

Projeto de Implementação: Gerenciador de Memória

Giuliana Leon
Guilherme Goulart
João Antonio Antunes
Thamires Sampaio

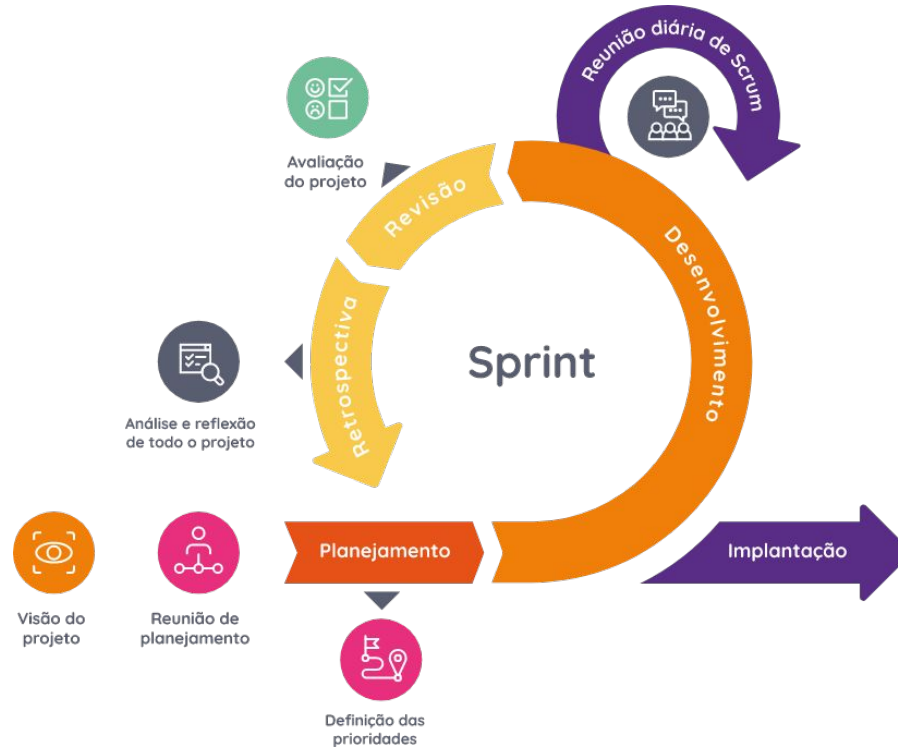
Objetivo

O objetivo desta apresentação é fornecer todas as informações necessárias sobre o desenvolvimento do software de Gerenciamento de Memória, utilizando os princípios da disciplina de Engenharia de Software, como a notação UML (Unified Modeling Language), os modelos de desenvolvimento de software, verificação e validação, diagrama de sequência, etc.

Método de Trabalho

- Metodologia Ágil;
- Método Scrum;
- Requisitos;
- UML;
- Cronograma de Atividades;
- Plano de Testes.

Metodologia Ágil



Método Scrum

1. Thamires Sampaio
2. Giuliana Leon
3. Guilherme Goulart
4. João Antonio Antunes
5. Thamires Sampaio



Cronograma de Atividades

1	SPRINT 1 - Elaboração do Projeto (SM Thamires)	<ul style="list-style-type: none"> - Backlog do produto - Consulta com Cliente - Definição dos Scrums 	ALTA
2	SPRINT 2 - Definição e Plano do Projeto (SM Giuliana)	<ul style="list-style-type: none"> - Reuniões de Equipe - Delimitação do Problema - Organização do Trabalho - Implementação Back-end (UML) - Revisão da documentação 	ALTA
3	SPRINT 3 - Execução do projeto (SM Guilherme)	<ul style="list-style-type: none"> - Reuniões de Equipe - Implementação Back-end - Teste de Aplicação - Revisão da documentação 	ALTA
4	SPRINT 4 - Controle e desempenho do projeto (SM João)	<ul style="list-style-type: none"> - Reuniões de Equipe - Implementação Back-end - Teste de Aplicação - Revisão da documentação 	ALTA
5	SPRINT 5 - Interface e fechamento do projeto (SM Thamires)	<ul style="list-style-type: none"> - Implementação Front-end - Testes Finais - Relatório 	ALTA

Descrição Geral do Sistema

O projeto tem como propósito a criação de um software que simula o funcionamento de um Gerenciador de Memória dinâmico, fazendo alocações e desalocações tanto de forma sequencial quanto de forma paralela, de acordo com a necessidade do usuário.

