

UNIVERSIDADE ESTADUAL DO NORTE FLUMINENSE  
LABORATÓRIO DE ENGENHARIA E EXPLORAÇÃO DE PETRÓLEO

PROJETO DE ENGENHARIA  
SIMULADOR DE ESTIMATIVA DE RESERVA DE RESERVATÓRIO  
UTILIZANDO EBM,  
COM ENFOQUE NOS RESERVATÓRIOS DE GÁS

TRABALHO DA DISCIPLINA PROGRAMAÇÃO PRÁTICA

Versão 1:

LEONARDO MESQUITA CAETANO - 2012  
THIAGO PESSANHA DE MACEDO - 2012

Versão 2:

JALILE BARTOLE PEREIRA - 2014  
THAÍSA DE CASTRO ALEIXO - 2014

Versão 3:

IGOR BENJAMIN DE SOUZA - 2023  
MATHEUS RODRIGUES NASCIMENTO - 2023

Prof. André Duarte Bueno

MACAÉ - RJ  
Dezembro - 2023

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Escopo do problema . . . . .	1
1.2	Objetivos . . . . .	2
1.3	Objetivos - V2 . . . . .	2
1.4	Objetivos - V3 . . . . .	2
1.5	Metodologia utilizada . . . . .	3
<b>2</b>	<b>Concepção</b>	<b>5</b>
2.1	Nome do sistema/produto . . . . .	5
2.2	Especificação . . . . .	5
2.2.1	Diagrama de caso de uso específico . . . . .	7
2.3	Requisitos . . . . .	7
2.3.1	Requisitos funcionais . . . . .	7
2.3.2	Requisitos não funcionais . . . . .	10
2.4	Casos de uso . . . . .	10
2.4.1	Diagrama de caso de uso geral . . . . .	11
2.4.2	Diagrama de caso de uso específico . . . . .	11
<b>3</b>	<b>Elaboração</b>	<b>17</b>
3.1	Análise de domínio . . . . .	17
3.2	Formulação teórica . . . . .	19
3.3	Descrição dos mecanismos de produção para reservatório de gás . . . . .	21
3.3.1	Reservatórios de gás seco . . . . .	21
3.3.2	Reservatórios de gás condensado . . . . .	23
3.4	Descrição dos mecanismos de produção para reservatório de óleo . . . . .	24
3.5	Referências Bibliográficas . . . . .	25
3.6	Identificação de pacotes – assuntos . . . . .	25
3.7	Diagrama de pacotes – assuntos . . . . .	26
<b>4</b>	<b>AOO – Análise Orientada a Objeto</b>	<b>27</b>
4.1	Diagramas de classes . . . . .	27
4.1.1	Dicionário de classes . . . . .	27

4.2	Diagramas de classes . . . . .	27
4.2.1	Dicionário de classes . . . . .	27
4.3	Diagrama de seqüência – eventos e mensagens . . . . .	34
4.3.1	Diagrama de sequência geral . . . . .	34
4.3.2	Diagrama de sequência específico . . . . .	35
4.4	Diagrama de comunicação – colaboração . . . . .	36
4.5	Diagrama de estado . . . . .	37
4.6	Diagrama de atividades . . . . .	38
<b>5</b>	<b>Projeto</b>	<b>39</b>
5.1	Projeto do sistema . . . . .	39
5.2	Projeto orientado a objeto – POO . . . . .	43
5.3	Diagrama de componentes . . . . .	47
5.4	Diagrama de implantação . . . . .	48
5.4.1	Lista de características <<features>> . . . . .	49
5.4.2	Tabela classificação sistema . . . . .	50
<b>6</b>	<b>Ciclos de Planejamento/Detalhamento</b>	<b>52</b>
6.1	Versão 0.1 - xxx . . . . .	52
6.2	Versão 0.2 - xxx . . . . .	52
<b>7</b>	<b>Ciclos Construção - Implementação</b>	<b>54</b>
7.1	Código fonte . . . . .	54
<b>8</b>	<b>Teste</b>	<b>58</b>
8.1	Teste 1: Descrição . . . . .	58
8.2	Teste 2: Descrição . . . . .	59
8.3	Teste 3: Descrição . . . . .	60
<b>9</b>	<b>Documentação para o Desenvolvedor</b>	<b>62</b>
9.1	Dependências para compilar o software . . . . .	62
9.2	Como gerar a documentação usando doxygen . . . . .	62
<b>A</b>	<b>Título do Apêndice</b>	<b>66</b>
A.1	Sub-Título do Apêndice . . . . .	66
<b>B</b>	<b>Usando <math>\text{\TeX}</math>/<math>\text{\LaTeX}</math>/<math>\text{\LyX}</math></b>	<b>67</b>
B.1	Pacotes a serem instalados . . . . .	67
B.1.1	Windows . . . . .	67
B.1.2	GNU/Linux . . . . .	67
B.2	Inclusão de Figuras . . . . .	69
B.3	Inclusão de Equações . . . . .	72

B.4 Inclusão de Tabelas . . . . .	72
B.5 Inclusão de Listagens de Código . . . . .	73
B.6 Roteiro Para Uso do Sistema de Citações Com Banco de Dados .bib . . . . .	73
B.6.1 Citações no meio do texto . . . . .	76
B.6.2 Citações no início do texto . . . . .	76
B.6.3 Citações tipo apud . . . . .	76
B.6.4 Incluir nas referências bibliográficas (fim do documento), mas não citar . . . . .	76
B.7 Informações Adicionais . . . . .	77

# Listas de Figuras

1.1	Etapas para o desenvolvimento do software - <i>projeto de engenharia</i> . . . . .	4
2.1	Diagrama de especificação do Software . . . . .	6
2.2	Diagrama de Regressão Linear . . . . .	7
2.3	Diagrama de Modelagem de Reservatório de Gás . . . . .	8
2.4	Diagrama de Modelagem de Injeção de Fluido . . . . .	9
2.5	Diagrama de caso de uso – Caso de uso geral . . . . .	12
2.6	Diagrama de caso de uso específico – Escolha do Mecanismo de Produção .	13
2.7	Diagrama de caso de uso específico – Selecionar Método de Injeção . . . . .	14
2.8	Diagrama de caso de uso específico – Salvar Relatório . . . . .	15
2.9	Diagrama de caso de uso específico – Gerar Gráfico . . . . .	16
3.1	Diagrama de Pacotes . . . . .	26
4.1	Diagrama de classes . . . . .	28
4.2	Diagrama de seqüência . . . . .	35
4.3	Diagrama de sequência específico . . . . .	36
4.4	Diagrama de comunicação . . . . .	37
4.5	Diagrama de máquina de estado . . . . .	38
4.6	Diagrama de atividades . . . . .	38
5.1	Diagrama de componentes . . . . .	48
5.2	Diagrama de implantação . . . . .	49
6.1	Versão 0.1, imagem do programa rodando . . . . .	52
6.2	Versão 0.2, imagem do programa rodando . . . . .	53
8.1	Tela do programa mostrando xxx . . . . .	59
8.2	Tela do programa mostrando xxx . . . . .	60
8.3	Tela do programa mostrando xxx . . . . .	61
B.1	Etapas fundamentais no processamento digital de imagens, aplicadas ao estudo de lâminas de meios porosos [adaptado de Gonzales e Woods (1992)]	71
B.2	Titulo da figura. Em (a) blablabla; em (b) blablabla . . . . .	71

# **Lista de Tabelas**

2.1	Caso de uso 1	11
B.1	Exemplo de tabela flutuante.	73
B.2	Exemplo de tabela flutuante - coluna com dimensão fixa e rotacionada.	78

# Listagens

7.1	Arquivo de cabeçalho da classe CSimulador-nome . . . . .	54
7.2	Arquivo de implementação da classe CSimulador-nome . . . . .	56
7.3	Arquivo de implementação da função <code>main()</code> . . . . .	57
..../src/	Simulador-nome.out . . . . .	57

# Capítulo 1

## Introdução

No presente trabalho desenvolve-se um simulador de estimativa de reserva de reservatório utilizando EBM, com enfoque nos reservatórios de gás, um software desenvolvido usando-se a linguagem C++<sup>1</sup> e paradigma de orientação a objetos, que tem como principal objetivo implementar as equações de balanço de materiais vistas na disciplina de Engenharia de Reservatório para os diferentes mecanismos de produção de petróleo. Dessa forma, a principal finalidade do programa é fornecer uma estimativa do volume original de óleo ou gás do reservatório. **Além dos mecanismos de produção convencionais, o simulador também é capaz de modelar a injeção de água e gás ou uma mistura de fluidos.** Também é dado um enfoque em reservatórios de gás e gás condensado, incluindo, por exemplo, o seu comportamento quando há influxo de água, e alguns modelos para o cálculo deste influxo

### 1.1 Escopo do problema

Na indústria do petróleo é fundamental ter uma estimativa da reserva de óleo ou gás de um campo. Além disso, é importante ter uma previsão do comportamento do reservatório antes de começar a produzir e durante toda a sua vida produtiva. Tais estimativas são importantes por diversos fatores, tais como: Estudo de Viabilidade Técnica e Econômica (EVTE), declaração de comercialidade para a ANP, tomada de decisões estratégicas (investir ou abandonar um campo).

Para fazer as estimativas de uma acumulação petrolífera se faz uso da Equação de Balanço de Materiais (EBM), que é um balanço de massas dos fluidos existentes no interior dos poros das rochas-reservatório. Suas principais utilizações práticas são a determinação do volume original de óleo, a determinação do volume original de gás e a previsão do comportamento do reservatório.

---

<sup>1</sup>A linguagem utilizada é uma linguagem orientada a objeto, que utiliza o conceito de classes, o que facilita na criação de programas complexos, como os de engenharia. É uma linguagem de alto nível, fortemente tipada. É amplamente utilizada para montagem de sistemas operacionais como Windows, GNU/Linux, e programas como o Lyx, Office e OpenOffice.

## 1.2 Objetivos

Os objetivos da primeira versão do trabalho são:

- Objetivo geral:
  - Desenvolver uma solução para determinar o volume original de fluido no reservatório.
- Objetivos específicos:
  - Selecionar o tipo de fluido presente no reservatório.
  - Selecionar o mecanismo de produção.

## 1.3 Objetivos - V2

Os objetivos deste trabalho são:

- Objetivo geral:
  - Desenvolver uma solução para determinar o volume original de fluido (principalmente gás) no reservatório.
- Objetivos específicos:
  - Selecionar o tipo de fluido presente no reservatório (gás).
  - Selecionar o mecanismo de produção.
  - Quando houver influxo de água, escolher o modelo que será utilizado para o cálculo deste influxo.

## 1.4 Objetivos - V3

- Objetivo geral:
  - Aperfeiçoar a solução para determinar o volume original de fluido (principalmente gás) no reservatório, incluindo a modelagem de diferentes mecanismos de produção e a injeção de fluidos.
- Objetivos específicos:
  - Implementar classes e métodos para representar a injeção de água, gás, polímero, surfactante, vapor e gás lift no reservatório.

- Aprimorar a flexibilidade do programa para lidar com uma variedade de cenários de produção e injeção.
- Refinar a lógica de cálculo do volume original de fluido para considerar os efeitos das novas classes de injeção.

As EBM foram implementadas no segundo semestre de 2012 pelos alunos Leonardo Mesquita Caetano e Thiago Pessanha de Macedo. No segundo semestre de 2014, Jalile Bartole Pereira e Thaís de Castro Aleixo fizeram a ampliação deste código, adicionando novas premissas para a determinação do volume original de gás: considerando-se reservatórios de gás condensado (retrógrado e não retrógrado), analisando-se a influência dos aquíferos e detalhando o comportamento de alguns modelos de influxo. **Nesta versão do projeto, o código foi estendido para incluir a modelagem da injeção de fluidos ao reservatório, aumentando a gama de cenários de simulação da produção para determinar o volume original de óleo e gás em reservatórios que estão sendo submetidos a injeção de fluidos.**

## 1.5 Metodologia utilizada

O software a ser desenvolvido utiliza a metodologia de engenharia de software apresentada pelo Prof. André Bueno na disciplina de programação e ilustrado na Figura 1.1. Note que o “Ciclo de Concepção e Análise” é composto por diversas partes representadas neste trabalho em diferentes capítulos. Os ciclos de planejamento/detalhamento tem seu próprio capítulo, assim como o ciclo de construção - implementação.

Esta metodologia é utilizada nas disciplinas:

- LEP01447 : Programação Orientada a Objeto em C++.
- LEP01446 : Programação Prática.

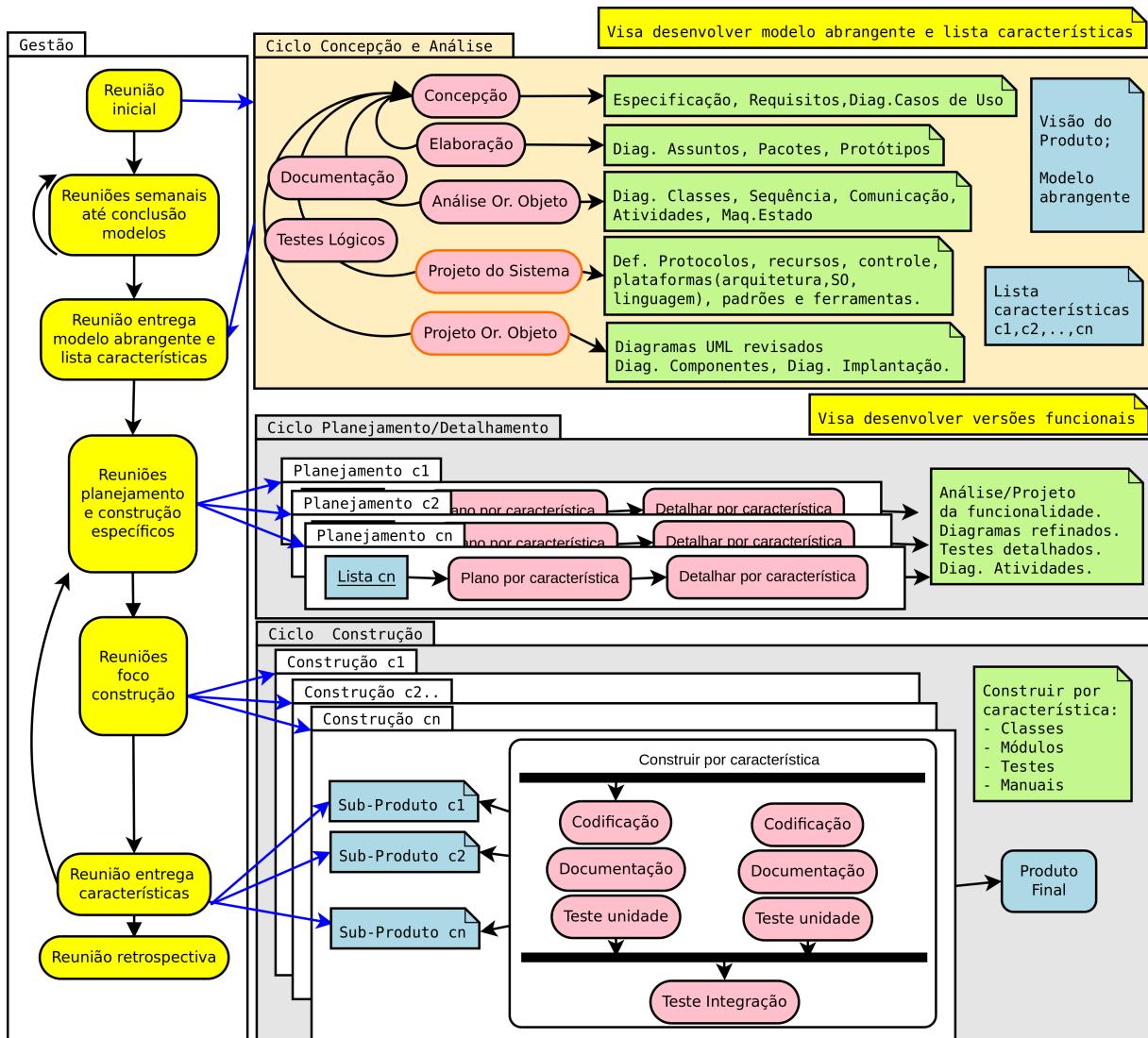


Figura 1.1: Etapas para o desenvolvimento do software - *projeto de engenharia*

# Capítulo 2

## Concepção

Apresenta-se neste capítulo do projeto de engenharia a concepção, a especificação do sistema a ser modelado e desenvolvido.

### 2.1 Nome do sistema/produto

Nome	Simulador de Reservatório (SimRes)
Componentes principais	Regressão Linear Conceitos da Engenharia de Reservatório Conceitos dos Mecanismos de Produção de Óleo Conceitos dos Mecanismos de Produção de Gás Métodos de injeção
Missão	Fornecer uma plataforma para simulações de estimativa de reservas usando EBM na área de engenharia de reservatórios.

### 2.2 Especificação

Apresenta-se nesta seção a especificação detalhada do software de Simulação de Estimativa de Reservas.

O software tem como objetivo principal realizar simulações determinar os volumes originais de gás e óleo, vai salvar os dados analisados do histórico de produção e gerar um gráfico com o auxílio do programa externo Gnuplot., considerando diversos mecanismos de produção e métodos de injeção.

O diagrama especificação do software da Figura 2.1 mostra o que se pretente (primeira ideia) do software.

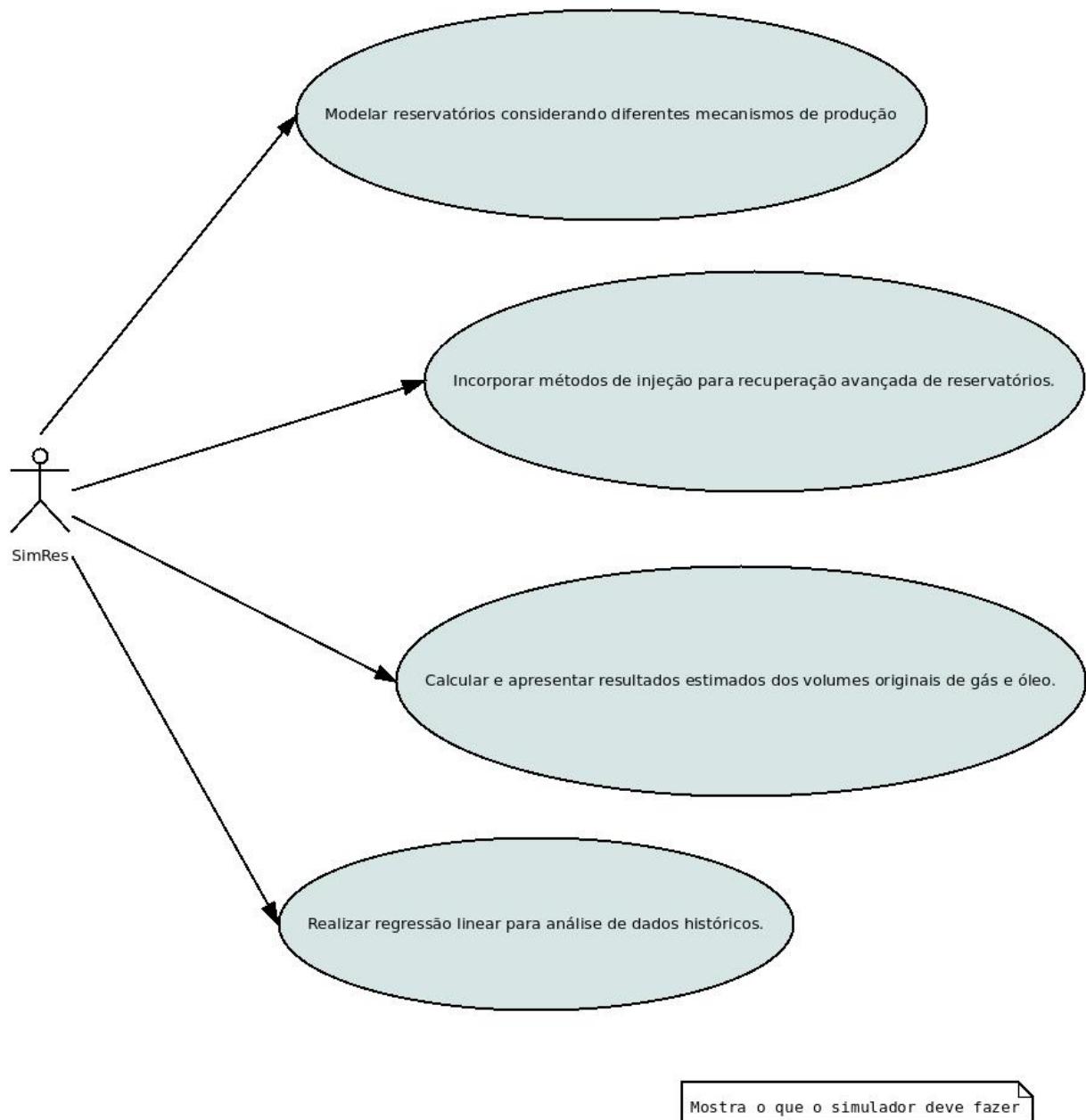


Figura 2.1: Diagrama de especificação do Software

### 2.2.1 Diagrama de caso de uso específico

A seguir, são apresentados casos de uso específicos do software, detalhando as principais interações entre o usuário e o sistema.

O diagrama de regressão linear da Figura 2.2 mostra o usuário utilizando o Simulaor de reservatório para análise de tendências e padrões dos dados históricos.

