo	T_Chegada	T_Saída	Terminado
<valor>	<valor>	<valor>	<valor>
<valor>	<valor>	<valor>	<valor>
...			

A primeira coluna indica o processo ao qual as informações da linha se referem. Este valor é o mesmo do campo <id\_processo> lido no arquivo de entrada.

A segunda coluna indica o tempo de chegada do processo à CPU, em unidades de tempo, do processo.

A terceira coluna indica o tempo de saída do processo da CPU, em unidades de tempo, do processo.

A quarta e última coluna indica se o processo foi finalizado (valor 1) ou não (valor 0) no tempo de saída especificado na coluna T\_Saída da mesma linha.

Os cabeçalhos não constarão no arquivo de saída e estão aqui presentes apenas para facilitar o entendimento.

## 4. Funcionalidade do Sistema

### 4.1 Descrição

A funcionalidade do sistema é simular o escalonamento dos processos definidos pelo usuário, de acordo com os parâmetros passados, gerando, no final, um arquivo de texto com o log do resultado do escalonamento, que deve informar os períodos de execução de cada processo. A prioridade dessa funcionalidade é máxima, uma vez que representa o total funcionamento do sistema.

### 4.2 Sequência de Estímulo/Resposta

O estímulo inicial é a própria chamada para execução do software. Para isso, o usuário deve informar o arquivo de entrada na “linha de comando” utilizada para iniciar a aplicação, bem

como, opcionalmente, a política de escalonamento a ser utilizada e se o programa deve mostrar o passo-a-passo do escalonamento. O formato do comando, seguido de um exemplo, é apresentado a seguir:

```
java -jar simEscTar.jar <caminho_para_o_arquivo_de_entrada> [(FCFS | SJN | SRT) [--verbose]]
```

Exemplo: `java -jar simEscTar.jar /home/user/input.txt SRT --verbose`

A resposta esperada é um log do escalonamento dos processos definidos no arquivo de entrada, tal qual feito através da política de escalonamento SRT (Shortest Remaining Time), além de mostrar, durante a execução do programa, o estado da simulação a cada passo.

Nota-se que os parâmetros após a especificação do arquivo de entrada são opcionais. Se forem omitidos, o sistema utilizará por padrão o parâmetro FCFS sem a opção `--verbose`.

Nota-se também que somente é possível especificar a opção `--verbose` se for especificada, antes, uma das políticas de escalonamento.

Caso ocorra algum erro durante a execução da funcionalidade, mensagens de erro serão disparadas informando o motivo.

### 4.3 Requisitos Funcionais

REQ-1: Leitura do arquivo de entrada

O sistema deve carregar os dados do arquivo de entrada especificado pelo usuário a fim de poder escalonar os processos por ele definidos.

Caso ocorra algum erro com relação à leitura do arquivo de entrada, o sistema apresentará uma mensagem de erro ao usuário e, logo em seguida, será finalizado. Estas mensagens são listadas e descritas a seguir:

- “Arquivo de entrada não especificado.”: não foi especificado o arquivo de entrada a ser utilizado.
- “Arquivo de entrada não encontrado.”: o arquivo de entrada especificado não existe.
- “Arquivo de entrada com formato de conteúdo inválido.”: o conteúdo do arquivo de entrada não segue o formato especificado no tópico 3.3.

REQ-2: Simulação do escalonamento dos processos

O sistema deve simular o escalonamento dos processos passados pelo usuário de acordo com os parâmetros passados pelo mesmo. À medida que for sendo feito o escalonamento, deve ir memorizando as informações sobre quais processos foram executados durante quais períodos de tempo e quando os mesmos foram terminados. Serão permitidas simulações com



o uso de 3 (três) algoritmos diferentes - First-Come, First-Served (FCFS), Shortest Job Next (SJN) e Shortest Remaining Time (SRT). Estes são explicados em mais detalhes na seção 5.2.

REQ-3: Escrita do resultado no arquivo de saída

O resultado da simulação do escalonamento dos processos será salvo no arquivo “output.txt”, localizado na mesma pasta onde o programa é executado. O formato do conteúdo do arquivo é descrito no tópico 3.3.

Caso ocorra algum erro com relação à escrita do arquivo de saída, o sistema apresentará uma mensagem de erro ao usuário e, logo em seguida, será finalizado.

## **5. Outros Requisitos Não-Funcionais**

### **5.1 Atributos de Qualidade de Software**

O software deve ter como atributos a Extensibilidade, que consiste na modificação de uma percentagem pequena de código caso seja necessário o acréscimo de alguma nova funcionalidade, Robustez, que consiste num tratamento abrangente e eficiente de exceções, e Legibilidade, que consiste na estruturação, organização e escrita do código de forma facilmente legível por outros desenvolvedores.

### **5.2 Regras de Trabalho**

O programa trabalhará com 3 (três) algoritmos de escalonamento, no entanto, o usuário deverá escolher qual política de escalonamento será utilizada. Isso é feito como visto na seção 4.2. A seguir, uma breve explicação do que se entende por cada política de escalonamento:

Na política FCFS, a priorização dos processos é por ordem de chegada. O processo que chegar primeiro será executado primeiro. É um algoritmo não-preemptivo, assim, o processo permanece executando na CPU mesmo que haja outro processo de maior prioridade esperando.

Já as políticas SJN e SRT priorizam o processo que acabará mais rápido. Se houver empate entre processos, é priorizado aquele que chegou primeiro na fila de espera. A diferença entre os dois algoritmos é que SRT é preemptivo, enquanto SJN não é. Ser preemptivo significa que pode interromper a atual execução de um processo para dar lugar a outro. No caso do SRT, se houver um processo na fila de espera com maior prioridade do que o que está atualmente em execução, este último deve ser interrompido e dar lugar ao de maior prioridade. Já no caso do SJN, isto não ocorre, e o processo que está atualmente em execução permanece na CPU.

## Apêndice A: Glossário

Dr.-Ing: Doktor-Ingenieur  
PDF: Portable Document Format  
FCFS: First-Come, First-Server  
SJN: Shortest Job Next  
SRT: Shortest Remaining Time

## Apêndice B: Modelos de Análise

Tópico 2.4 - Fluxo de uso do sistema