Requisitos Funcionais

Quanto ao usuário:

- O usuário deve ser capaz de configurar o tamanho da heap;
- O usuário deve ser capaz de configurar o número de requisições;
- O usuário deve ser capaz de configurar o valor mínimo e máximo da variável dinâmica a ser alocada na heap;
- O usuário deve ser capaz de configurar a porcentagem mínima e máxima de ocupação da heap.

Requisitos Funcionais

Quanto ao funcionamento do sistema:

- O algoritmo de alocação deve alocar a memória para cada variável na heap sempre de forma contígua;
- A execução do desalocador deve se iniciar sempre que a porcentagem de espaço ocupado estiver acima do percentual máximo informado pelo usuário;
- O algoritmo de desalocação deve executar de forma paralela com o alocador;
- O desalocador deve liberar espaço até que o percentual de memória livre fique igual ou abaixo da porcentagem mínima informada;

Requisitos Funcionais

Quanto ao funcionamento do sistema:

- O vetor deve ser implementado como uma fila circular;
- Cada requisição deve informar ao alocador o tamanho da variável dinâmica a ser alocada, bem como seu respectivo identificador;
- Um gerador de requisições randômicas deve alimentar o vetor de requisições;
- É necessário implementar um algoritmo de compactação para eliminar a fragmentação externa.

Requisitos Não Funcionais

- O sistema deverá ser desenvolvido na linguagem Java.
- É necessário o uso de algum mecanismo de controle para proteger as seções críticas de dados.
- A aplicação deve rodar em qualquer máquina compatível com a linguagem utilizada.
- O sistema deverá processar n requisições no menor tempo possível.

Protótipo da Interface

GERENCIADOR DE MEMÓRIA

Configurações Heap:

JLabel

Tamanho da Heap

JTextField1

JLabel

Percentual Máximo

JTextField2

JLabel

Percentual Mínimo

JTextField3

Configurações Vetor Requisição:

JLabel

Número de Requisições

JTextField4

Configurações Requisição:

JLabel

Tamanho Mínimo

JTextField5

JLabel

Tamanho Máximo

JTextField6

Adicionar Dados

Jbutton1

Informações Extras:

JLabel

Ver Heap

Jbutton2

JLabel

Ver Ocupados

Jbutton3

JLabel

Ver Vetor de Requisições

Jbutton4

Executar em modo:

JLabel

Paralelo

Jbutton5

JLabel

Sequencial

Jbutton6

Protótipo da Interface

GERENCIADOR DE MEMÓRIA

Configurações Heap:

Tamanho da Heap

Percentual Máximo

Percentual Mínimo

Configurações Vetor Requisição:

Número de Requisições

Configurações Requisição:

Tamanho Mínimo

Tamanho Máximo

Adicionar Dados

Informações Extras:

Ver Heap

Ver Ocupados

Ver Vetor de Requisições

Executar em modo:

Paralelo

Sequencial

Instruções JTextField

JTextField1: Armazenará a informação do tamanho da heap.

JTextField2: Armazenará a informação de percentual máximo de ocupação da heap.

JTextField3: Armazenará a informação de percentual mínimo de ocupação da heap.

JTextField4: Armazenará a informação de quantas requisições serão atendidas ao total.

JTextField5: Armazenará o tamanho mínimo da variável a ser alocada de cada requisição.

JTextField6: Armazenará o tamanho máximo da variável a ser alocada de cada requisição.