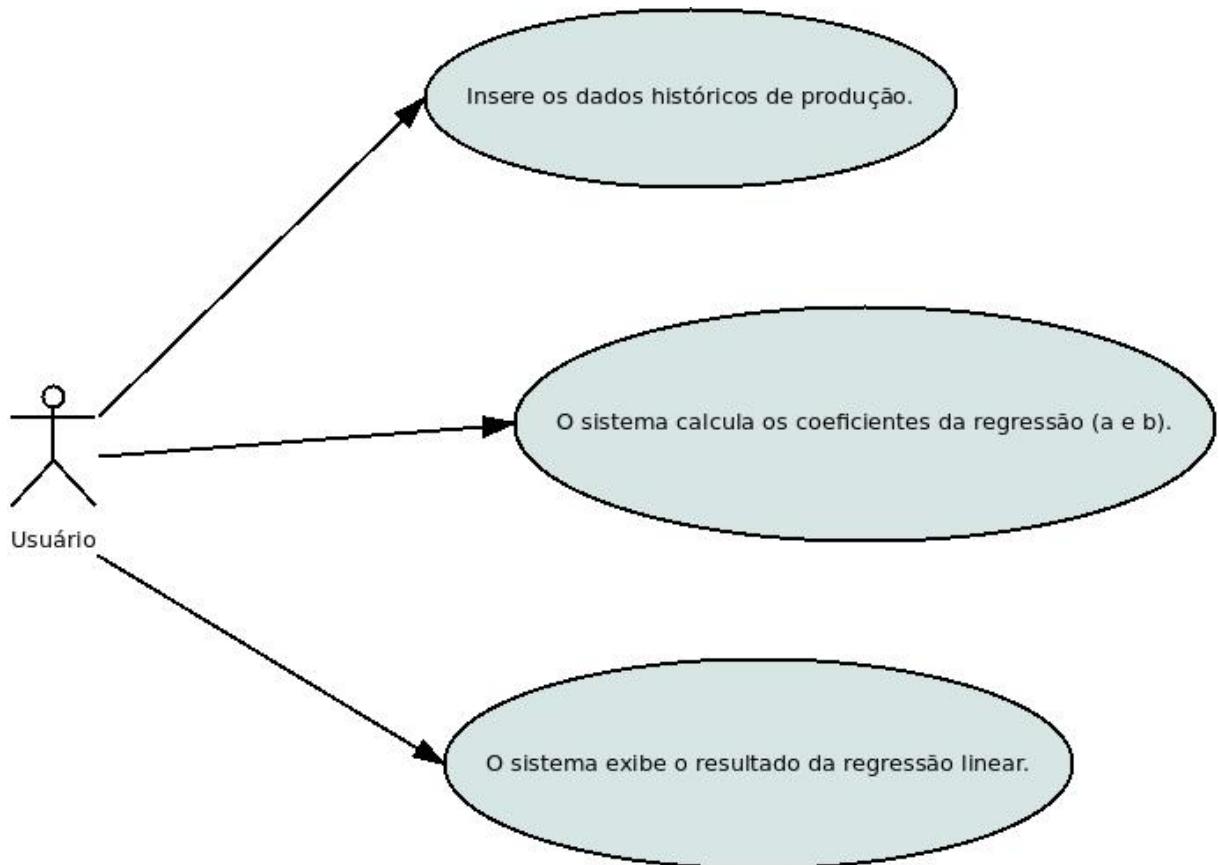


Figura 2.2: Diagrama de Regressão Linear

O diagrama de Modelagem de Reservatório de Gás da Figura 2.3 mostra o usuário modelando um reservatório considerando mecanismos de produção de gás.

O diagrama de Modelagem de Injeção de Fluido da Figura 2.4 mostra o usuário modelando um reservatório considerando a injeção de fluido em caso de recuperação avançada de reservatórios.

## 2.3 Requisitos

Apresenta-se nesta seção os requisitos funcionais e não funcionais.

### 2.3.1 Requisitos funcionais

Apresenta-se a seguir os requisitos funcionais.

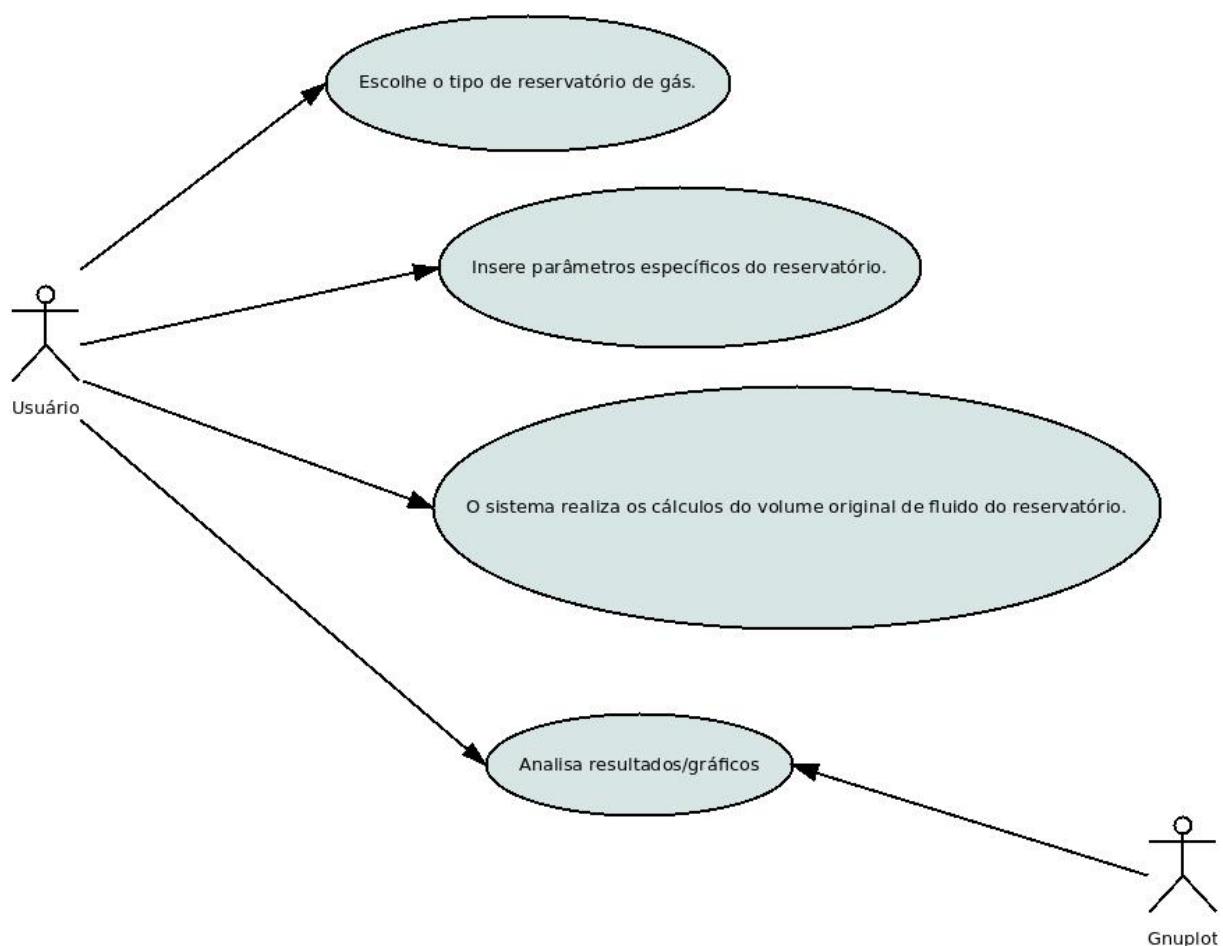


Figura 2.3: Diagrama de Modelagem de Reservatório de Gás

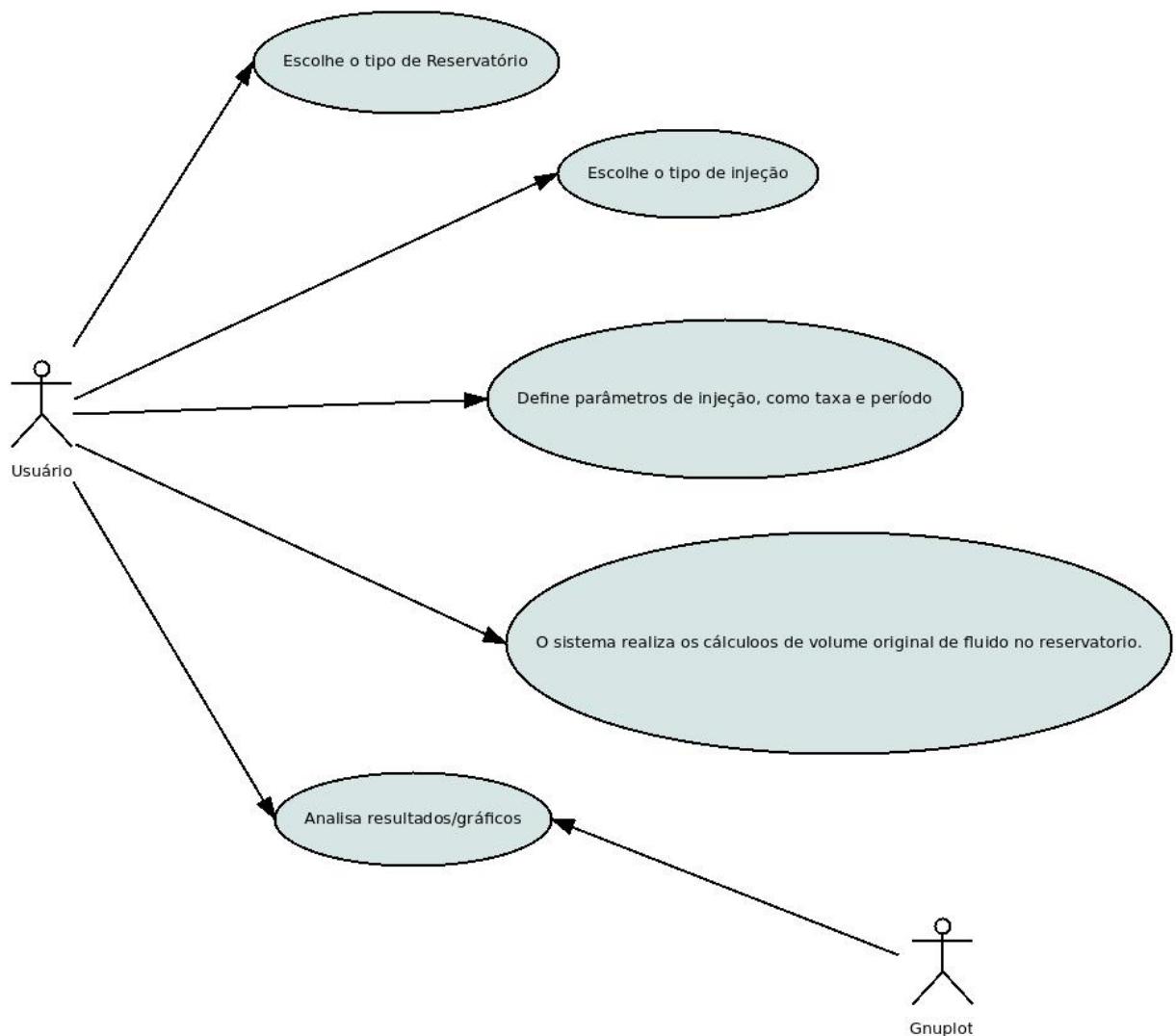


Figura 2.4: Diagrama de Modelagem de Injeção de Fluido

<b>RF-01</b>	O sistema deve conter uma base de dados para armazenar informações essenciais. O usuário deve ser capaz de acessar e manipular dados na base de dados.
<b>RF-02</b>	Deve permitir a escolha da equação apropriada para estimar reservas. O usuário deve ter a liberdade de selecionar a equação mais adequada para o cenário
<b>RF-03</b>	O sistema deve permitir o carregamento de arquivos criados pelo software. Os arquivos carregados devem conter dados relevantes para a simulação.
<b>RF-04</b>	O usuário deve ter a liberdade de configurar parâmetros essenciais para a simulação. Deve permitir a personalização de variáveis importantes para o cálculo de reservas.
<b>RF-05</b>	O sistema deve gerar gráficos representando os resultados da simulação. Os gráficos devem ser interativos e informativos.
<b>RF-06</b>	O usuário deve poder salvar os gráficos gerados como imagem. Os dados dos gráficos também devem ser exportáveis em formato de texto.

### 2.3.2 Requisitos não funcionais

<b>RNF-01</b>	Os cálculos devem ser realizados utilizando algoritmos eficientes e precisos, a EBM. O método de cálculo deve garantir resultados confiáveis e consistentes.
<b>RNF-02</b>	O programa deve ser desenvolvido de forma a ser executado em sistemas operacionais Windows, GNU/Linux e Mac. A interface do usuário deve ser responsiva em diferentes ambientes operacionais.

## 2.4 Casos de uso

A Tabela 2.1 mostra a descrição de um caso de uso.

Tabela 2.1: Caso de uso 1

Nome do caso de uso:	Simulação do Volume Original de Gás e Óleo
Resumo/descrição:	Realizar simulações para determinar os volumes originais de gás e óleo, salvar dados analisados do histórico de produção e gerar gráficos com Gnuplot.
Etapas:	<ul style="list-style-type: none"> <li>-O usuário inicia a simulação.</li> <li>-Durante a simulação, o usuário escolhe o mecanismo de produção.</li> <li>-O usuário seleciona o método de injeção desejado.</li> <li>-Os dados analisados são salvos para referência futura.</li> <li>-Um gráfico é gerado com o auxílio do programa externo Gnuplot.</li> </ul>
Cenários alternativos:	<ul style="list-style-type: none"> <li>-O usuário pode escolher entre diferentes mecanismos de produção, como produção de gás, produção de óleo, etc.</li> <li>-O usuário tem a opção de selecionar métodos de injeção, como injeção de água, injeção de gás, entre outros.</li> <li>-Caso o usuário deseje, pode salvar os relatórios analisados para uso posterior.</li> <li>-A geração do gráfico é realizada utilizando o programa externo Gnuplot para uma representação visual dos resultados da simulação.</li> </ul>

#### 2.4.1 Diagrama de caso de uso geral

O diagrama de caso de uso geral da Figura 2.5 mostra o usuário acessando as interações principais do sistema. É o ponto central, e a partir dele, o usuário pode escolher diferentes mecanismos de produção, selecionar métodos de injeção, salvar dados analisados do histórico de produção e analisar os gráficos. Cada uma dessas interações é um caso de uso separado, permitindo uma representação visual das funcionalidades principais do seu software.

#### 2.4.2 Diagrama de caso de uso específico

O caso de uso específico Escolha do Mecanismo de Produção descrito na Figura 2.5 e na Tabela 2.1 é detalhado na Figura 2.6. O usuário escolhe o mecanismo de produção desejado. O sistema oferece opções como produção de gás, produção de óleo, expansão da capa de gás, influxo de água, e produção de gás condensado. O usuário deve selecionar o mecanismo desejado para prosseguir com a simulação.

O caso de uso específico Selecionar Método de Injeção na Figura 2.5 e na Tabela 2.1 é detalhado na Figura 2.7. O usuário pode escolher o método de injeção desejado para otimizar a produção. As opções incluem injeção de água, injeção de gás, injeção de polímero, injeção de surfactante, injeção de vapor e Gas Lift. A escolha do método de injeção influenciará os resultados da simulação.

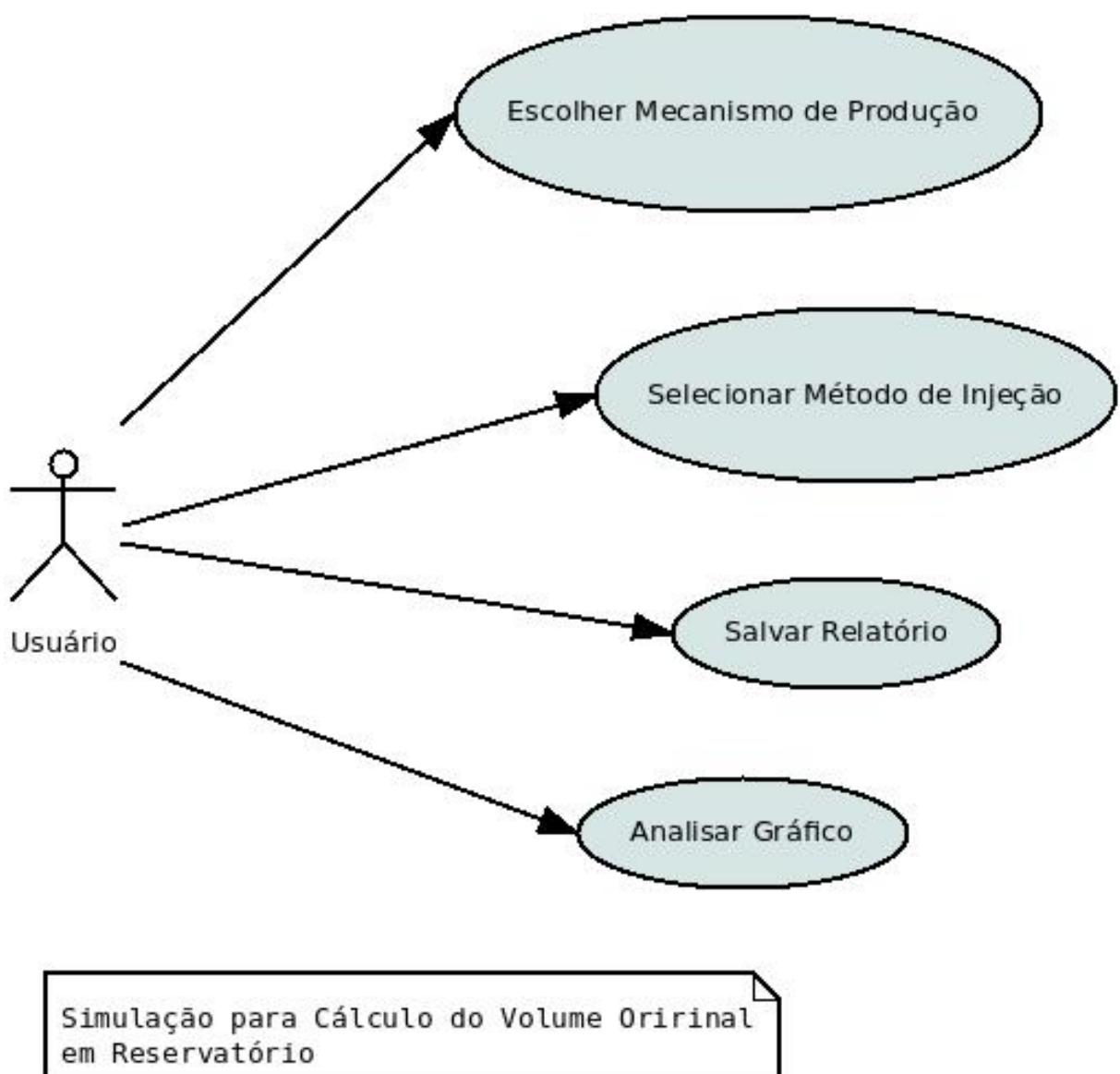


Figura 2.5: Diagrama de caso de uso – Caso de uso geral

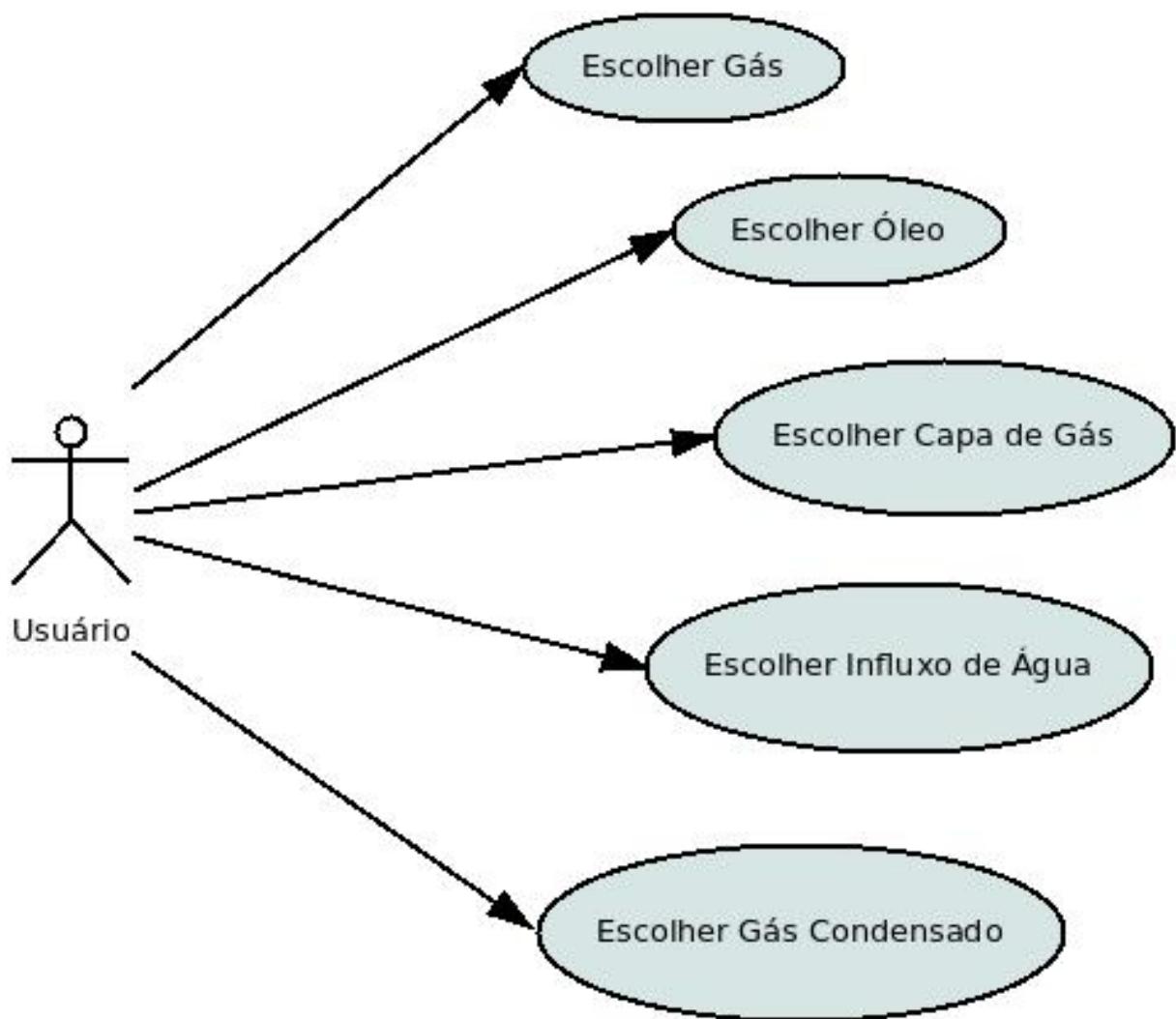


Figura 2.6: Diagrama de caso de uso específico – Escolha do Mecanismo de Produção

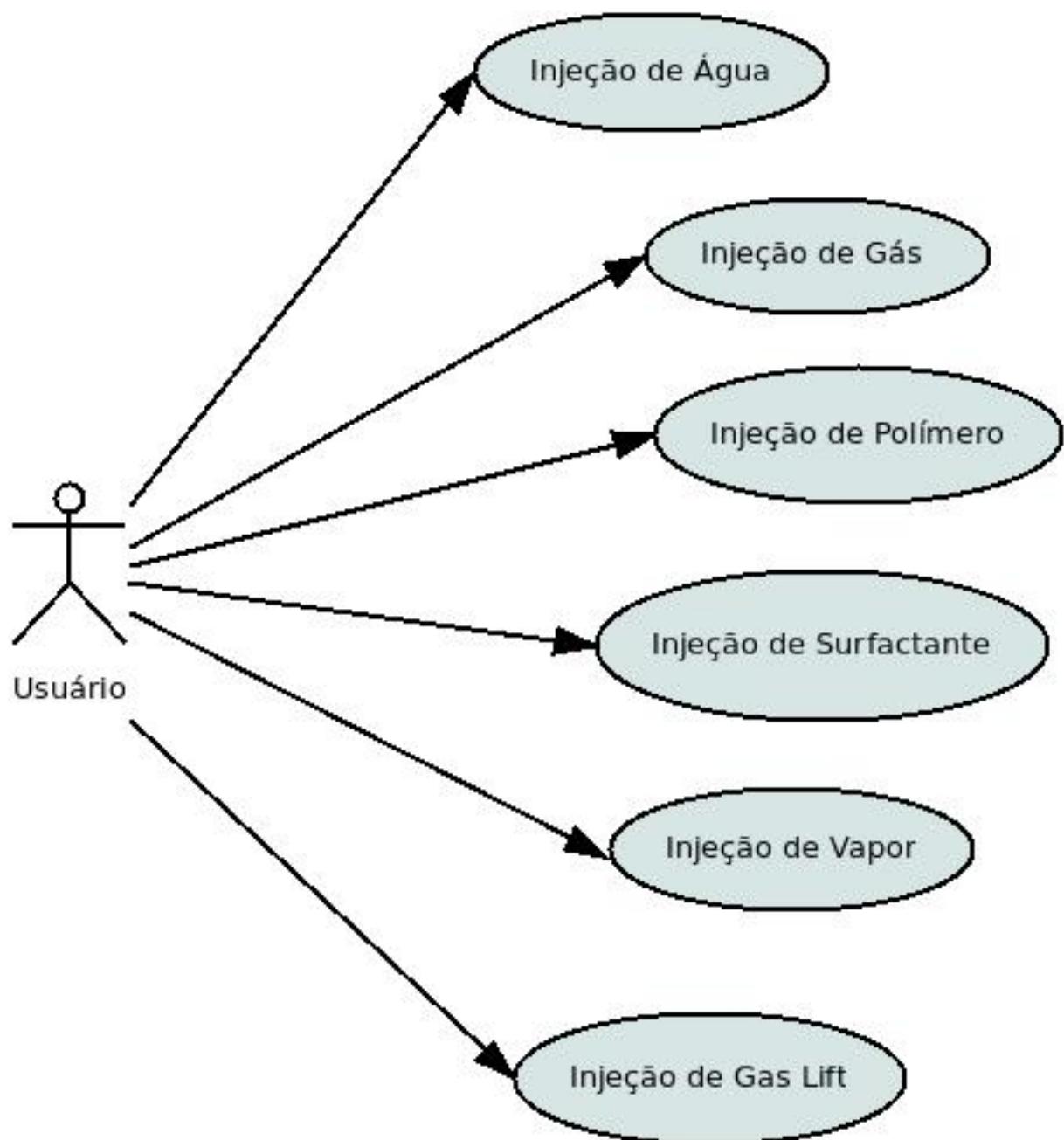


Figura 2.7: Diagrama de caso de uso específico – Selecionar Método de Injeção

O caso de uso específico Salvar Relatório na Figura 2.5 e na Tabela 2.1 é detalhado na Figura 2.8. O usuário pode escolher o local de salvamento, definir um nome para o arquivo e confirmar o salvamento. Essa funcionalidade é útil para revisão futura dos resultados ou compartilhamento com outras partes interessadas.

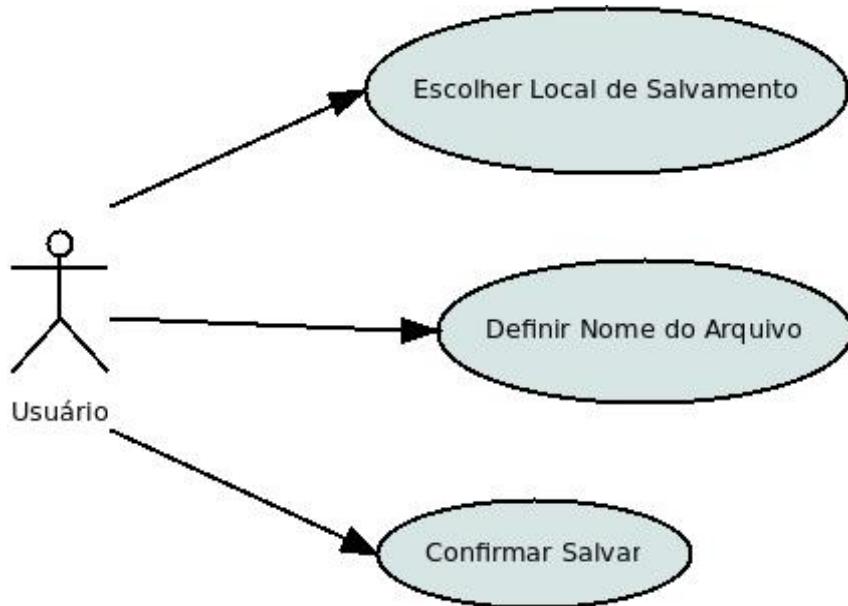


Figura 2.8: Diagrama de caso de uso específico – Salvar Relatório

O caso de uso específico Gerar Gráfico na Figura 2.5 e na Tabela 2.1 é detalhado na Figura 2.9. O usuário pode gerar um gráfico visualizando os resultados da simulação usando o programa externo Gnuplot. Neste caso de uso, o usuário escolhe o tipo de gráfico desejado, personaliza as configurações conforme necessário, visualiza uma prévia do gráfico e, finalmente, confirma a geração do gráfico. Isso proporciona uma representação visual dos resultados da simulação.

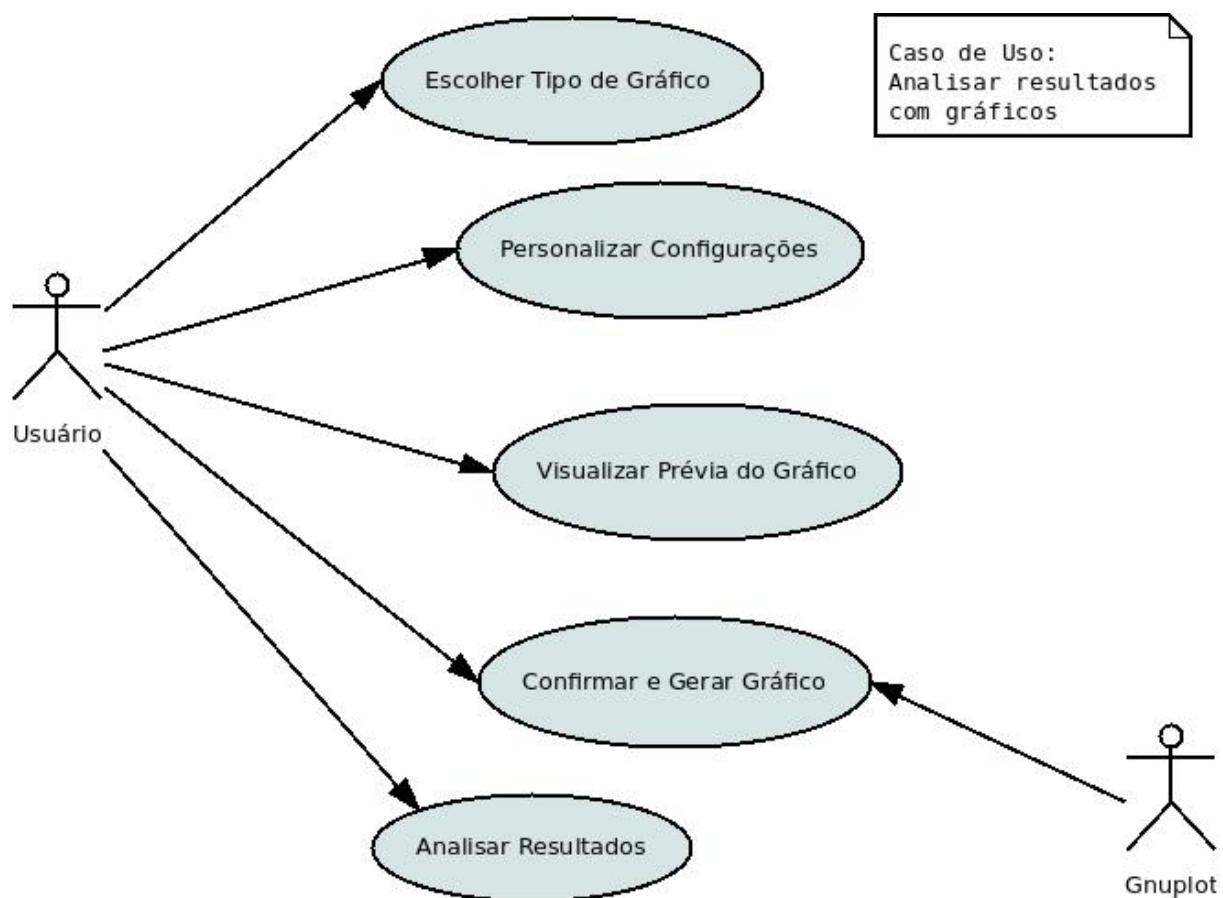


Figura 2.9: Diagrama de caso de uso específico – Gerar Gráfico

# Capítulo 3

## Elaboração

Nesse estágio, o foco se volta para o aprimoramento dos conceitos relacionados ao sistema a ser desenvolvido, envolvendo a análise aprofundada do domínio do problema e a identificação de pacotes que comporão a arquitetura do sistema.

A elaboração é marcada por uma análise minuciosa dos requisitos, visando ajustar as diretrizes iniciais de modo a conceber um sistema que não apenas atenda às necessidades do usuário, mas também proporcione flexibilidade para reuso e futuras extensões. Durante esse processo, são identificados e eliminados requisitos considerados "impossíveis" ou inviáveis, e a concepção do sistema é ajustada para otimizar sua utilidade e adaptabilidade.

A flexibilidade é um ponto-chave, levando em consideração aspectos como custos e prazos. A equipe busca criar uma estrutura que possa se adaptar a mudanças, seja na evolução das necessidades do usuário ou em resposta a novos desafios. Isso contribui para a robustez do sistema ao longo do tempo, garantindo que ele possa ser expandido e mantido de maneira eficiente.

Além disso, a elaboração envolve a identificação de pacotes, que são conjuntos lógicos de funcionalidades ou módulos que compõem a arquitetura do sistema. Essa subdivisão ajuda a organizar e estruturar o desenvolvimento, facilitando a compreensão e manutenção do sistema.

### 3.1 Análise de domínio

Após estudo dos requisitos/especificações do sistema, algumas entrevistas, estudos na biblioteca e disciplinas do curso foi possível identificar nosso domínio de trabalho:

- Definição e caracterização do domínio a ser investigado
  - O domínio de trabalho abrange o setor de engenharia de reservatórios, especificamente na simulação e análise de produção de campos de petróleo e gás. O software visa modelar diferentes mecanismos de produção, métodos de injeção e cenários de reservatórios para estimar volumes originais de gás e óleo.

- Descrição das áreas/disciplinas relacionadas.

– Área: Engenharia de Petróleo

A área relacionada ao desenvolvimento do programa é a Engenharia de petróleo. É a área da engenharia que trata de todos os ramos relacionados à produção de hidrocarbonetos, que podem ser óleo ou gás natural. As atividades são divididas geralmente em duas grandes áreas: upstream, refere-se às atividades de exploração e produção e downstream, que refere-se às atividades de refino e distribuição.

– Sub-área: Engenharia de Reservatório

A Engenharia de Reservatório constitui uma sub-área de extrema importância na engenharia de petróleo, e está associada a descrição do comportamento do óleo, do gás e da água no interior dos meios porosos que compõem as acumulações petrolíferas na sub-superfície, essencial para o desenvolvimento de campos de petróleo. Os engenheiros, geólogos e geofísicos de petróleo, assim como outros profissionais que atuam na área de engenharia de reservatórios, utilizam informações sobre as propriedades e características das rochas e dos fluidos contido nas formações portadoras de petróleo, bem como seu comportamento passado (no caso de parte do fluido já ter sido produzida), para inferir o comportamento futuro desses reservatórios.

– Área: Modelagem Numérica Computacional

A Modelagem Numérica Computacional é a área que trata do desenvolvimento de soluções para problemas científicos de Engenharia, analisando os fenômenos, desenvolvendo modelos matemáticos para sua descrição, e elaborando algorítimos e códigos computacionais para obtenção daquelas soluções.

- Descrição de questões associadas a espaço/tempo (espaço físico, local, instalação).
  - Espaço Físico: Consideração das características físicas do reservatório, como geometria, dimensões e propriedades dos fluidos, influenciando diretamente nas simulações.
  - Local/Instalação: Adaptação do software para lidar com diferentes locais de reservatórios, levando em conta variações nas condições geográficas e climáticas.
- Fotografias de sistemas equivalentes/parecidos.
  - Software de Simulação de Reservatórios: Visualização de softwares existentes no mercado com funcionalidades semelhantes, destacando interfaces, gráficos e formas de apresentação de resultados.

- Equipamentos de Perfuração e Produção: Imagens de instalações e equipamentos utilizados na indústria de petróleo, fornecendo insights sobre a infraestrutura física associada.

- Outras Considerações

- Regulamentações e Normas: Exploração das regulamentações e normas que impactam as atividades de exploração e produção de petróleo e gás.
- Tendências Tecnológicas: Investigação de tendências recentes em simulação de reservatórios, como o uso de inteligência artificial ou métodos avançados de modelagem.

## 3.2 Formulação teórica

O software calcula as reservas dos reservatórios de óleo e gás, utilizando diferentes métodos de acordo com o tipo de mecanismo de produção. Para isto será utilizada a equação de balanço de materiais, que se baseia no balanço de massas dos fluidos existentes no interior dos poros da rocha-reservatório.

A massa de fluidos existente no reservatório em um determinado instante é a diferença entre a massa original e a massa produzida. Como os volumes dos fluidos produzidos são geralmente medidos em uma determinada condição-padrão de pressão e de temperatura, a equação de balanço de materiais é comumente escrita de tal maneira que, em um instante qualquer, o volume de fluidos existente no reservatório seja a diferença entre o volume inicialmente existente e o produzido, ambos medidos em condição-padrão.

**Para reservatórios de gás o programa utiliza a equação de balanço de materiais:**

$$\frac{p}{Z} = \frac{1}{V_i - W_e} \left( \frac{p_i V_i}{Z_i} - \frac{T p_0}{T_0} G_p \right) \quad (3.1)$$

Onde:

- $p$  [kgf/cm<sup>2</sup>] = pressão
- $Z$  [adimensional] = coeficiente ou fator de compressibilidade
- $V$  [m<sup>3</sup>] = volume ocupado de gás
- $W_e$  [m<sup>3</sup>] = influxo de água acumulado
- $G_p$  [m<sup>3</sup> std] = gás produzido
- Índice  $i$  = condições iniciais no reservatório
- Índice 0 = condições padrão

A equação de balanço de materiais utilizada para reservatório de óleo pode ser escrita como:

$$N = \frac{N_p [B_o + (R_p - R_s) B_g] - W_e}{B_o - B_{oi} + (R_{si} - R_s) B_g + m B_{oi} \left( \frac{B_g}{B_{gi}} - 1 \right) + (1 + m) B_{oi} \left( \frac{c_w S_{wi} + c_f}{1 - S_{wi}} \right) \Delta p} \quad (3.2)$$

Onde:

- $N$  [ $\text{m}^3\text{std}$ ] = volume original de óleo
- $N_p$  [ $\text{m}^3$  std] = volume de óleo produzido
- $B_o$  [ $\text{m}^3/\text{m}^3$  std] = fator volume-formação do óleo
- $R_p$  [ $\text{m}^3 \text{ std}/\text{m}^3 \text{ std}$ ] = razão gás/óleo acumulada
- $R_s$  [ $\text{m}^3/\text{m}^3$  std] = razão de solubilidade
- $B_g$  [ $\text{m}^3/\text{m}^3$  std] = fator volume-formação do gás
- $W_e$  [ $\text{m}^3$ ] = influxo de água acumulado
- $m$  [adimensional] = razão entre o volume original de gás e o volume original de óleo em condições de reservatório
- $c_w$  [ $(\text{k}\text{g}\text{f}/\text{cm}^2)^{-1}$ ] = compressibilidade da água
- $S_{wi}$  [adimensional] = saturação de água conata
- $c_f$  [ $(\text{k}\text{g}\text{f}/\text{cm}^2)^{-1}$ ] = compressibilidade da formação ou de poros
- $\Delta p = p_i - p$  [ $\text{k}\text{g}\text{f}/\text{cm}^2$ ] = variação de pressão
- Índice  $i$  = condições iniciais no reservatório

Podemos simplificar esta equação, para facilitar a regressão linear posteriormente:

$$F = N(E_o + mE_g + E_{f,w}) + W_e, \quad (3.3)$$

onde:

$$F = N_p[B_o + (R_p - R_s)B_g] + W_pB_w - (V_{inj}B_{vinj} + S_{inj}B_{sinj} + P_{inj}B_{pinj} + G_{inj}B_{ginj} + W_{inj}B_{winj} + GL_{inj}B_{gl}) \quad (3.4)$$

representa o volume de fluidos injetados e produzidos;

$$E_o = B_o + (R_{si} - R_s)B_g - B_{oi}, \quad (3.5)$$

representa a expansão hidrocarbonetos na zona de óleo::

$$E_g = B_{oi} \frac{(B_g - B_{gi})}{B_{gi}}, \quad (3.6)$$

representa a expansão dos hidrocarbonetos na capa de gás::

$$E_{f,w} = (1 + m)B_{oi}((c_w S_{wi} + c_f)/(1 - S_{wi}))(p_i - p), \quad (3.7)$$

representa os efeitos da expansão da água conata e redução do volume poroso;

$$m = GB_{gi}/NB_{oi}. \quad (3.8)$$

A partir dessa equação geral o programa determina o volume de óleo original do reservatório para cada tipo de mecanismo associado analisando suas características e considerações.

### 3.3 Descrição dos mecanismos de produção para reservatório de gás

Os mecanismos de produção para reservatórios de gás são:

- Depleção: quando o reservatório é colocado para produzir, há um diferencial de pressão entre o fundo do poço e o interior do reservatório, o que causa o início da produção. À medida que o gás vai sendo produzido, este se expande. Quando o reservatório produz somente por depleção, é chamado de reservatório volumétrico.
- Influxo de água: quando há um aquífero contíguo ao reservatório de gás. À medida que ocorre a retirada de gás através da produção, há um influxo de água para o interior do reservatório, que empurra o gás ainda existente no reservatório e dá continuidade à produção.

Os reservatórios de gás a serem considerados serão:

#### 3.3.1 Reservatórios de gás seco

Reservatórios de gás seco são aqueles em que o fluido se encontra no estado gasoso tanto em condições de reservatório, quanto de superfície.

Para um reservatório volumétrico de gás seco, que produz somente por depleção, ou seja, por expansão da massa de gás existente no meio poroso, não

havendo, portanto, influxo de água proveniente de aquífero, e desconsidera-se as variações de volume poroso e água conata (por serem desprezíveis comparadas à expansão do gás), a equação (3.1) reduz-se a:

$$p/Z = p_i/Z_i - Tp_0G_p/(V_iT_0), \quad (3.9)$$

ou ainda:

$$p/Z = p_i/Z_i - p_iG_p/(Z_iG). \quad (3.10)$$

Para um reservatório de gás seco anormalmente pressurizado, que apresenta grandes gradientes de pressão, a rocha é inconsolidada e altamente compressível, logo não podemos desprezar as variações do volume poroso e de água conata. Também produz por depleção. Portanto, a EBM fica:

$$\frac{p}{Z} \left[ 1 - \frac{(c_wS_{wi} + c_f)\Delta p}{1 - S_{wi}} \right] = \frac{p_i}{Z_i} - \frac{p_i}{Z_iG}G_p. \quad (3.11)$$

Definindo a compressibilidade efetiva do sistema água formação ( $c_{ewf}$ ):

$$c_{ewf} = \frac{c_wS_{wi} + c_f}{1 - S_{wi}}, \quad (3.12)$$

simplificamos a equação (3.11) em:

$$\frac{p}{Z}(1 - c_{ewf}\Delta p) = \frac{p_i}{Z_i} - \frac{p_i}{Z_iG}G_p. \quad (3.13)$$

Reservatórios de gás seco sob influxo de água, que são aqueles em que à medida em que ocorre a retirada de massa através da produção dos poços, há influxo de água para o seu interior proveniente de um aquífero contíguo à zona de gás (este mecanismo também ocorre nos reservatórios de óleo). Para reservatórios de gás sob influxo de água considera-se que:

$$V = V_i - W_e. \quad (3.14)$$

Para determinação do influxo de água são utilizados dois modelos. No programa o usuário irá definir o modelo usado: Pot Aquifer ou Schilthuis.

- **Modelo Pot Aquifer**

Para aquíferos de tamanho próximo ao do reservatório. Em reservatórios com mecanismo de influxo de água, a queda de pressão devida à produção de fluidos causa uma expansão da água do aquífero contíguo à zona de óleo. Essa expansão provoca um fluxo de água para o interior do reservatório (Dake, 1978). De acordo com a definição de compressibilidade, aplicada ao aquífero, pode-se dizer que:

$$W_e = (c_w + c_f)W_i\Delta p, \quad (3.15)$$

onde  $W_i$  [m<sup>3</sup>] é o volume inicial de água no aquífero,  $\Delta p$  [kg/cm<sup>2</sup>] é a queda de pressão no aquífero, e a compressibilidade total do aquífero é a soma das compressibilidades da água ( $c_w$ ) e dos poros ( $c_f$ ), uma vez que o meio poroso se encontra 100% saturado com água. Se o aquífero possui uma geometria radial, o influxo de água pode ser calculado por (Dake, 1978):

$$W_e = (c_w + c_f)\pi(r_e^2 - r_o^2)h\phi\Delta p, \quad (3.16)$$

onde  $r_e$  [m] é o raio do aquífero,  $r_o$  [m] é o raio do reservatório e  $h$  [m] a espessura da formação. Nesse modelo, admite-se que a queda de pressão no aquífero em um determinado instante seja igual à queda de pressão no contato reservatório-aquífero.

- **Modelo Schilthuis**

Se, por outro lado, o aquífero é bem maior que o reservatório, essas equações se tornam inadequadas para descrever o influxo de água, porque a queda de pressão no limite do reservatório (contato água-óleo) não se transmite instantaneamente através do aquífero. Para aquíferos maiores há um intervalo de tempo entre a queda de pressão no reservatório e a resposta do aquífero. Um dos modelos utilizados para o cálculo de influxo é o modelo de Schilthuis (1936),

$$dW_e/dt = J(p_i - p), \quad (3.17)$$

onde  $J$  [m<sup>3</sup>/(d\*kgf/cm<sup>2</sup>)] é denominada constante de influxo de água, também conhecida como índice de produtividade do aquífero,  $p_i$  [kgf/cm<sup>2</sup>] é a pressão inicial do sistema e  $p$  [kgf/cm<sup>2</sup>] é a pressão no contato óleo/água.

### 3.3.2 Reservatórios de gás condensado

Reservatórios de gás condensado são aqueles que em condições de reservatório o fluido se encontra no estado gasoso, mas na superfície produz uma certa quantidade de líquido.

Reservatórios de gás condensado não-retrógrado, que são aqueles em que os fluidos se encontram no estado gasoso dentro do reservatório, mas em condições de superfície produzem volumes equivalentes de água e gás.