Protótipo da Interface

GERENCIADOR DE MEMÓRIA

Configurações Heap:

JLabel

Tamanho da Heap

JTextField1

Percentual Máximo

JTextField2

Percentual Mínimo

JTextField3

Configurações Vetor Requisição:

JLabel

Número de Requisições

JTextField4

Configurações Requisição:

JLabel

Tamanho Mínimo

JTextField5

Tamanho Máximo

JTextField6

Adicionar Dados

JButton1

Informações Extras:

JLabel

Ver Heap

JButton2

Ver Ocupados

JButton3

Ver Vetor de Requisições

JButton4

Executar em modo:

JLabel

Paralelo

JButton5

Sequencial

JButton6

Instruções JButton

JButton1: Pegará as informações dadas pelo usuário e as aloca em suas respectivas variáveis. OBS: JButton1 apenas funcionará se o usuário preencher todos os campos e o tamanho máximo da requisição não for maior que o tamanho da heap.

JButton2: Irá imprimir as informações da heap, mais especificamente seus 0's e 1's (0 nas posições livres e 1 nas posições ocupadas).

JButton3: Irá imprimir o vetor de ocupados, organizado por ordem decrescente e mostrando as seguintes informações: Início (endereço da página que ele iniciou a alocação), tamanho (tamanho da variável alocada, quantos blocos ela está ocupando) e ID (identificador da requisição alocada).

JButton4: Irá imprimir o vetor de requisições.

JButton5: Executará o programa em modo paralelo, pode executar apenas se JButton1 tiver sido utilizado de forma correta ao menos 1 vez.

JButton6: Executará o programa em modo sequencial, pode executar apenas se JButton1 tiver sido utilizado de forma correta ao menos 1 vez.

14


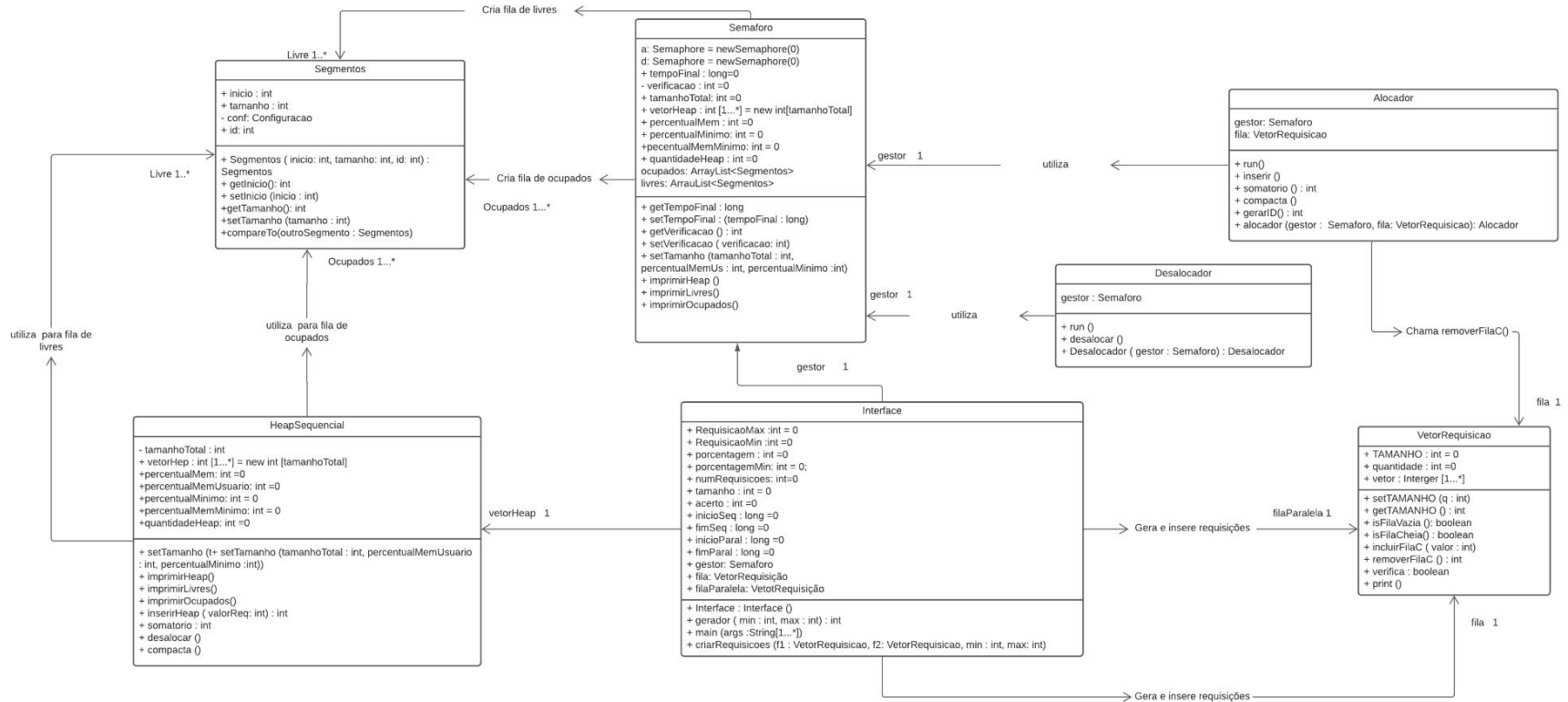
 **unipampa** Universidade Federal do Pampa