Para reservatórios de gás condensado não-retrógrado o programa faz alguns cálculos e aplica-se a equação de balanço de materiais (3.9) acima citada. A

razão disso, é que para esse tipo de reservatório precisa-se achar um novo valor para  $G_p$  [ $\text{m}^3 \text{ std}$ ] e para isso considera que:

Precisa-se achar o volume de gás equivalente para cada  $1 \text{ m}^3$  de condensado produzido:

$$(GE)_c = 23686d_c/M_c, \quad (3.18)$$

onde  $(GE)_c$  [ $\text{m}^3 \text{ std}/\text{m}^3 \text{ std}$ ] é o volume de gás equivalente a  $1 \text{ m}^3$  produzido,  $d_c$  [adimensional] é a densidade do condensado e  $M_c$  [adimensional] é a massa molecular do condensado. Sendo que:

$$M_c = 44.29d_c/(1.03 - d_c). \quad (3.19)$$

Precisa-se saber também o volume de vapor de água produzido para cada  $1 \text{ m}^3$  de condensado produzido e esse valor será

$$(GE)_w = 1316 \text{ [m}^3\text{std/m}^3\text{std]}. \quad (3.20)$$

Chegando-se então no seguinte valor para  $(G_p)_t$

$$(G_p)_t = G_p + (GE)_{ct} + (GE)_{wt}, \quad (3.21)$$

onde  $(G_p)_t$  [ $\text{m}^3 \text{ std}$ ] é o novo valor para  $G_p$  a ser usado na equação de balanço de materiais (3.9),  $G_p$  [ $\text{m}^3 \text{ std}$ ] é o volume de gás produzido que continua no estado gasoso na superfície,  $(GE)_{ct}$  [ $\text{m}^3 \text{ std}$ ] é o volume acumulado de condensado produzido e  $(GE)_{wt}$  é o volume acumulado de água produzida.

Reservatórios de gás condensado retrógrado, que são aqueles em que há formação de líquido ainda dentro do reservatório, à medida que ocorre a depleção. Emprega-se o fator de compressibilidade duas fases  $Z_{2f}$  [adimensional] no lugar do fator de compressibilidade  $Z$ . O valor de  $Z_{2f}$  pode ser obtido por análises PVT ou por correlações empíricas.

### 3.4 Descrição dos mecanismos de produção para reservatório de óleo

Os reservatórios de óleo a serem considerados são:

- Reservatórios com gás em solução, cujo mecanismo de produção se baseia na expansão do gás inicialmente dissolvido no óleo e que foi liberado devido à redução de pressão, consequência da produção de fluidos. Para este reservatório, considerando-se a compressibilidade desprezível obtemos a equação de balanço de materiais:

$$F = NE_o; \quad (3.22)$$

- Reservatórios com capa de gás, cujo mecanismo de produção se baseia na expansão da capa de gás devido à diminuição de pressão. Para um reservatório com mecanismo de capa de gás, temos:

$$F = N(E_o + mE_g); \quad (3.23)$$

- Reservatórios com influxo de água, onde a queda de pressão devida à produção de fluidos causa uma expansão da água do aquífero contíguo à zona de óleo. Para um reservatório com mecanismo de influxo de água:

$$F = NE_o + W_e. \quad (3.24)$$

Estas equações obtidas são apenas para a linearização da EBM, para facilitar a regressão linear. Seus termos já foram explicados no início deste capítulo.

### 3.5 Referências Bibliográficas

- ROSA, ADALBERTO J.; CARVALHO, RENATO S.; XAVIER, JOSÉ AUGUSTO D. Engenharia de Reservatório de Petróleo. Rio de Janeiro: Editora Interciência, 2006.
- AHMED, TAREK. Reservoir Engineering Handbook. Second Edition. Texas: Gulf Professional Publishing, 2001.
- DAKE, L. P. Fundamentals of reservoir engineering. Rio de Janeiro: Elsevier, 1978.

### 3.6 Identificação de pacotes – assuntos

- Pacote Reservatório:

Contém todos os conceitos associados à reservatório incluindo modelos de equação do balanço de materiais, modelos para determinação do influxo de água, e os tipos de mecanismos associados aos reservatórios como mecanismo de influxo de água, capa de gás, gás em solução, etc.

- Pacote Métodos Numéricos:

Esse pacote inclui pacotes para realização de operações numéricas como o pacote regressão linear, que será utilizado para determinação de funções lineares a partir de dados de

produção do reservatório para os diferentes mecanismos de produção, e o pacote integral, que será utilizado nos modelos de influxo de água.

- Pacote Gráfico:

Responsável por todas as funcionalidades gráficas, incluindo plotagem de gráficos, exportação de dados e configurações visuais.

### 3.7 Diagrama de pacotes – assuntos

Este diagrama de pacotes organiza as principais funcionalidades do software, facilitando a compreensão das diferentes partes e suas interações. Veja Figura 3.1.

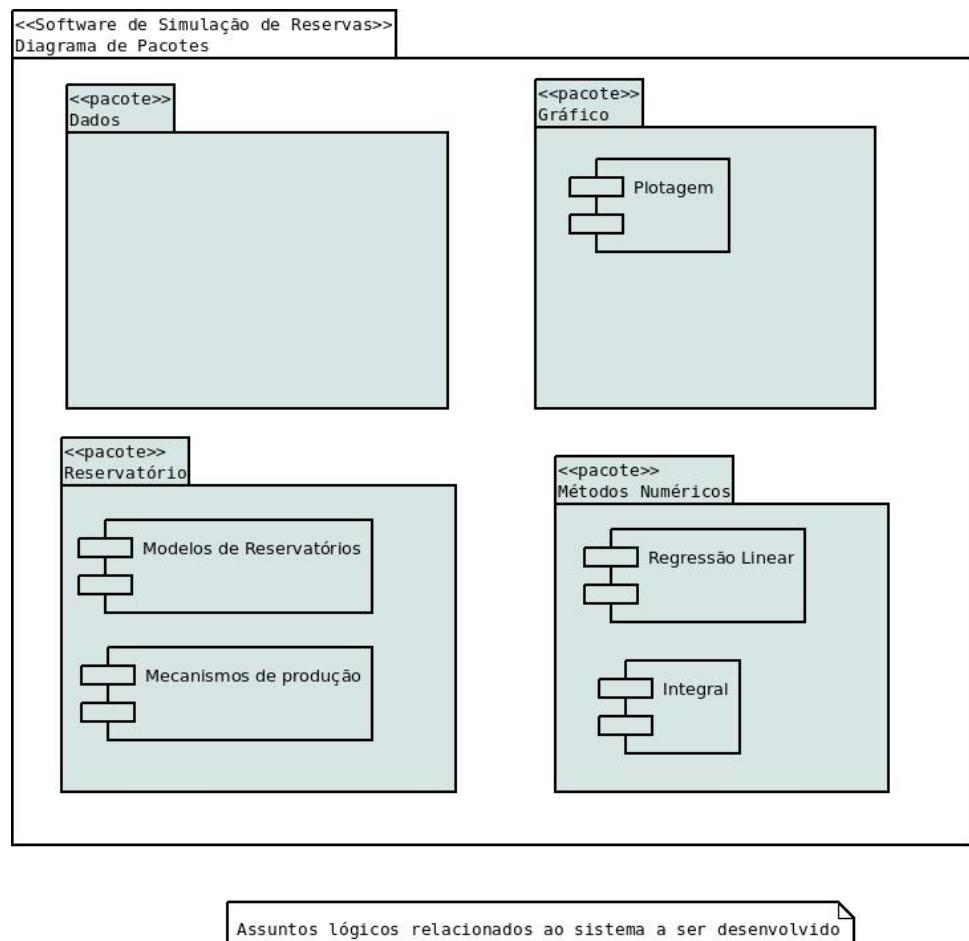


Figura 3.1: Diagrama de Pacotes

# Capítulo 4

## AOO – Análise Orientada a Objeto

A terceira etapa do desenvolvimento de um projeto de engenharia, no nosso caso um software aplicado a engenharia de petróleo, é a AOO – Análise Orientada a Objeto. A AOO utiliza algumas regras para identificar os objetos de interesse, as relações entre os pacotes, as classes, os atributos, os métodos, as heranças, as associações, as agregações, as composições e as dependências.

O modelo de análise deve ser conciso, simplificado e deve mostrar o que deve ser feito, não se preocupando como isso será realizado.

O resultado da análise é um conjunto de diagramas que identificam os objetos e seus relacionamentos.

### 4.1 Diagramas de classes

O diagrama de classes é uma representação visual das classes que compõem um sistema, juntamente com seus atributos, métodos e relacionamentos. No contexto do software que estamos discutindo, que envolve simulações e análises de reservatórios de gás e óleo, o diagrama de classes pode ser estruturado da seguinte forma:

O diagrama de classes é apresentado na Figura 4.1.

#### 4.1.1 Dicionário de classes

### 4.2 Diagramas de classes

O diagrama de classes é apresentado na Figura ???. Uma descrição das classes, seus atributos e métodos é apresentada a seguir.

#### 4.2.1 Dicionário de classes

- Classe CSimuladior: representa a classe que geral do simulador.
  - método CSimulador: representa o simulador.

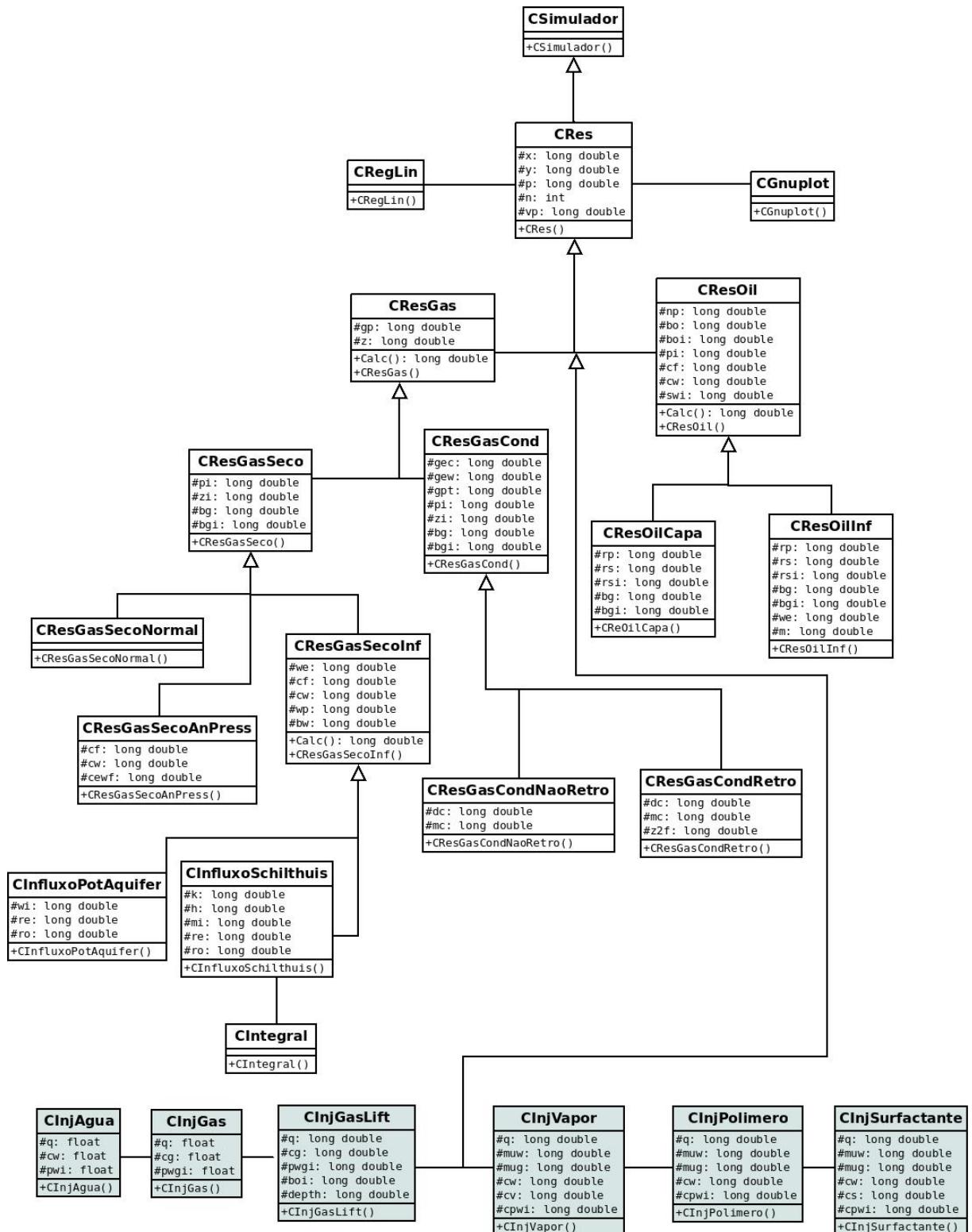


Figura 4.1: Diagrama de classes

- Classe CRegLin: representa a classe que fará a regressão linear dos pontos gerados pelas classes herdeiras da classe CRes.
  - atributo a: coeficiente linear que vai caracterizar uma grandeza física de um dos mecanismos de produção.
  - atributo b: coeficiente angular que vai caracterizar uma grandeza física de um dos mecanismos de produção.
  - atributo sx: representa o somatório das variáveis x.
  - atributo sy: representa o somatório das variáveis y.
  - atributo sxx: representa o somatório do quadrado das variáveis x.
  - atributo syy: representa o somatório do quadrado das variáveis y.
  - método Reg: representa os cálculos da regressão.
- Classe CRes: representa os conceitos básicos da engenharia de reservatório.
  - atributo x: representa a variável das abscissas para cada mecanismo de produção.
  - atributo y: representa a variável das ordenadas para cada mecanismo de produção.
  - atributo n: determina o número de variáveis dos atributos x e y.
  - atributo p [kgf/cm<sup>2</sup>]: representa a pressão no reservatório para cada histórico de produção.
  - método Calc:
  - método CRes: representa o construtor da classe.
- Classe CResGas: representa os conceitos fundamentais dos mecanismos de produção de gás.
  - atributo gp [m<sup>3</sup> std]: este armazena o volume de gás produzido.
  - atributo z [adimensional]: representa o fator de compressibilidade do gás.
  - método Calc: calcula as variáveis x e y da classe CRes.
  - método CResGas: representa o construtor da classe.
- Classe CResOil: representa os conceitos fundamentais dos mecanismos de produção de óleo.
  - atributo np [m<sup>3</sup> std]: representa o volume de óleo produzido para cada histórico de produção.

- atributo bo [ $m^3/m^3$  std]: representa o fator volume de formação do óleo para cada histórico de produção.
  - atributo boi [ $m^3/m^3$  std]: representa o fator volume de formação inicial do óleo para o primeiro histórico de produção.
  - atributo pi [ $kgf/cm^2$ ]: representa a pressão inicial no primeiro histórico de produção.
  - atributo cf [ $(kgf/cm^2)^{-1}$ ]: representa a compressibilidade da rocha reservatório.
  - atributo cw [ $(kgf/cm^2)^{-1}$ ]: representa a compressibilidade do fluido do reservatório.
  - atributo swi: representa a saturação de água inicial do reservatório.
  - método Calc: calcula as variáveis x e y da classe CRes.
  - método CResOil: representa o construtor da classe.
- Classe CResGasSecoPres: representa os conceitos fundamentais do mecanismo de produção de gás anormalmente pressurizado.
    - atributo cw [ $(kgf/cm^2)^{-1}$ ]: representa a compressibilidade da água.
    - atributo cf [ $(kgf/cm^2)^{-1}$ ]: representa compressibilidade da formação.
    - atributo swi: representa a saturação de água inicial para cada histórico de produção.
    - método CResGasPres: representa o construtor da classe.
  - Classe CResGasSecoInf: representa os conceitos fundamentais do mecanismo de gás com influxo de água.
    - atributo we [ $m^3$ ]: respesta o influxo acumulado do aquífero.
    - atributo bg [ $m^3/m^3$  std]: representa o fator volume de formação para cada histórico de produção.
    - atributo bgi [ $m^3/m^3$  std]: representa o fator volume de formação inicial para cada histórico de produção.
    - método CResGasSecoInf: representa o construtor da classe.
  - Classe CResGasSecoInfPotAquiifer: representa os conceitos do mecanismo de gás com influxo de água, utilizando o método Pot Aquifer.
    - atributo we [ $m^3$ ]: respesta o influxo acumulado do aquífero.

- atributo bg [ $m^3/m^3$  std]: representa o fator volume de formação para cada histórico de produção.
  - atributo bgi [ $m^3/m^3$  std]: representa o fator volume de formação inicial para cada histórico de produção.
  - atributo deltap [ $kgf/cm^2$ ]: representa o diferencial de pressão.
  - atributo re [m]: representa o raio do aquífero.
  - atributo ro [m]: representa o raio do reservatório
  - atributo we: representa o influxo acumulado do aquífero.
  - atributo fi: representa a porosidade do aquífero.
  - atributo h [m]: representa a espessura do aquífero.
  - atributo cw [ $(kgf/cm^2)^{-1}$ ]: representa a compressibilidade da água.
  - atributo cf [ $(kgf/cm^2)^{-1}$ ]: representa a compressibilidade da formação.
  - método CResGasSecoInfPotAQUIFER: representa o construtor da classe.
- Classe CResGasSecoInfSchilthuis: representa os conceitos do mecanismo de gás com influxo de água, utilizando o método de Schilthuis.
    - atributo we [ $m^3$ ]: representa o influxo acumulado do aquífero.
    - atributo bg [ $m^3/m^3$  std]: representa o fator volume de formação para cada histórico de produção.
    - atributo bgi [ $m^3/m^3$  std]: representa o fator volume de formação inicial para cada histórico de produção.
    - atributo deltap [ $kgf/cm^2$ ]: representa o diferencial de pressão.
    - atributo j [ $m^3/(d*kgf/cm^2)$ ]: representa a constante de influxo.
    - atributos t1 e t2 [d]: representam os limites fixos de integração.
    - método CResGasSecoInfSchilthuis: representa o construtor da classe.
  - Classe CResGasCondNaoRetro: representa os conceitos do mecanismo de gás condensado não-retrógrado.
    - atributo gpp [ $m^3$  std]: representa o volume de gás produzido parcial.
    - atributo gec [ $m^3$  std]: representa o volume de gás condensado equivalente.
    - atributo gew [ $m^3$  std]: representa o volume de vapor d'água equivalente.
    - atributo gpt [ $m^3$  std]: representa o volume total de gás produzido.
    - atributo vw [ $m^3$ ]: representa o volume de água produzida.
    - atributo vc [ $m^3$ ]: representa o volume de gás condensado produzido.

- atributo dc: representa a densidade do gás.
  - atributo mc: representa a massa de condensado.
  - atributo bgi [ $m^3/m^3$  std]: representa o fator volume de formação inicial para cada histórico de produção.
  - método CResGasCondNaoRetro: representa o construtor da classe.
- Classe CResGasCondRetro: representa os conceitos do mecanismo de gás condensado retrógrado.
    - atributo bgi [ $m^3/m^3$  std]: representa o fator volume de formação inicial para cada histórico de produção.
    - atributo sl: representa a saturação de líquido no reservatório.
    - atributo n1: número de mols de líquido.
    - atributo n2: número de mols de líquido.
    - atributo temp [K]: temperatura do gás
    - atributo dl: representa a densidade do líquido.
    - atributo ml: representa a massa do líquido.
    - atributo z2f: representa o fator de compressibilidade de duas fases.
    - método CResGasCondRetro: representa o construtor da classe.
- Classe COilCapa: representa os conceitos do reservatório que tem como mecanismo de produção a expansão da capa de gás.
    - atributo bg: representa o fator volume de formação do gás para cada histórico de produção.
    - atributo bgi: representa o fator volume de formação inicial do gás para o primeiro histórico de produção.
    - atributo rp: representa a razão gás/óleo acumulada em condições de temperatura e pressão do reservatório.
    - atributo rs: representa razão de solubilidade do gás no óleo.
    - atributo rsi: representa razão de solubilidade inicial do gás no óleo.
    - método COilCapa: representa o construtor da classe.
- Classe COilInf: representa os conceitos do reservatório que tem como mecanismo de produção o influxo de água do aquífero para zona de óleo.
    - atributo bg: representa o fator volume de formação do gás para cada histórico de produção.

- atributo bgi: representa o fator volume de formação inicial do gás para o primeiro histórico de produção.
  - atributo rp: representa a razão gás/óleo acumulada em condições de temperatura e pressão do reservatório.
  - atributo rs: representa razão de solubilidade do gás no óleo.
  - atributo rsi: representa razão de solubilidade inicial do gás no óleo.
  - atributo we: respresenta o influxo acumulado do aquífero.
  - atributo m: representa a razão volume de gás na capa de gás pelo volume de óleo na zona de óleo.
  - método COilInf: representa o construtor da classe.
- Classe CInjAgua: representa os conceitos de injeção de água no reservatório.
    - atributo q ( $m^3/dia$ ): Taxa de injeção de água.
    - atributo cw [ $(kgf/cm^2)^{-1}$ ]: Compressibilidade da água.
    - atributo pwi ( $kgf/cm^2$ ): Pressão inicial da água na injeção.
    - atributo método CInjAgua: Construtor da classe CInjAgua
  - Classe CInjGas: representa os conceitos de injeção de gás no reservatório.
    - atributo q ( $m^3/dia$ ): Taxa de injeção de gás .
    - atributo cg [ $(kgf/cm^2)^{-1}$ ]: Compressibilidade do gás.
    - atributo pwgi ( $kgf/cm^2$ ): Pressão inicial do gás na injeção.
    - método CInjGas: Construtor da classe CInjGas
  - Classe CInjPolimero: representa os conceitos de injeção de polímero no reservatório.
    - atributo q ( $m^3/dia$ ): Taxa de injeção de polímero.
    - atributo muw (cP): Viscosidade da água.
    - atributo mug (cP): Viscosidade do gás.
    - atributo cw [ $(kgf/cm^2)^{-1}$ ]: Compressibilidade da água.
    - atributo cpwi ( $kgf/cm^2$ ): Pressão inicial do polímero na injeção .
    - método CInjPolimero: Construtor da classe CInjPolimero
  - Classe CInjSurfactante: representa os conceitos de injeção de surfactante no reservatório.

- atributo **q** ( $m^3/\text{dia}$ ): Taxa de injeção de surfactante.
  - atributo **muw** (cP): Viscosidade da água.
  - atributo **mug** (cP): Viscosidade do gás.
  - atributo **cw** [ $(\text{kgf/cm}^2)^{-1}$ ]: Compressibilidade da água.
  - atributo **cs** [ $(\text{kgf/cm}^2)^{-1}$ ]: Compressibilidade do surfactante.
  - atributo **cpwi** ( $\text{kgf/cm}^2$ ): Pressão inicial do surfactante na injeção.
  - método **CInjSurfactante**: Construtor da classe **CInjSurfactante**.
- Classe **CInjVapor**: representa os conceitos de injeção de vapor no reservatório. Construtor da classe **CInjVapor**.
    - atributo **q** ( $m^3/\text{dia}$ ): Taxa de injeção de vapor.
    - atributo **muw** (cP): Viscosidade da água.
    - atributo **mug** (cP): Viscosidade do gás.
    - atributo **cw** [ $(\text{kgf/cm}^2)^{-1}$ ]: Compressibilidade da água.
    - atributo **cv** [ $(\text{kgf/cm}^2)^{-1}$ ]: Compressibilidade do vapor.
    - atributo **cpwi** ( $\text{kgf/cm}^2$ ): Pressão inicial do vapor na injeção.
    - método **CInjVapor**: Construtor da classe **CInjVapor**.
  - Classe **CInjGasLift**: representa os conceitos de injeção de gás lift no reservatório.
    - atributo **q** ( $m^3/\text{dia}$ ): Taxa de injeção de gás lift.
    - atributo **cg** [ $(\text{kgf/cm}^2)^{-1}$ ]: Compressibilidade do gás.
    - atributo **pwgi** ( $\text{kgf/cm}^2$ ): Pressão inicial do gás lift na injeção. -
    - atributo **depth** (m): Profundidade do poço
    - método **CInjGasLift**: Construtor da classe **CInjGasLift**.

## 4.3 Diagrama de seqüência – eventos e mensagens

### 4.3.1 Diagrama de sequência geral

Veja o diagrama de seqüência na Figura 4.2. É uma representação gráfica da interação entre objetos em um sistema ao longo do tempo. Ele mostra a ordem das mensagens trocadas entre objetos à medida que uma operação é executada.

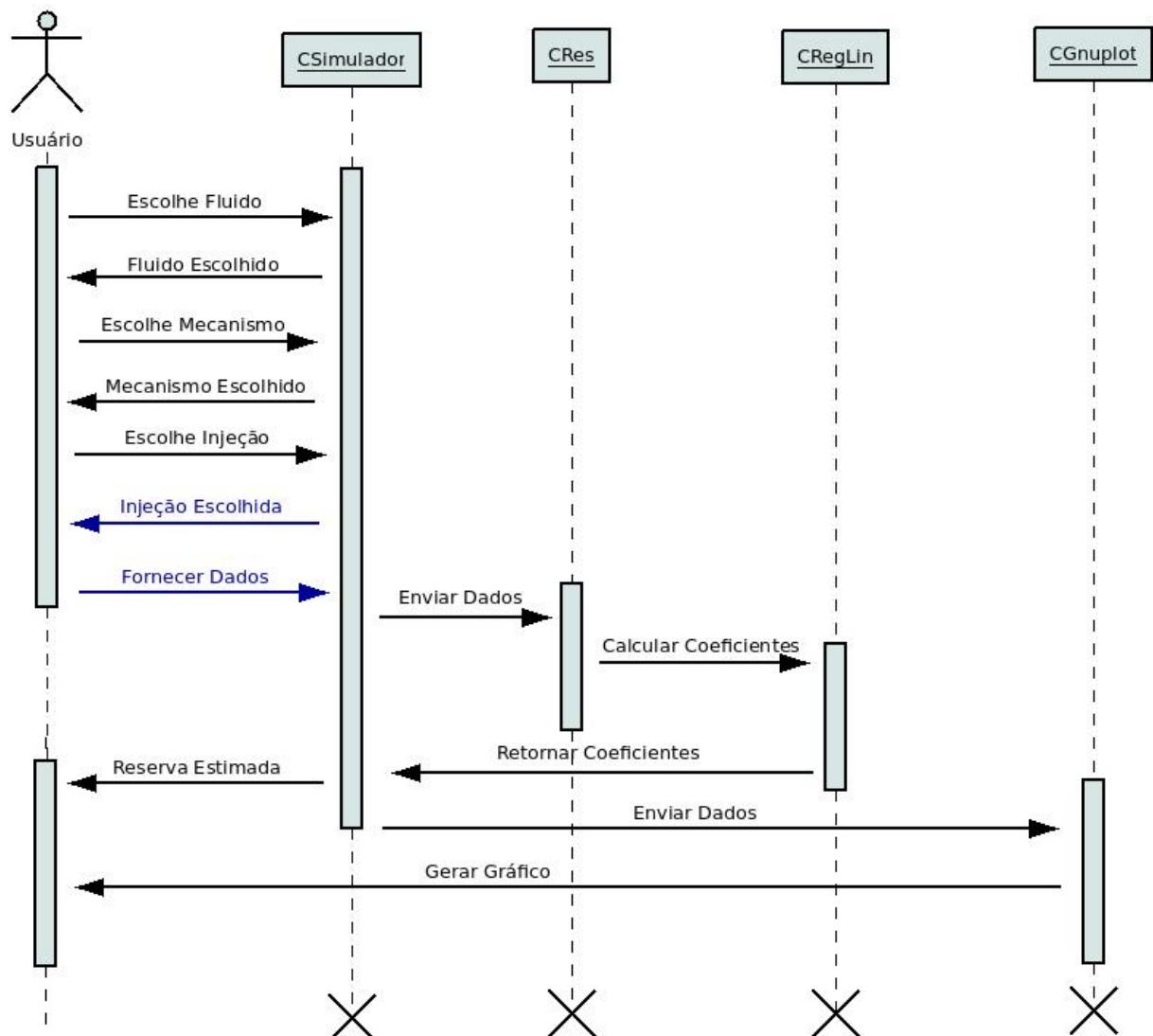


Figura 4.2: Diagrama de seqüência

### 4.3.2 Diagrama de sequência específico

Veja o diagrama de seqüência na Figura 4.3. É uma representação visual que descreve a interação entre os componentes do software em um cenário específico dentro de um sistema. Ele mostra como esses componentes colaboram ao longo do tempo para realizar uma funcionalidade específica ou completar um cenário de uso.

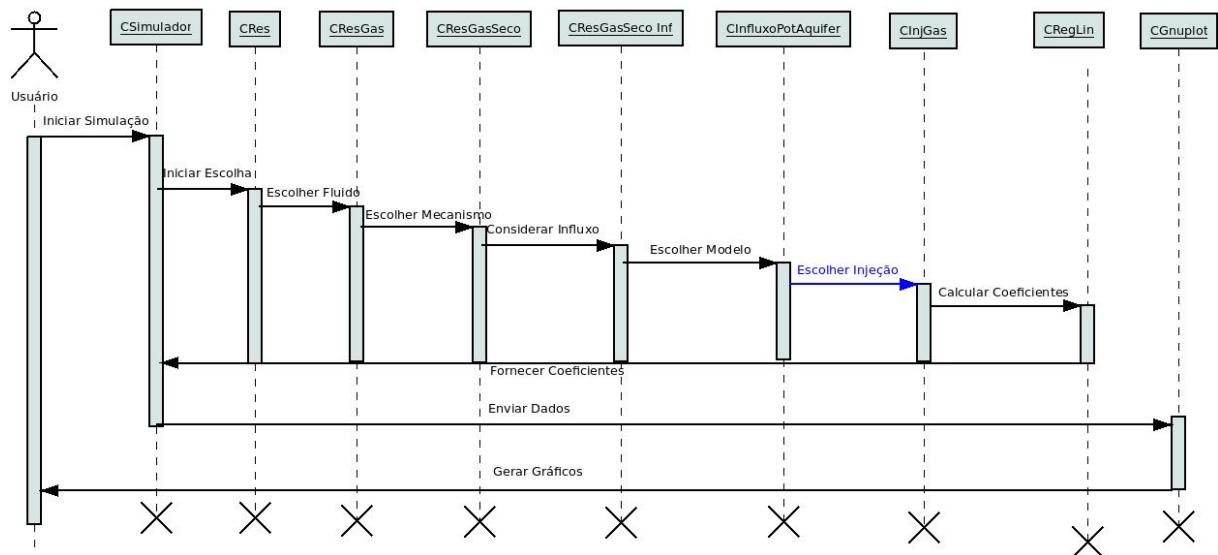


Figura 4.3: Diagrama de sequência específico

## 4.4 Diagrama de comunicação – colaboração

Veja na Figura 4.4 o diagrama de comunicação mostrando a sequência de simulação. No diagrama de comunicação o foco é a interação e a troca de mensagens e dados entre os objetos.

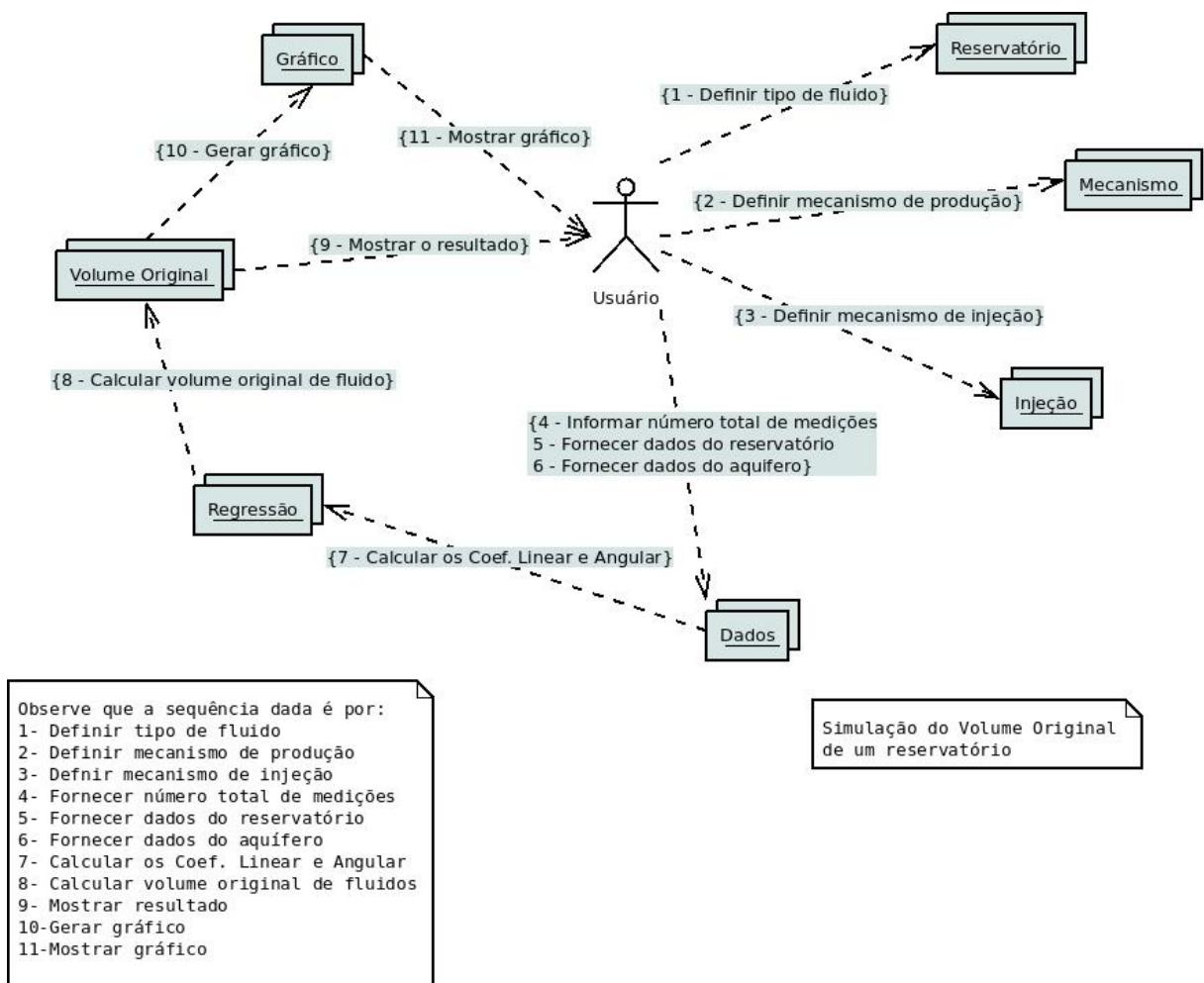


Figura 4.4: Diagrama de comunicação

## 4.5 Diagrama de estado

Veja na Figura 4.5 o diagrama de máquina de estado para uma ocorrência específica que desencadeia uma transição de estado. Representa os diversos estados que o objeto assume e os eventos que ocorrem ao longo de sua vida ou mesmo ao longo de um processo (histórico do objeto). É usado para modelar aspectos dinâmicos do programa.

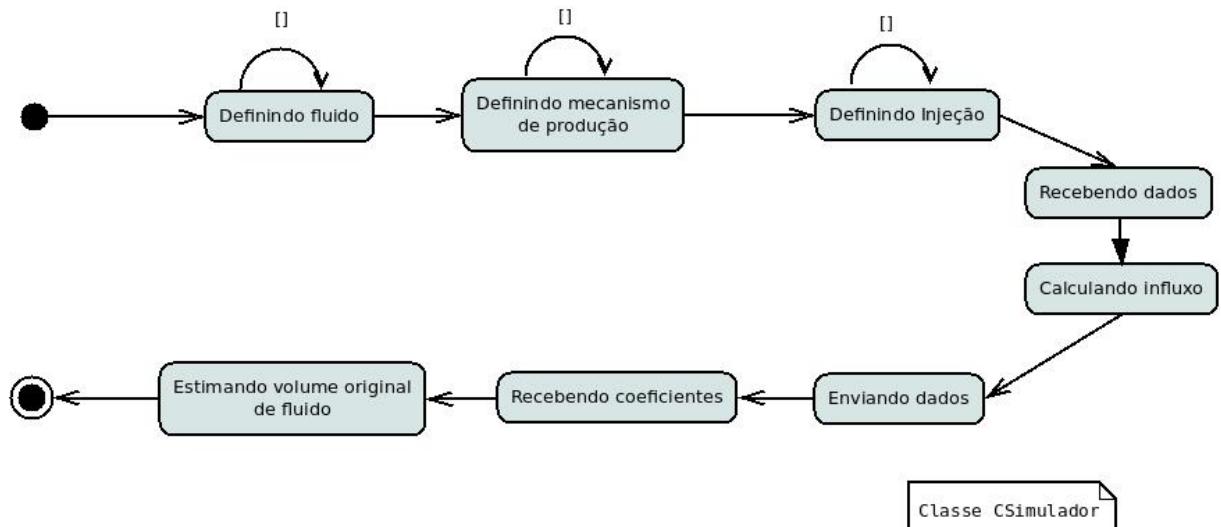


Figura 4.5: Diagrama de máquina de estado

## 4.6 Diagrama de atividades

Veja na Figura 4.6 o diagrama de atividades correspondente a uma atividade específica do diagrama de máquina de estado.

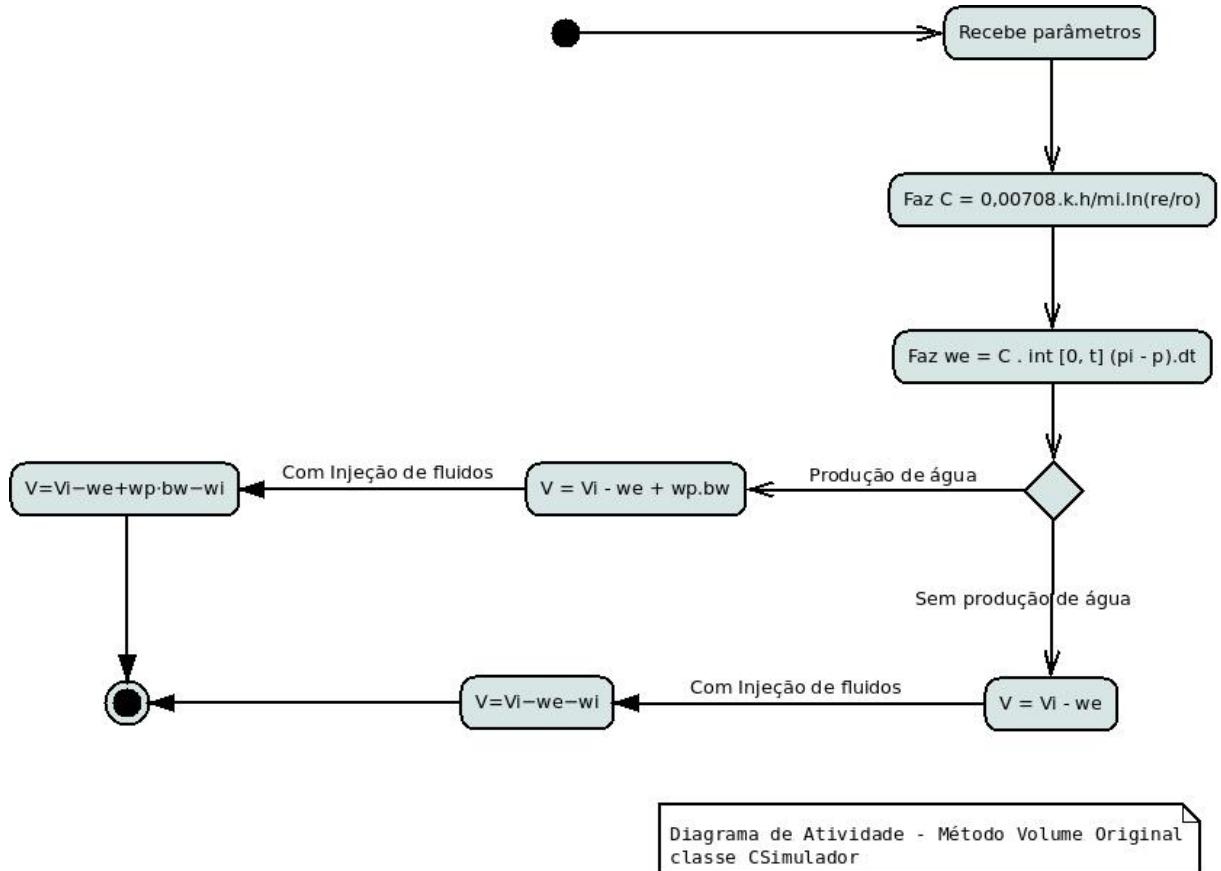


Figura 4.6: Diagrama de atividades

# Capítulo 5

## Projeto

Neste capítulo do projeto de engenharia veremos questões associadas ao projeto do sistema, incluindo protocolos, recursos, plataformas suportadas, implicações nos diagramas feitos anteriormente, diagramas de componentes e implantação. Na segunda parte revisamos os diagramas levando em conta as decisões do projeto do sistema.

### 5.1 Projeto do sistema

Depois da análise orientada a objeto desenvolve-se o projeto do sistema, o qual envolve etapas como a definição dos protocolos, da interface API, o uso de recursos, a subdivisão do sistema em subsistemas, a alocação dos subsistemas ao hardware e a seleção das estruturas de controle, a seleção das plataformas do sistema, das bibliotecas externas, dos padrões de projeto, além da tomada de decisões conceituais e políticas que formam a infraestrutura do projeto.

Segundo [Rumbaugh et al., 1994, Blaha and Rumbaugh, 2006], o projeto do sistema é a estratégia de alto nível para resolver o problema e elaborar uma solução. Gerando itens como:

1. Protocolos

- Protocolos de Comunicação Externa

- No contexto deste software, será estabelecido um protocolo de comunicação entre os diferentes elementos externos, especialmente dispositivos. Se o sistema envolve a utilização de nós em um cluster, será necessário considerar cuidadosamente os aspectos relacionados ao protocolo de comunicação entre esses nós. Essa definição garantirá uma integração eficiente e confiável, promovendo uma troca de dados suave entre os dispositivos.

- Protocolos de Comunicação Interna

- Internamente, o software definirá protocolos de comunicação entre diversos elementos, como objetos. Essa definição visa estabelecer diretrizes claras

para interações eficientes entre diferentes partes do sistema. A padronização interna contribuirá para a coesão do sistema, facilitando a manutenção e compreensão do código.

- Interface API de Bibliotecas e Sistema

- Será definida uma Interface API clara e bem documentada para todas as bibliotecas e sistemas integrados ao software. Isso proporcionará uma experiência consistente para os desenvolvedores que utilizarão ou estenderão funcionalidades do sistema. A documentação detalhada será uma parte essencial dessa definição, assegurando que os usuários compreendam como interagir efetivamente com as funcionalidades disponíveis.

- Formato dos Arquivos Gerados

- O formato dos arquivos gerados pelo software será definido com a preferência por formatos abertos, como arquivos txt e xml. Isso visa garantir que os dados gerados pelo sistema sejam acessíveis e interoperáveis, promovendo a flexibilidade na utilização desses dados por outros sistemas ou ferramentas. A escolha de formatos abertos contribui para a portabilidade e preservação a longo prazo dos dados.

## 2. Recursos

- Identificação e Alocação de Recursos Globais

- Neste projeto, será realizada a identificação e alocação dos recursos globais do sistema. Isso inclui definir como esses recursos serão alocados, utilizados, compartilhados e liberados durante a execução do software. Essa etapa implicará modificações no diagrama de componentes para refletir a distribuição e interação desses recursos.

- Necessidade de Banco de Dados

- Será feita a identificação da necessidade de utilização de um banco de dados no contexto deste software. Essa identificação levará a modificações nos diagramas de atividades e componentes para integrar eficientemente o banco de dados ao sistema. Será considerada a escolha do tipo de banco de dados mais apropriado para atender aos requisitos de armazenamento e recuperação de dados.

- Necessidade de Sistemas de Armazenamento em Massa

- No âmbito deste projeto, será identificada a necessidade de sistemas de armazenamento em massa. Exemplos incluem a possibilidade de integrar um storage em um sistema de cluster ou a implementação de sistemas de backup. Essa identificação implicará modificações nos diagramas de

atividades e componentes para incorporar adequadamente esses sistemas, garantindo um gerenciamento eficiente e seguro de grandes volumes de dados.

### 3. Controle

- Identificação e Seleção da Implementação de Controle
  - Neste projeto, será realizada a identificação e seleção da implementação de controle, considerando opções como controle sequencial ou concorrente, baseado em procedimentos ou eventos. Essa decisão terá impacto nas modificações do diagrama de execução, refletindo como o sistema responde a diferentes eventos e interações.
- Identificação de Condições Extremas e Prioridades
  - Serão identificadas as condições extremas que podem ocorrer durante a execução do software, e será dada prioridade a essas situações. Essa análise influenciará as modificações no diagrama de execução para garantir que o sistema responda eficazmente em circunstâncias críticas.
- Necessidade de Otimização
  - Neste projeto, será identificada a necessidade de otimização, considerando fatores como capacidade de memória e a preferência por vários discos rígidos menores em vez de um grande. Essa identificação levará a modificações nas estratégias de implementação e será refletida no diagrama de execução.
- Identificação e Definição de Loops de Controle e Escalas de Tempo
  - Serão identificados e definidos os loops de controle necessários para o funcionamento eficiente do sistema. Isso incluirá a especificação das escalas de tempo para diferentes processos. As modificações correspondentes serão realizadas no diagrama de execução para representar esses loops e escalas de tempo.
- Identificação de Concorrências
  - Serão identificadas as situações em que algoritmos podem ser implementados usando processamento paralelo para otimização de desempenho. Essa identificação terá impacto nas modificações do diagrama de execução, refletindo a implementação de concorrências quando aplicável.