Diagrama de Classes



Plano de Testes

- Tabela de Requisitos;
- Detalhamento da abordagem de testes;
- Casos de Testes.

Plano de Testes

Requisito	Prioridade	Descrição
RF001	Essencial	O usuário do sistema deve ser capaz de configurar o tamanho da heap para funcionamento do sistema.
RF002	Essencial	O usuário do sistema deve ser capaz de configurar o intervalo de valores (mínimo e máximo) para geração randômica do tamanho das variáveis a cada requisição produzida pelo gerador.
RF003	Essencial	O algoritmo de alocação deve alocar a memória para cada variável na heap sempre de forma contígua.

Plano de Testes

Tipo de teste:	Funcional
Subtipo de teste:	Requisitos
Requisitos que motivaram:	RF001
Objetivo do teste:	Testar se o tamanho da heap informado pelo usuário está sendo armazenado corretamente em sua respectiva variável.

Tipo de teste:	Funcional
Subtipo de teste:	Requisitos
Requisitos que motivaram:	RF001
Objetivo do teste:	Testar se a heap está realmente sendo gerada com o tamanho especificado.

Plano de Testes

Caso de uso	ID	Passos	Resultado Esperado
UC001 - Testar se a heap está realmente sendo gerada com o tamanho especificado (RF001).	1	Na interface do programa, coloque o tamanho da heap em 10 e preencha os outros dados normalmente.	
	2	Clique em "Adicionar dados" e após isso clique em "imprimir heap".	Tanto a impressão da heap paralela quanto da heap sequencial devem mostrar que suas páginas vão de 0 a 9.

Manual do Usuário

Gerenciador de Memória

GERENCIADOR DE MEMORIA

Configurações Heap:

TAMANHO DA HEAP

PERCENTUAL MÁXIMO

PERCENTUAL MÍNIMO

Configurações Vetor Requisição:

NÚMERO DE REQUISIÇÕES

Configurações Requisição

TAMANHO MÍNIMO

TAMANHO MÁXIMO

ADICIONAR DADOS

Informações Extras:

VER HEAP

VER OCUPADOS

VER VETOR DE REQUISIÇÕES...

EXECUTAR EM MODO

PARALELO SEQUENCIAL

Materiais Utilizados

- Google Drive;
- Google Docs;
- Google Apresentações;
- Eclipse;
- GitHub;
- Lucid;
- Discord;
- Grupo no Whatsapp.

Entrega

- Gerenciador de Memória Dinâmico;
- Documentação do Gerenciador;
- Código comentado;
- Backlog da aplicação.