### 4. Plataformas

- Identificação das Estruturas Arquitetônicas Comuns

- Neste projeto, serão identificadas as estruturas arquitetônicas comuns necessárias para o funcionamento do software. Isso incluirá a definição de componentes essenciais que serão utilizados de forma geral no sistema.
- Identificação de Subsistemas Relacionados à Plataforma Selecionada
  - Serão identificados subsistemas que estão diretamente relacionados à plataforma escolhida para a implementação do software. Essa identificação poderá implicar em modificações nos diagramas de pacotes e componentes, refletindo a integração desses subsistemas.
- Identificação e Definição das Plataformas a Serem Suportadas
  - Neste projeto, será realizada a identificação e definição das plataformas que o software irá suportar. Isso incluirá especificações relacionadas ao hardware, sistema operacional e linguagem de software. Essa decisão terá impacto direto na forma como o software será implementado e distribuído.
- Seleção de Bibliotecas Externas
  - Serão selecionadas as bibliotecas externas necessárias para complementar as funcionalidades do software. Essa seleção considerará a eficiência, confiabilidade e compatibilidade das bibliotecas com a plataforma escolhida.
- Seleção da Biblioteca para Interface Gráfica - GDI
  - Neste projeto, será realizada a seleção da biblioteca gráfica GDI (Graphics Device Interface) para a construção da interface gráfica do software. Essa escolha será baseada em critérios de desempenho, compatibilidade e recursos oferecidos pela biblioteca.
- Seleção do Ambiente de Desenvolvimento para Interface de Desenvolvimento - IDE
  - Será selecionado o ambiente de desenvolvimento que melhor se adequa à construção da interface de desenvolvimento do software. Isso incluirá considerações sobre facilidade de uso, suporte a linguagens de programação escolhidas e ferramentas disponíveis.

## 5. Padrões de projeto

- Identificação de Padrões de Projeto
  - Neste projeto, será realizada a identificação e documentação dos padrões de projeto que serão aplicados. Os padrões de projeto representam soluções consolidadas para problemas recorrentes no desenvolvimento de software.
- Construção da Biblioteca de Padrões da Empresa

- Os padrões de projeto identificados serão compilados e organizados para formar uma biblioteca de padrões da empresa. Esta biblioteca servirá como referência para futuros projetos, promovendo consistência e boas práticas no desenvolvimento de software.
- Aplicação dos Padrões de Projeto no Desenvolvimento
  - Durante a fase de implementação, os padrões de projeto identificados serão aplicados de acordo com as necessidades específicas do software. Isso garantirá uma abordagem consistente e eficiente na resolução de problemas de design e desenvolvimento.
- Revisão e Adaptação dos Padrões (se necessário)
  - Caso seja identificada a necessidade de revisão ou adaptação dos padrões de projeto durante o ciclo de vida do projeto, ajustes serão realizados. Isso pode ocorrer com base em experiências específicas do projeto ou mudanças nas melhores práticas da indústria.

## 5.2 Projeto orientado a objeto – POO

O projeto orientado a objeto é a etapa posterior ao projeto do sistema. Baseia-se na análise, mas considera as decisões do projeto do sistema. Acrescenta a análise desenvolvida e as características da plataforma escolhida (hardware, sistema operacional e linguagem de software). Passa pelo maior detalhamento do funcionamento do software, acrescentando atributos e métodos que envolvem a solução de problemas específicos não identificados durante a análise.

Envolve a otimização da estrutura de dados e dos algoritmos, a minimização do tempo de execução, de memória e de custos. Existe um desvio de ênfase para os conceitos da plataforma selecionada.

Exemplo: na análise você define que existe um método para salvar um arquivo em disco, define um atributo nomeDoArquivo, mas não se preocupa com detalhes específicos da linguagem. Já no projeto, você inclui as bibliotecas necessárias para acesso ao disco, cria um objeto específico para acessar o disco, podendo, portanto, acrescentar novas classes àquelas desenvolvidas na análise.

### Efeitos do projeto no modelo estrutural

- Adicionar nos diagramas de pacotes as bibliotecas e subsistemas selecionados no projeto do sistema (exemplo: a biblioteca gráfica selecionada).
  - Neste projeto blablabla
- Novas classes e associações oriundas das bibliotecas selecionadas e da linguagem escolhida devem ser acrescentadas ao modelo.

- Neste projeto blablabla
- Estabelecer as dependências e restrições associadas à plataforma escolhida.
  - Neste projeto blablabla

### Efeitos do projeto no modelo dinâmico

- Revisar os diagramas de seqüência e de comunicação considerando a plataforma escolhida.
  - Neste projeto blablabla
- Verificar a necessidade de se revisar, ampliar e adicionar novos diagramas de máquinas de estado e de atividades.
  - Neste projeto blablabla

### Efeitos do projeto nos atributos

- Atributos novos podem ser adicionados a uma classe, como, por exemplo, atributos específicos de uma determinada linguagem de softwareção (acesso a disco, ponteiros, constantes e informações correlacionadas).
  - Neste projeto blablabla

### Efeitos do projeto nos métodos

- Em função da plataforma escolhida, verifique as possíveis alterações nos métodos. O projeto do sistema costuma afetar os métodos de acesso aos diversos dispositivos (exemplo: hd, rede).
  - Neste projeto blablabla
- De maneira geral os métodos devem ser divididos em dois tipos: i) tomada de decisões, métodos políticos ou de controle; devem ser claros, legíveis, flexíveis e usam polimorfismo. ii) realização de processamentos, podem ser otimizados e em alguns casos o polimorfismo deve ser evitado.
  - Neste projeto blablabla
- Algoritmos complexos podem ser subdivididos. Verifique quais métodos podem ser otimizados. Pense em utilizar algoritmos prontos como os da STL (algoritmos genéricos).

- Neste projeto blablabla
- Responda a pergunta: os métodos da classes estão dando resposta às responsabilidades da classe?
  - Neste projeto blablabla
- Revise os diagramas de classes, de seqüência e de máquina de estado.
  - Neste projeto blablabla

### Efeitos do projeto nas heranças

- Reorganização das classes e dos métodos (criar métodos genéricos com parâmetros que nem sempre são necessários e englobam métodos existentes).
  - Neste projeto blablabla
- Abstração do comportamento comum (duas classes podem ter uma superclasse em comum).
  - Neste projeto blablabla
- Utilização de delegação para compartilhar a implementação (quando você cria uma herança irreal para reaproveitar código). Usar com cuidado.
  - Neste projeto blablabla
- Revise as heranças no diagrama de classes.
  - Neste projeto blablabla

### Efeitos do projeto nas associações

- Deve-se definir na fase de projeto como as associações serão implementadas, se obedecerão um determinado padrão ou não.
  - Neste projeto blablabla
- Se existe uma relação de "muitos", pode-se implementar a associação com a utilização de um dicionário, que é um mapa das associações entre objetos. Assim, o objeto A acessa o dicionário fornecendo uma chave (um nome para o objeto que deseja acessar) e o dicionário retorna um valor (um ponteiro) para o objeto correto.
  - Neste projeto blablabla

- Evite percorrer várias associações para acessar dados de classes distantes. Pense em adicionar associações diretas.
  - Neste projeto blablabla

### Efeitos do projeto nas otimizações

- Faça uma análise de aspectos relativos à otimização do sistema. Lembrando que a otimização deve ser desenvolvida por analistas/desenvolvedores experientes.
  - Neste projeto blablabla
- Identifique pontos a serem otimizados em que podem ser utilizados processos concorrentes.
  - Neste projeto blablabla
- Pense em incluir bibliotecas otimizadas.
- Se o acesso a determinados objetos (atributos/métodos) requer um caminho longo (exemplo: A->B->C->D.atributo), pense em incluir associações extras (exemplo: A-D.atributo).
  - Neste projeto blablabla
- Atributos auxiliares podem ser incluídos.
  - Neste projeto blablabla
- A ordem de execução pode ser alterada.
  - Neste projeto blablabla
- Revise as associações nos diagramas de classes.
  - Neste projeto blablabla

Depois de revisados os diagramas da análise você pode montar dois diagramas relacionados à infraestrutura do sistema. As dependências dos arquivos e bibliotecas podem ser descritos pelo diagrama de componentes, e as relações e dependências entre o sistema e o hardware podem ser ilustradas com o diagrama de implantação.

### 5.3 Diagrama de componentes

O diagrama de componentes mostra a forma como os componentes do software se relacionam, suas dependências. Inclui itens como: componentes, subsistemas, executáveis, nós, associações, dependências, generalizações, restrições e notas. Exemplos de componentes são bibliotecas estáticas, bibliotecas dinâmicas, dlls, componentes Java, executáveis, arquivos de disco, código-fonte.

Veja na Figura 5.1 um exemplo de diagrama de componentes. Observe que este inclui muitas dependências, ilustrando as relações entre os arquivos. Por exemplo: o subsistema biblioteca inclui os arquivos das classes A e B, e a geração dos objetos A.obj e B.obj depende dos arquivos A.h, A.cpp, B.h e B.cpp. A geração da biblioteca depende dos arquivos A.obj e B.obj. O subsistema biblioteca Qt, um subsistema externo, inclui os arquivos de código da biblioteca Qt e a biblioteca em si. O subsistema banco de dados representa o banco de dados utilizado pelo sistema e tem uma interface de acesso que é utilizada pelo software para acesso aos dados armazenados no banco de dados. O software executável a ser gerado depende da biblioteca gerada, dos arquivos da biblioteca Qt, do módulo de arquivos MinhaJanela e do banco de dados.

Algumas observações úteis para o diagrama de componentes:

- De posse do diagrama de componentes, temos a lista de todos os arquivos necessários para compilar e rodar o software.
- Observe que um assunto/pacote pode se transformar em uma biblioteca e será incluído no diagrama de componentes.
- A ligação entre componentes pode incluir um estereótipo indicando o tipo de relacionamento ou algum protocolo utilizado.

Neste projeto blablabla

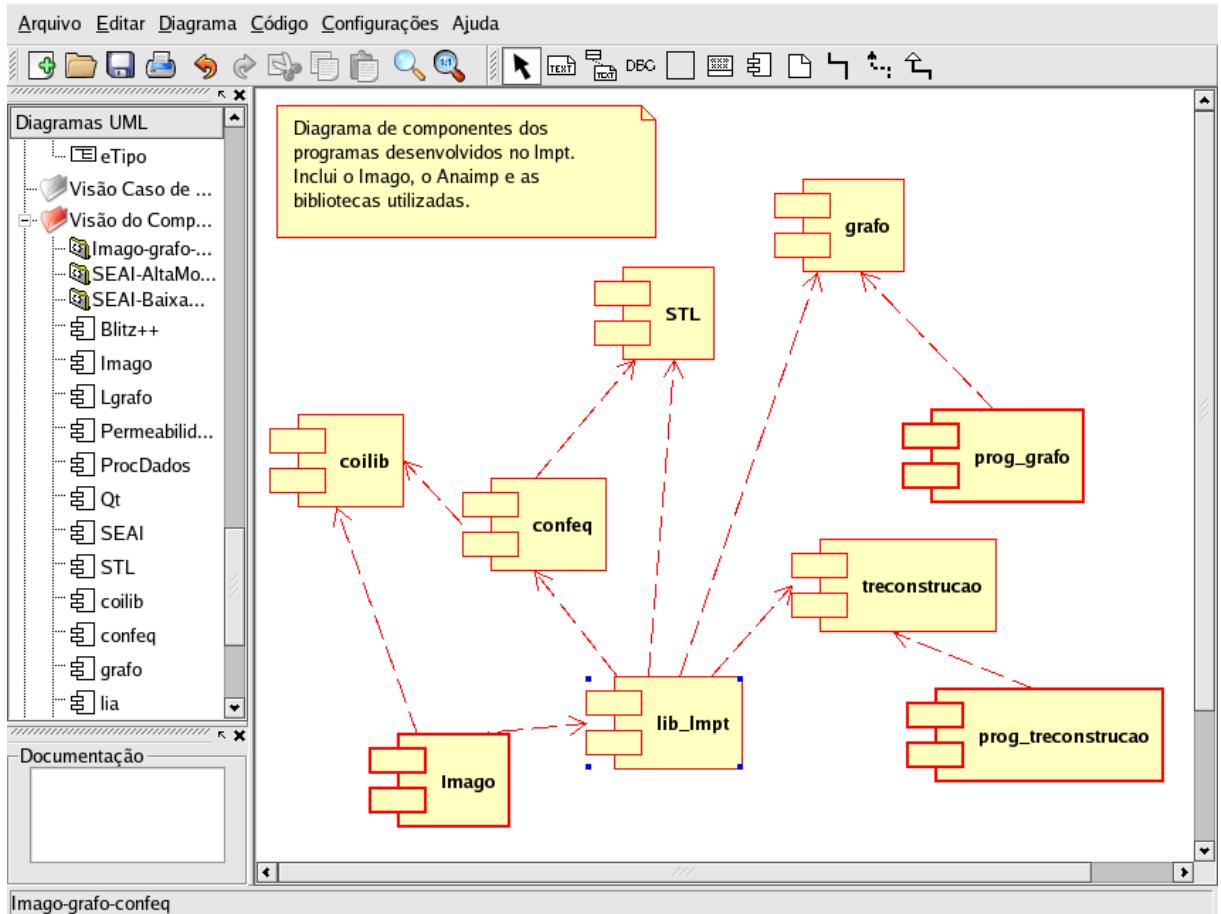


Figura 5.1: Diagrama de componentes

## 5.4 Diagrama de implantação

O diagrama de implantação é um diagrama de alto nível que inclui relações entre o sistema e o hardware e que se preocupa com os aspectos da arquitetura computacional escolhida. Seu enfoque é o hardware, a configuração dos nós em tempo de execução.

O diagrama de implantação deve incluir os elementos necessários para que o sistema seja colocado em funcionamento: computador, periféricos, processadores, dispositivos, nós, relacionamentos de dependência, associação, componentes, subsistemas, restrições e notas.

Veja na Figura 5.2 um exemplo de diagrama de implantação de um cluster. Observe a presença de um servidor conectado a um switch. Os nós do cluster (ou clientes) também estão conectados ao switch. Os resultados das simulações são armazenados em um servidor de arquivos (*storage*).

Pode-se utilizar uma anotação de localização para identificar onde determinado componente está residente, por exemplo {localização: sala 3}.

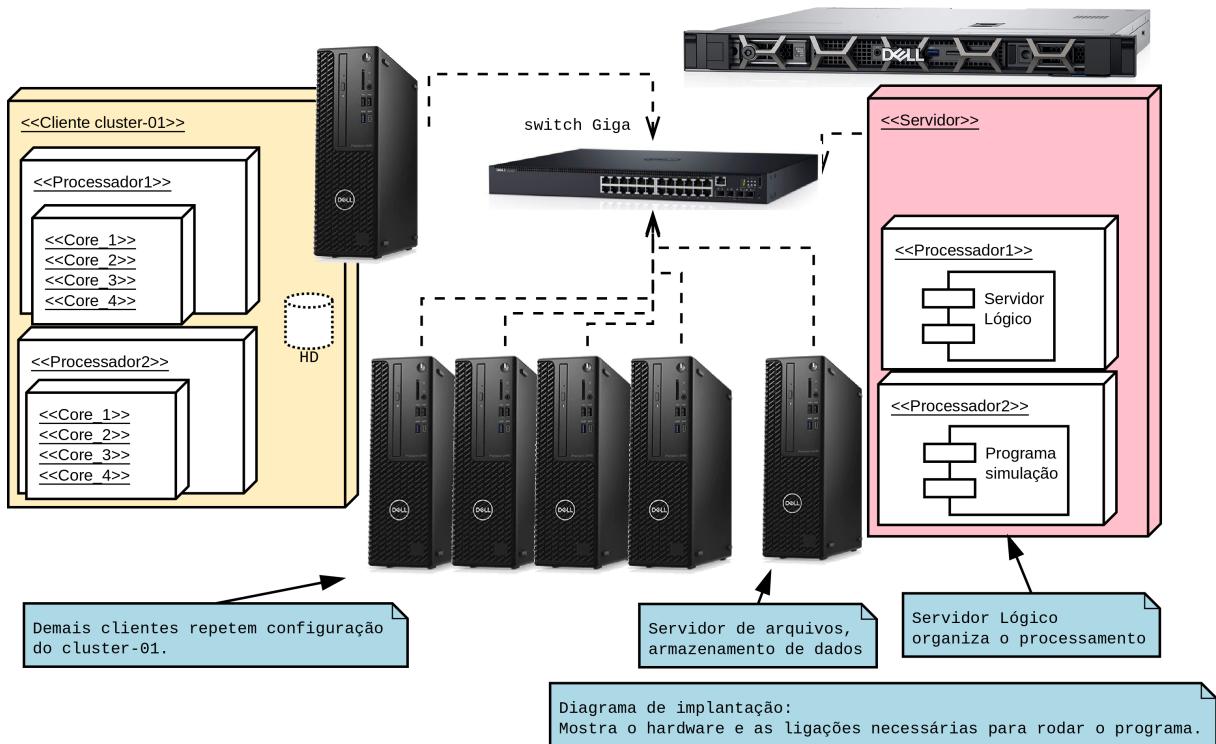


Figura 5.2: Diagrama de implantação

**Nota:**

Não perca de vista a visão do todo; do projeto de engenharia como um todo. Cada capítulo, cada seção, cada parágrafo deve se encaixar. Este é um diferencial fundamental do engenheiro em relação ao técnico, a capacidade de desenvolver projetos, de ver o todo e suas diferentes partes, de modelar processos/sistemas/produtos de engenharia.

#### 5.4.1 Lista de características <<features>>

No final do ciclo de concepção e análise chegamos a uma lista de características <<features>> que teremos de implementar.

Após a análises desenvolvidas e considerando o requisito de que este material deve ter um formato didático, chegamos a seguinte lista:

- v0.1
  - Lista de classes a serem implementadas
  - Testes
- v0.3
  - Lista de classes a serem implementadas
  - Testes

- v0.7
  - Lista de classes a serem implementadas
  - Testes

### 5.4.2 Tabela classificação sistema

A Tabela a seguir é utilizada para classificação do sistema desenvolvido. Deve ser preenchida na etapa de projeto e revisada no final, quando o software for entregue na sua versão final.

Licença:	[X] livre GPL-v3 [ ] proprietária
Engenharia de software:	[ ] tradicional [X] ágil [ ] outras
Paradigma de programação:	[ ] estruturada [X] orientado a objeto - POO [ ] funcional
Modelagem UML:	[X] básica [X] intermediária [ ] avançada
Algoritmos:	[X] alto nível [X] baixo nível
	implementação: [ ] recursivo ou [X] iterativo; [X] determinístico ou [ ] não-determinístico; [ ] exato ou [X] aproximado
	concorrências: [X] serial [X] concorrente [X] paralelo
	paradigma: [X] dividir para conquistar [ ] programação linear [ ] transformação/ redução [ ] busca e enumeração [ ] heurístico e probabilístico [ ] baseados em pilhas
Software:	[ ] de base [X] aplicados [ ] de cunho geral [X] específicos para determinada área [X] educativo [X] científico
	instruções: [X] alto nível [ ] baixo nível
	otimização: [X] serial não otimizado [X] serial otimizado [X] concorrente [X] paralelo [ ] vetorial
	interface do usuário: [ ] kernel numérico [ ] linha de comando [ ] modo texto [X] híbrida (texto e saídas gráficas) [ ] modo gráfico (ex: Qt) [ ] navegador
Recursos de C++:	[X] C++ básico (FCC): variáveis padrões da linguagem, estruturas de controle e repetição, estruturas de dados, struct, classes(objetos, atributos, métodos), funções; entrada e saída de dados ( <i>streams</i> ), funções de cmath

	[X] C++ intermediário: funções lambda. Ponteiros, referências, herança, herança múltipla, polimorfismo, sobrecarga de funções e de operadores, tipos genéricos (templates), <i>smart pointers</i> . Diretrizes de pré-processador, classes de armazenamento e modificadores de acesso. Estruturas de dados: enum, uniões. Bibliotecas: entrada e saída acesso com arquivos de disco, redirecionamento. Bibliotecas: <i>filesystem</i>
	[X] C++ intermediário 2: A biblioteca de gabaritos de C++ (a STL), containers, iteradores, objetos funções e funções genéricas. Nocões de processamento paralelo (múltiplas threads, uso de <i>thread, join</i> e <i>mutex</i> ). Bibliotecas: <i>random, threads</i>
	[ ] C++ avançado: Conversão de tipos do usuário, especializações de templates, excessões. Cluster de computadores, processamento paralelo e concorrente, múltiplos processos (pipes, memória compartilhada, sinais). Bibliotecas: <i>expressões regulares, múltiplos processos</i>
Bibliotecas de C++:	[X] Entrada e saída de dados ( <i>streams</i> ) [X] cmath [X] <i>filesystem</i> [ ] random [X] <i>threads</i> [ ] <i>expressões regulares</i> [ ] <i>múltiplos processos</i>
Bibliotecas externas:	[X] CGnuplot [X] QCustomPlot [ ] Qt diálogos [ ] QT Janelas/menus/BT _____
Ferramentas auxiliares:	Montador: [X] make [] cmake [X] qmake
IDE:	[X] Editor simples: kate/gedit/emacs [ ] kdevelop [ ] QT-Creator [ ] _____
SCV:	[ ] cvs [ ] svn [X] git
Disciplinas correlacionadas	[ ] estatística [ ] cálculo numérico [ ] modelamento numérico [X] análise e processamento de imagens

# Capítulo 6

## Ciclos de Planejamento/Detalhamento

Apresenta-se neste capítulo os ciclos de planejamento/detalhamento para as diferentes versões desenvolvidas.

### 6.1 Versão 0.1 - xxx

NAs EBM foram implementadas no segundo semestre de 2012 pelos alunos Leonardo Mesquita Caetano e Thiago Pessanha de Macedo. Veja Figura 6.1.

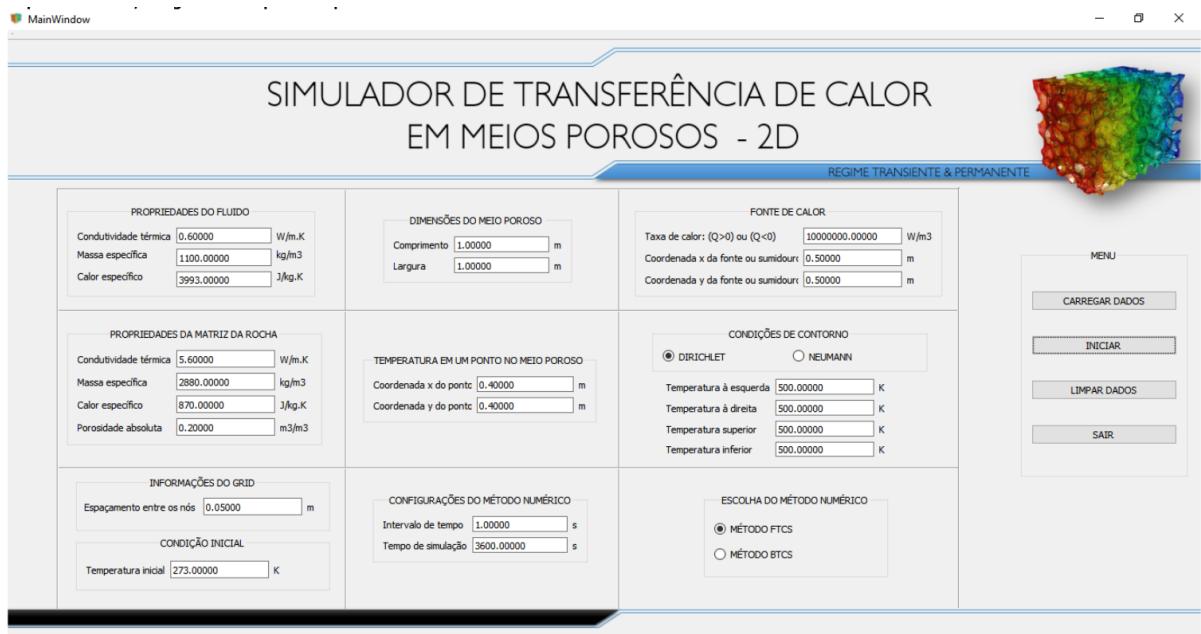


Figura 6.1: Versão 0.1, imagem do programa rodando

### 6.2 Versão 0.2 - xxx

Nesta segunda versão ...

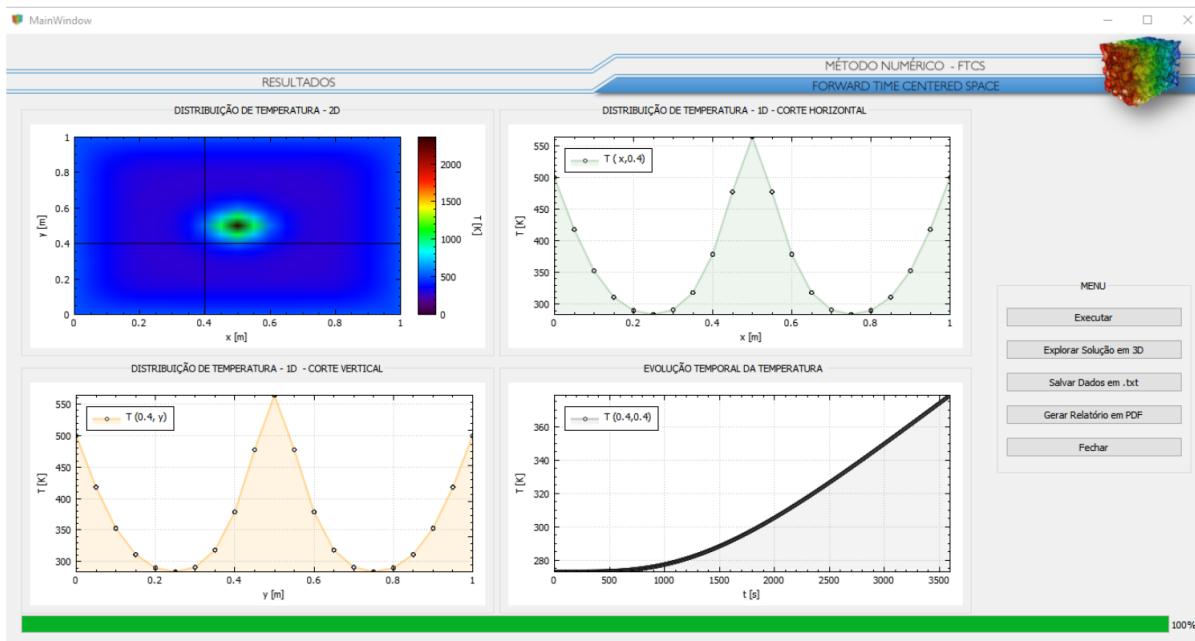


Figura 6.2: Versão 0.2, imagem do programa rodando

# Capítulo 7

## Ciclos Construção - Implementação

Neste capítulo, são apresentados os códigos fonte implementados.

**Nota:** os códigos devem ser documentados usando padrão **javadoc**. Posteriormente usar o programa **doxygen** para gerar a documentação no formato html.

- Veja informações gerais aqui <http://www.doxygen.org/>.
- Veja exemplo aqui <http://www.stack.nl/~dimitri/doxygen/manual/docblocks.html>.

**Nota:** ao longo deste capítulo usamos inclusão direta de arquivos externos usando o pacote *listings* do L<sup>A</sup>T<sub>E</sub>X. Maiores detalhes de como a saída pode ser gerada estão disponíveis nos links abaixo.

- [http://en.wikibooks.org/wiki/LaTeX/Source\\_Code\\_Listings](http://en.wikibooks.org/wiki/LaTeX/Source_Code_Listings).
- <http://mirrors.ctan.org/macros/latex/contrib/listings/listings.pdf>.

### 7.1 Código fonte

Apresenta-se a seguir um conjunto de classes (arquivos .h e .cpp) além do programa **main**.

Apresenta-se na listagem 7.1 o arquivo com código da classe **CSimulador-nome**.

---

Listagem 7.1: Arquivo de cabeçalho da classe **CSimulador-nome**

```
1// Este programa exemplifica a estrutura de um programa típico em C
2// Note que no arquivo .h documentamos a interface; a forma de uso;
3// No arquivo .cpp detalhes dos códigos; lógica numérica-
4#include <string>
5#include <vector>
6
```

```
7/** Breve descrição da classe termina com ponto.
8 * ... descrição detalhada da classe...
9 * ... pode ter várias linhas...
10*/
11class CAplicacao
12{
13public:
14    /// Descrição breve do construtor.
15    /** Descrição detalhada do construtor.
16     * ....blablabla....
17     */
18    CAplicacao()           {};
19
20    /// Descrição breve do construtor.
21    /** Descrição detalhada do construtor.
22     * ....blablabla....
23     */
24    ~CAplicacao()          {};
25
26    /// Apenas exibe mensagem na tela.
27    void Run();
28
29    /// Seta valor de x
30    void X( int _x)        { x = _x; }
31
32    /// Retorna valor de x
33    int X()                { return x; }
34
35private:
36    /// Descrição breve do método M1.
37    /**
38     * Descrição detalhada....
39     * Posso incluir informações sobre parâmetros e retorno.
40     * @param a um inteiro que representa ....
41     * @param s uma string que representa ....
42     * @return retorna ...
43     */
44    int M1(int a, std::string s);
45
46    /// Descrição breve do atributo...
47    /** Descrição detalhada do atributo... */
48    std::vector<int> vy;
```

```

49
50     /// Descrição breve do atributo...
51     int x;
52
53     int z; ///< Descrição breve (use apenas se for bem curta!).
54
55     /// Enum representa (descrição breve).
56     /** Descrição detalhada. */
57     enum Enum {
58         EVal1, ///< Breve descrição EVal1.
59         EVal2, ///< Breve descrição EVal2.
60         EVal3 ///< Breve descrição EVal3.
61     };
62
63     /// Descrição breve.
64     /** Descrição detalhada. */
65     Enum variavelDoTipoEnumeracao;
66
67 };
```

---

Apresenta-se na listagem 7.2 o arquivo de implementação da classe **CAplicacao**.

Listagem 7.2: Arquivo de implementação da classe CSimulador-nome

---

```

1// Este programa exemplifica a estrutura de um programa típico em
C++
2#include <iostream>
3
4// Inclui a declaração da classe
5#include "CAplicacao.h"
6
7/** Note que no arquivo .cpp não é necessário colocar novamente a
documentação
8 * que foi colocada no arquivo .h.
9 * A documentação no arquivo .cpp costuma usar o padrão básico de C
++ que é //
10 * e costuma estar mais diretamente relacionada a implementação em
sí,
11 * ou seja, aos detalhes numéricos e computacionais;
12 * detalhes e explicação das contas e da lógica computacional.
13 * */
14void CAplicacao::Run()
15{
16    // std::cout escreve na tela o texto "Bem-vindo ao C++!"
```

```

17 std::cout << "Bem-vindo ao C++!" << std::endl;
18 }
```

---

Apresenta-se na listagem 7.3 o programa que usa a classe `CSimulador-nome`.

Listagem 7.3: Arquivo de implementação da função `main()`

---

```

1
2/** Este programa exemplifica a estrutura/layout de um programa
   típico em C++ */
3
4// Inclui o arquivo "CAplicacao.h" que tem a declaração da classe
   CAplicacao
5#include "CAplicacao.h"
6
7/// A função main(), retorna um inteiro, se chama main() e não tem
   nenhum parâmetro
8int main ()
9{
10  CAplicacao ap; // Cria objeto do tipo CAplicacao com nome ap
11
12  ap.Run (); // Executa o método Run() do objeto ap
13
14  return 0; // A função main() deve retornar um inteiro
   // o zero indica que o programa terminou bem.
15
16}
```

---

Bem vindo ao C++!

### Nota:

Não perca de vista a visão do todo; do projeto de engenharia como um todo. Cada capítulo, cada seção, cada parágrafo deve se encaixar. Este é um diferencial fundamental do engenheiro em relação ao técnico, a capacidade de desenvolver projetos, de ver o todo e suas diferentes partes, de modelar processos/sistemas/produtos de engenharia.

# Capítulo 8

## Teste

Todo projeto de engenharia passa por uma etapa de testes. Neste capítulo apresentamos alguns testes do software desenvolvido. Estes testes devem dar resposta aos diagramas de caso de uso inicialmente apresentados (diagramas de caso de uso geral e específicos).

### 8.1 Teste 1: Descrição

No início apresente texto explicativo do teste:

- O que esta sendo testado?
- Como o teste vai ser realizado?
- Como o programa será validado?
- Resultados e análises

A seguir apresente texto explicando a sequência do teste e imagens do programa (captura de tela).

coloque aqui texto falando do diagrama de pacotes, referencia a figura. Veja Figura 8.1.

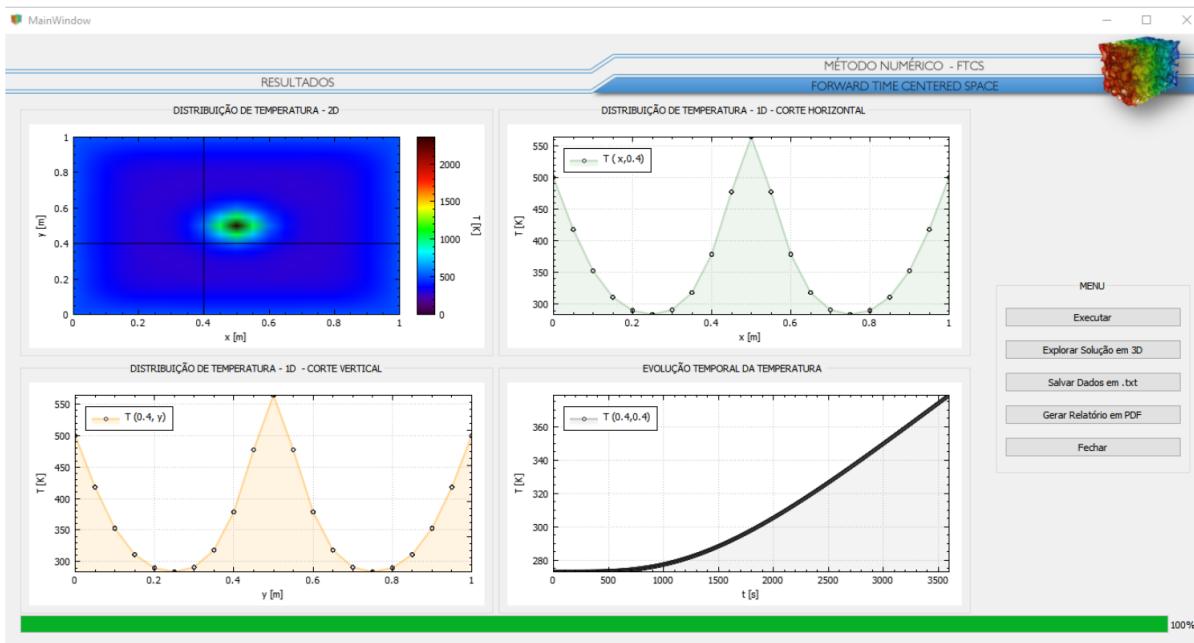


Figura 8.1: Tela do programa mostrando xxx

## 8.2 Teste 2: Descrição

No início apresente texto explicativo do teste:

- O que esta sendo testado?
- Como o teste vai ser realizado?
- Como o programa será validado?
- Resultados e análises

A seguir apresente texto explicando a sequência do teste e imagens do programa (captura de tela).

Coloque aqui texto falando do diagrama de pacotes, referencie a figura. Veja Figura 8.2.

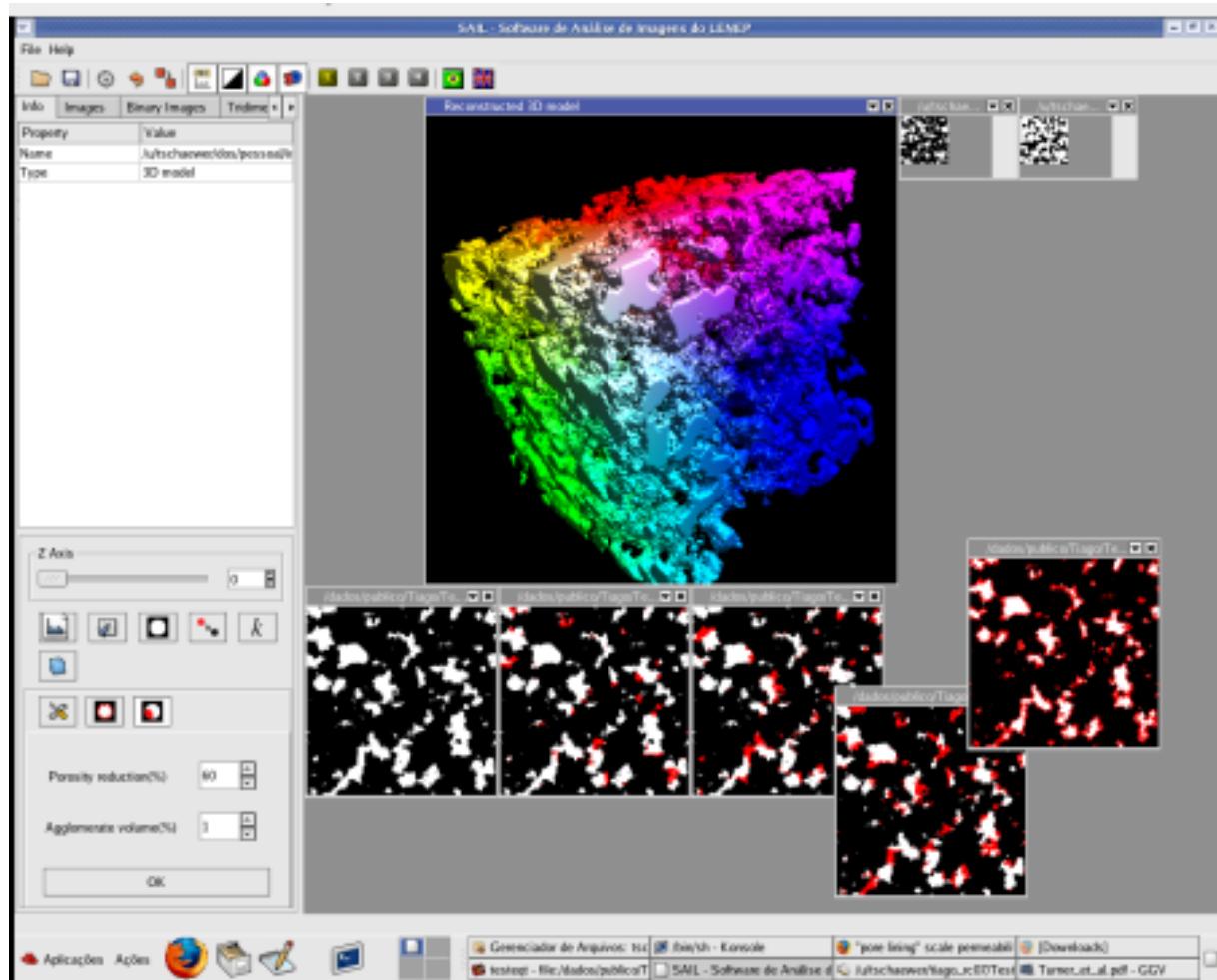


Figura 8.2: Tela do programa mostrando xxx

### 8.3 Teste 3: Descrição

No início apresente texto explicativo do teste:

- O que esta sendo testado?
- Como o teste vai ser realizado?
- Como o programa será validado?
- Resultados e análises

A seguir apresente texto explicando a sequência do teste e imagens do programa (captura de tela).

coloque aqui texto falando do diagrama de pacotes, referenciaie a figura. Veja Figura 8.3.

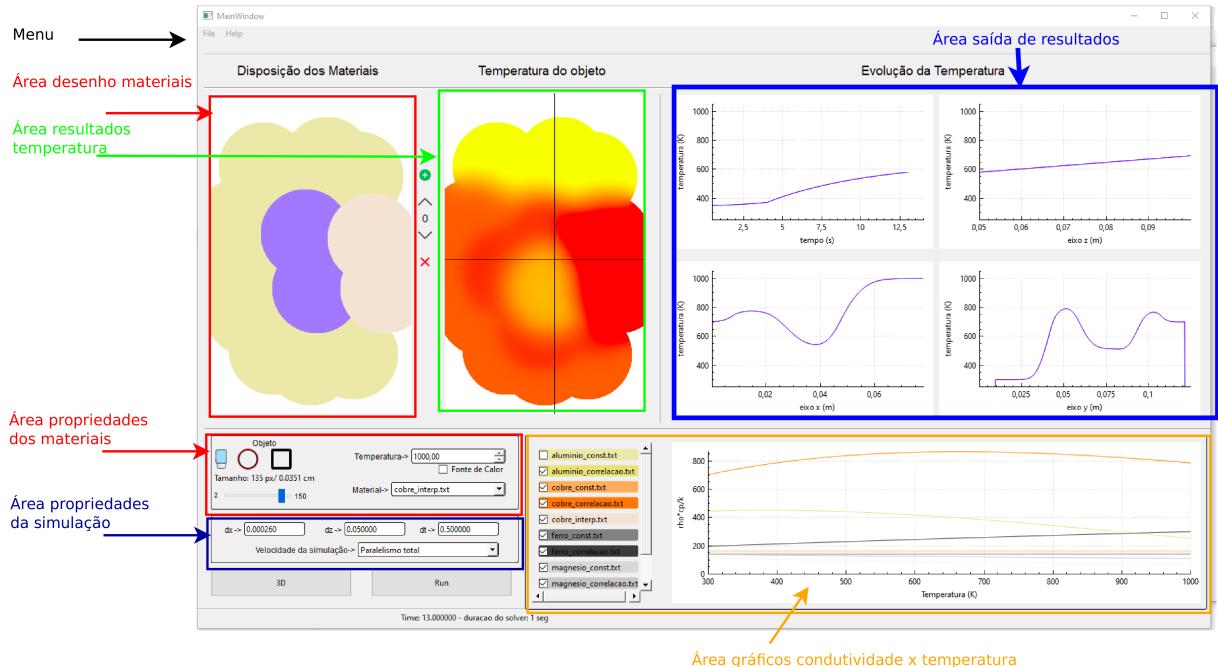


Figura 8.3: Tela do programa mostrando xxx

**Nota:**

Não perca de vista a visão do todo; do projeto de engenharia como um todo. Cada capítulo, cada seção, cada parágrafo deve se encaixar. Este é um diferencial fundamental do engenheiro em relação ao técnico, a capacidade de desenvolver projetos, de ver o todo e suas diferentes partes, de modelar processos/sistemas/produtos de engenharia.

# Capítulo 9

## Documentação para o Desenvolvedor

Todo projeto de engenharia precisa ser bem documentado. Neste sentido, apresenta-se neste capítulo documentações extras para o desenvolvedor. Ou seja, instruções para pessoas que venham a dar continuidade a este projeto de engenharia.

**Nota:** O manual do usuário é apresentado em um documento separado. Veja diretório "doc/ManualDoUsuario".

### 9.1 Dependências para compilar o software

Para compilar o software é necessário atender as seguintes dependências:

- Instalar o compilador `g++` da GNU disponível em <http://gcc.gnu.org>. Para instalar no GNU/Linux use o comando `yum install gcc`.
- Biblioteca CGnuplot; os arquivos para acesso a biblioteca CGnuplot devem estar no diretório com os códigos do software;
- O software `gnuplot`, disponível no endereço <http://www.gnuplot.info/>, deve estar instalado. É possível que haja necessidade de setar o caminho para execução do `gnuplot`.
- .
- .

### 9.2 Como gerar a documentação usando doxygen

A documentação do código do software deve ser feita usando o padrão JAVADOC, conforme apresentada no Capítulo - Documentação, do livro texto da disciplina. Depois de documentar o código, use o software `doxygen` para gerar a documentação do desenvolvedor no formato html. O software `doxygen` lê os arquivos com os códigos (`*.h` e `*.cpp`) e gera uma documentação muito útil e de fácil navegação no formato html.

- Veja informações sobre uso do formato JAVADOC em:
  - <http://www.stack.nl/~dimitri/doxygen/manual/docblocks.html>
- Veja informações sobre o software doxygen em
  - <http://www.stack.nl/~dimitri/doxygen/>

Passos para gerar a documentação usando o doxygen.

- Documente o código usando o formato JAVADOC. Um bom exemplo de código documentado é apresentado nos arquivos da biblioteca CGnuplot, abra os arquivos `CGnuplot.h` e `CGnuplot.cpp` no editor de texto e veja como o código foi documentado.
- Abra um terminal.
- Vá para o diretório onde está o código.

```
cd /caminho/para/seu/codigo
```

- Peça para o doxygen gerar o arquivo de definições (arquivo que diz para o doxygen como deve ser a documentação).

```
doxygen -g
```

- Peça para o doxygen gerar a documentação.

```
doxygen
```

- Verifique a documentação gerada abrindo o arquivo `html/index.html`.

```
firefox html/index.html
```

ou

```
chrome html/index.html
```

Apresenta-se a seguir algumas imagens com as telas das saídas geradas pelo software doxygen.

**Nota:**

Não perca de vista a visão do todo; do projeto de engenharia como um todo. Cada capítulo, cada seção, cada parágrafo deve se encaixar. Este é um diferencial fundamental do engenheiro em relação ao técnico, a capacidade de desenvolver projetos, de ver o todo e suas diferentes partes, de modelar processos/sistemas/produtos de engenharia.



# Referências Bibliográficas

- [Blaha and Rumbaugh, 2006] Blaha, M. and Rumbaugh, J. (2006). *Modelagem e Projetos Baseados em Objetos com UML 2*. Campus, Rio de Janeiro. 39
- [e Patrick W. Daly, 1995] e Patrick W. Daly, H. K. (1995). *A Guide to Latex 2e*. Addison-Wesley, New York, 2 edition. 74, 76
- [Grossens et al., 1993] Grossens, M., Mittelbach, F., and Samarin, A. (1993). *Latex Companion*. Addison-Wesley, New York. 74, 76
- [Karger, 2004] Karger, A. (2004). *O Tutorial de Lyx*. LyX Team - <http://www.lyx.org>. 74, 76
- [Knuth, 1986] Knuth, D. E. (1986). *The Texbook*. Addison-Wesley. 74, 76
- [Lamport, 1985] Lamport, L. (1985). *Latex - A Document Preparation System*. Addison-Wesley. 74, 76
- [LyX-Team, 2004a] LyX-Team, editor (2004a). *Extended LyX Features*. LyX Team - <http://www.lyx.org>. 74, 76
- [LyX-Team, 2004b] LyX-Team, editor (2004b). *The LyX User's Guide*. LyX Team - <http://www.lyx.org>. 74, 76
- [Rumbaugh et al., 1994] Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., and Lohrensen, W. (1994). *Modelagem e Projetos Baseados em Objetos*. Edit. Campus, Rio de Janeiro. 39
- [Steding-Jessen, 2000] Steding-Jessen, K. (2000). *Latex demo: Exemplo com Latex 2e*. 74, 76

# Apêndice A

## Título do Apêndice

Descreve-se neste apêndice ...

- Os anexos ou apêndices contém material auxiliar. Por exemplo, tabelas, gráficos, resultados de experimentos, algoritmos, códigos e simulações.
- Um apêndice pode incluir assuntos mais gerais (geral demais para estar no núcleo do trabalho) ou mais específicos (detalhado demais para estar no núcleo do trabalho).
- Pode conter um artigo de auxílio fundamental ao trabalho.
- Pode conter artigos publicados.
- [tudo aquilo que for importante para a tese mas não essencial, deve ser colocado em apêndices]
- [como exemplo, revisão de metodologias, técnicas, modelos matemáticos, ítems desenvolvidos por terceiros]
- [algoritmos e programas devem ser colocados no apêndice]
- [imagens detalhadas de programas desenvolvidos devem ser colocados no apêndice]

### A.1 Sub-Título do Apêndice

.....conteúdo..

# Apêndice B

## Usando T<sub>E</sub>X/LA<sub>T</sub>E<sub>X</sub>/L<sub>Y</sub>X

### B.1 Pacotes a serem instalados

#### B.1.1 Windows

Basta ir no site do LyX - <https://www.lyx.org/> e baixar o instalador.

#### B.1.2 GNU/Linux

- Basta abrir um terminal e digitar:

```
dnf install lyx abntex
```

- De forma mais detalhada apresenta-se a seguir um conjunto de pacotes a serem instalados no GNU/Linux.

```
dnf install lyx 'tex(esint10.pfb)' 'tex(lgrcmr.fd)' 'tex(subscript.sty)'
dnf install texlive-prettyref
dnf install texlive-textgreek
dnf install texlive-accents
dnf install texlive-babel-portuges
dnf install texlive-multirow texlive-tablefootnote
dnf install texlive-pmboxdraw
dnf install texlive-sectsty
dnf install texlive-prettyref
dnf install texlive-greek-fontenc texlive-cbfonts
dnf install texlive-luainputenc
dnf install texlive-listings
dnf -y install gnuplot
```

- Dicas:

– Observe que este modelo disponibilizado para vocês não usa mais o formato abnt por default. Mudei para citações no formato apalike por ser semelhante e já vir instalado no LyX.

– Mas é fácil alternar para abnt:

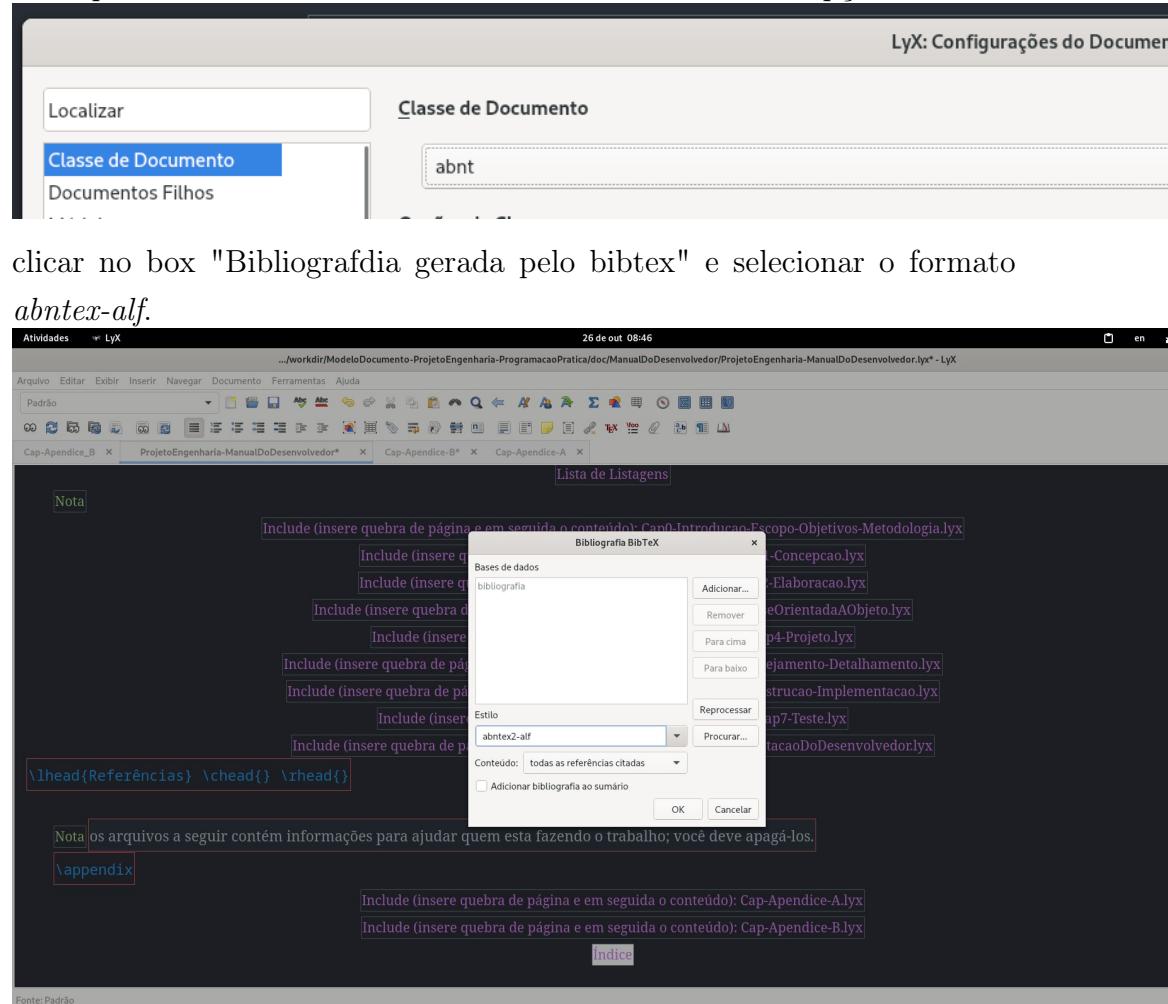
\* basta instalar os pacotes usando os comandos

```
dnf install texlive-abnt texlive-abntex2
```

\* e a seguir ler os manuais da abnt

```
https://www.abntex.net.br/
```

\* no arquivo master selecionar como classe de documento a opção ABNT.



**Nota:** em minha máquina estão instalados os seguintes pacotes principais:

**lyx-fonts-2.3.7-1.fc38.noarch**

**lyx-common-2.3.7-1.fc38.noarch**

**lyx-2.3.7-1.fc38.x86\_64**

**texlive-abntex2-svn49248-65.fc38.noarch**

**texlive-abnt-svn55471-65.fc38.noarch**

**texlive-listings-svn55265-65.fc38.noarch**

**e outras dezenas de pacotes texlive...**

## B.2 Inclusão de Figuras

- Todas as ilustrações, quadros, fotos, esquemas, esboços, desenhos, diagramas, gráficos e demais imagens são incluídos como Figuras; já as Tabelas serão incluídas como tabelas.
- Impressão colorida é caro, logo, quando fizer gráficos, prefira gráficos em preto e branco com uso de marcadores.
- **Lembre-se que se a figura, foto, esquemas, etc, for colorido o mesmo deve obrigatoriamente ser impresso em cores.**
- Veja o que diz o link <http://www.cedsme.ufpr.br/celsoishida/tutorial/artigo/figura.php>
  - "Toda figura deve ser explicada antes de aparecer. É preciso uma descrição detalhada de toda sigla, variável ou parte da figura.
  - Deve-se explicar todos os detalhes para que a figura seja autoexplicativa.
  - Toda figura deve ser referenciada no texto.
  - Foi tirado de alguma fonte? Cite a fonte 'Fonte: (Ishida 2013)'. Se modificou algo utilize 'Fonte: Adaptado de Ishida (2013)'.
  - Prefira traduzir do que deixar a figura em inglês [Sugestão Bueno: deixar conforme original e caso necessário explicar em português]
  - Não deixe tabelas como sendo figuras.
  - Evitar: figura 'abaixo' e 'acima', use referências cruzadas com numerações atualizadas automaticamente.
  - Figuras devem ser chamadas de 'Figura X', com a primeira letra maiúscula e o número da mesma (no Lyx use inserir referência cruzada).
  - Utilizar 'Inserir Label' no Lyx para incluir a referência abaixo da figura e 'Inserir Referência cruzada' para citar a mesma.

Veja a seguir exemplo de inclusão de Figura.

- No LyX 2.x, vá em "Inserir-Flutuante-Figura" (também pode clicar no ícone "Inserir Flutuante Figura").
- Dentro do box "float:figure" ou "flutuante: Figura" você deve incluir o título da Figura - gráfico/foto/mapa/etc.

- a seguir inclua um "label-rótulo", que será utilizado para referência cruzada.
  - No LyX 2.x vá em "Inserir etiqueta" ou pressione o botão "Inserir Legenda".
- acima do título coloque a figura.
  - No LyX 2.x vá em "Inserir Gráficos" ou click no ícone "Inserir Gráficos".
  - Para centralizar a figura, clicar no ícone "Configurações de parágrafo" e selecionar "center"/"centro".
- outra possibilidade é copiar um box de figura existente, e então modificar título, rótulo e figura.

**Dica:** no canto superior esquerdo do box que contém a figura, existe um texto em vermelho com box cinza. Ao clicar com o botão esquerdo do mouse neste box, o mesmo é iconizado. Repete o clic para retornar ao estado anterior. Ao clicar com o botão direito, abre-se um diálogo em que é possível selecionar a opção [ X ] Aqui definitivamente, se quiser obrigar o LyX/TEx-LATEX a colocar a figura aqui.

Apresenta-se na Figura B.1 as etapas fundamentais no processamento digital de imagens, adaptadas ao estudo de lâminas de meios porosos. Dentre as vantagens do uso da análise de imagens para a determinação das propriedades físicas das rochas destacam-se a possibilidade de análise de grande quantidade de amostras a um custo reduzido e o uso de amostras de calha e de testemunhos danificados.

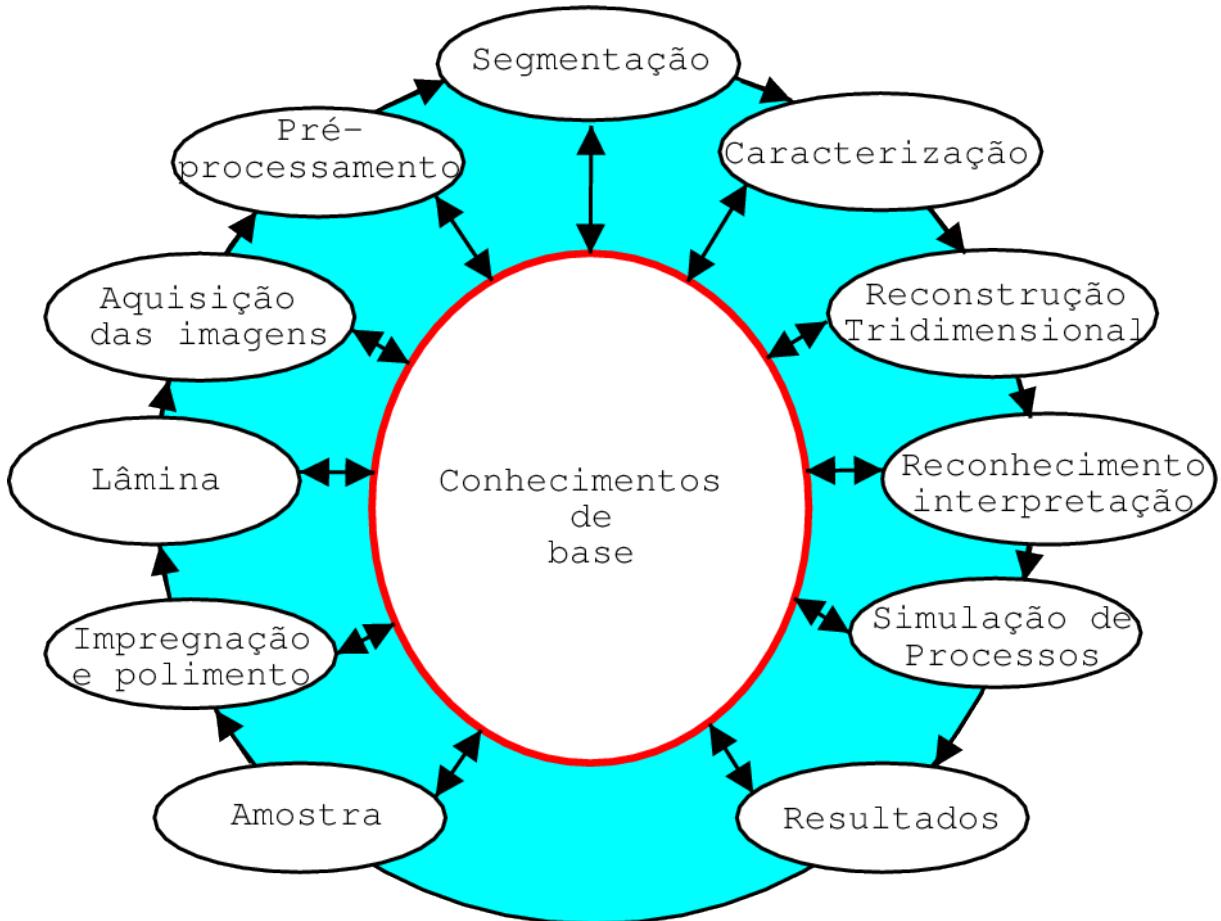
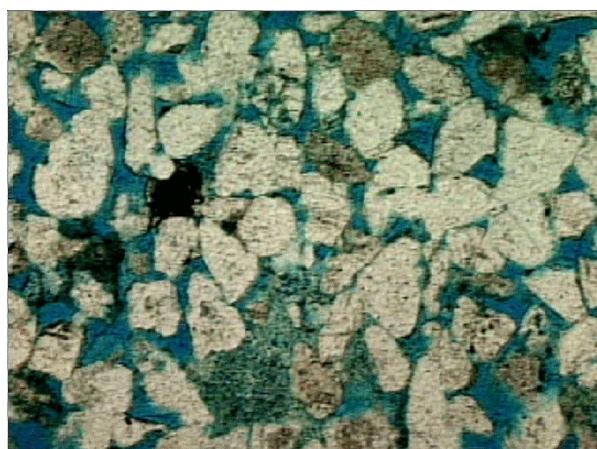


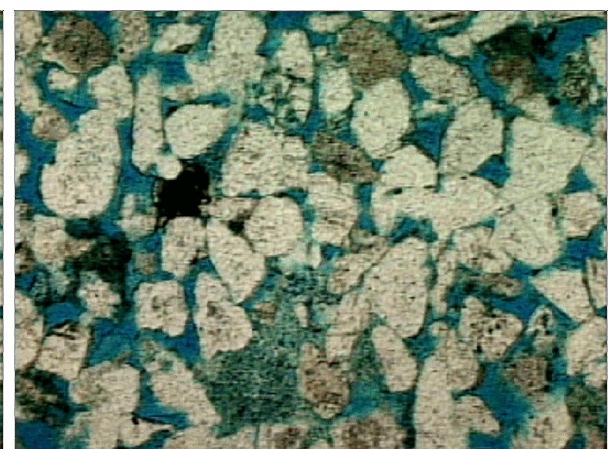
Figura B.1: Etapas fundamentais no processamento digital de imagens, aplicadas ao estudo de lâminas de meios porosos [adaptado de Gonzales e Woods (1992)]

**Veja a seguir exemplo com duas figuras em um box**

A Figura B.2 mostra uma imagem do Berea 200 colorida em (a) e depois da etapa de pré-processamento com o filtro passa baixa em (b). Observe a redução do contraste e um certo embaçamento da imagem.



(a) subtítulo para figura a



(b) subtítulo para figura b

Figura B.2: Título da figura. Em (a) blablabla; em (b) blablabla

**Dica:** digamos que tenha uma figura e não lembre de onde a tirou. Vá em <http://images.google.com/>, cole a figura que o google tenta achar a mesma.

## B.3 Inclusão de Equações

Veja a seguir exemplo de equações numeradas. No Lyx 2.0 vá em "Inserir Math Fórmula Numerada". Se selecionar a equação e pressionar no ícone "Inserir Legenda" ou em "Inserir Etiqueta", pode-se dar um nome para a equação, e este nome pode ser referenciado ao longo do texto.

A caracterização tem como objetivo identificar parâmetros geométricos do material em estudo. Em uma imagem bidimensional já segmentada procura-se determinar a porosidade, a distribuição de tamanho de poros, a função autocorrelação, a função conectividade, o número de objetos, a área dos objetos e fatores de forma. Alguns destes parâmetros são descritos a seguir.

Seja  $\mathbf{x}$  um ponto de coordenada  $(x,y)$  onde  $x,y$  são valores inteiros num espaço discreto bidimensional. Pode-se definir a função de fase  $FF(\mathbf{x})$  pela relação:

$$FF(\mathbf{x}) = \begin{cases} 1 & \text{se } \mathbf{x} \text{ pertence ao espaço poroso} \\ 0 & \text{se } \mathbf{x} \text{ não pertence ao espaço poroso} \end{cases} \quad (\text{B.1})$$

## B.4 Inclusão de Tabelas

Apresenta-se na Tabela B.1 blablabla.

- Para inserir uma tabela o procedimento é praticamente o mesmo da inserção de figuras, a diferença é que você deve selecionar "Inserir Flutuante Tabela".
- Para alterar manualmente a largura de uma coluna, posiciona o mouse sobre a coluna, click com o botão direito, selecione "Configurações", depois entre com a largura no campo largura e pressione ok.
- Quando colocamos o mouse dentro da tabela, aparece uma barra de tarefas na parte de baixo do Lyx, que nos permite manipular alguns aspectos da tabela.
  - O tamanho das colunas é definido pelo texto colocado dentro da coluna. Você pode setar o tamanho da coluna; clicar com botão direito em cima da coluna que quer modificar, a seguir vá em configurações e então sete a largura em cm.

Apresenta-se na Tabela B.2 blablabla. A primeira coluna foi definida com 3cm de largura, e a tabela foi rotacionada (configurações - rodar lado).

Tabela B.1: Exemplo de tabela flutuante.

	Coluna 1	Coluna 2	Coluna 3
Linha 1			
Linha 2			
Linha 3			
Linha 4			
Linha 5			
Linha 6			

## B.5 Inclusão de Listagens de Código

Veja exemplo no Cap6-Ciclos-Construcao-Implementacao.

Para detalhes de uso do pacote `listings` veja o manual no site da *nasa* <https://nasa.github.io/nasa-latex-docs/html/examples/listing.html>.

## B.6 Roteiro Para Uso do Sistema de Citações Com Banco de Dados .bib

O sistema de referências usando bibtex é extremamente simples e muito prático. O mesmo é composto de uma base de dados (um arquivo `.bib` que contém a lista de referencias a ser utilizada). Por exemplo, o arquivo `andre.bib`, inclui referencias bibliograficas no formato `bib` (de uma olhada agora no arquivo `andre.bib` usando um editor de texto como o `emacs`). A seguir, você deve incluir no arquivo do `lyx`, o nome de sua base de dados. Finalmente, você precisa incluir as referencias cruzadas.

Veja a seguir um roteiro:

1. Você deve fazer uma copia do arquivo `bibliografia.bib` e renomear para `seuNome.bib`.  
A seguir deve usar um editor qualquer (mas preferencialmente o `emacs` ou `kate`) para incluir suas referências bibliográficas. Ou seja, inclua no arquivo `seuNome.bib` todas as citações e referências bibliográficas a serem incluídas em seu TCC/dissertação/-tese (tudo que você leu, e que pode ser incluído na citação da tese e de outros artigos. É sua base de dados de citações).
  - (a) Você pode incluir ítems no arquivo `.bib` que não irão fazer parte da tese, mas poderão ser citadas em artigos futuros.
    - i. ou seja, seu arquivo `.bib` deve ter todas as referencias que irá citar ao longo da vida acadêmica.
    - ii. se ficar muito grande pode ser dividido por área (`seuNome-area.bib`).
  - (b) Cada item adicionado ao arquivo `.bib` tem uma chave, o nome desta chave não pode ter espaço nem acentos.

No exemplo abaixo a chave é **adb-mestrado-1994**:

```

@mastersthesis{adb-mestrado-1994,
  author = {André Duarte Bueno},
  title = {Transferência de Calor e Umidade em Telhas: Simulação e Análise},
  school = {Programa de Pós-Graduação em Engenharia Civil da Universidade de São Paulo},
  year = {1994},
  type = {Tese de Mestrado},
  address = {Florianópolis},
  month = {agosto},
  key = {Transferência de Calor, Telhas, Meios Porosos, Transporte de Calor}
}

```

- (c) A codificação de caracteres definida para o documento deve ser a mesma usada pelo arquivo .bib. Se seu documento do LyX usa ISO-8859-1, o arquivo .bib deve usar a mesma codificação. No editor *Kate* vá em "Salvar como com codificação..." e localize a opção ISO-8859-1.
- i. Em resumo. Todos os textos a serem incluídos no lyx, como arquivos txt, arquivos de código (\*.h. \*.cpp), scripts, como arquivos do *gnuplot*, etc, devem usar a mesma codificação de caracteres. Normalmente usamos ISO-8859-1 que é o latin 1.
2. Para fazer uma citação é necessário, antes, incluir no arquivo mestre do lyx um "Insert-> Lists & Toc->Bibtex reference". Vai aparecer um diálogo pedindo para você incluir o nome do arquivo com a base de dados de citações (digite seuNome.bib).
- Nota:** no modelo que vocês recebem isto já foi feito e aponta para o arquivo com a base de dados "bibliografia.bib".
3. Finalmente, faça referencias cruzadas usando o ítem de menu "Insert Cross-Reference".
  4. Aqui um exemplo, vou citar material sobre LyX e Latex.
    - (a) Veja maiores informações sobre latex em [Grossens et al., 1993, Knuth, 1986, Steding-Jessen, 2000, e Patrick W. Daly, 1995, Lamport, 1985, LyX-Team, 2004a, Karger, 2004, LyX-Team, 2004b].
  5. Dica: em alguns sites é possível encontrar os dados .bib da citação;
    - (a) Exemplo:
      - i. Vá no link [https://www.google.com.br/books/edition/C++\\_For\\_Dummies/-hMTzvUybLoC?hl=pt-BR&gbpv=0](https://www.google.com.br/books/edition/C++_For_Dummies/-hMTzvUybLoC?hl=pt-BR&gbpv=0)

**C++ For Dummies**

Por [Stephen R. Davis](#) - 2009

[Visualização](#) [Pesquisar](#) [+ Adicionar à minha biblioteca](#)

[Visão geral](#) [Comprar o livro](#) [Outras edições](#) [Mais desse autor](#)

**Sobre esta edição**

ISBN:	9780470524121, 047052412X	Número de páginas:	432
Publicação:	20 de abril de 2009	Formato:	Livro digital
Editora:	<a href="#">Wiley</a>	Idioma:	Inglês
Autor:	<a href="#">Stephen R. Davis</a>		

[Criar citação](#) [Índice](#)

**Sobre a obra**

Data da primeira publicação: 1994

**Autor(a)**

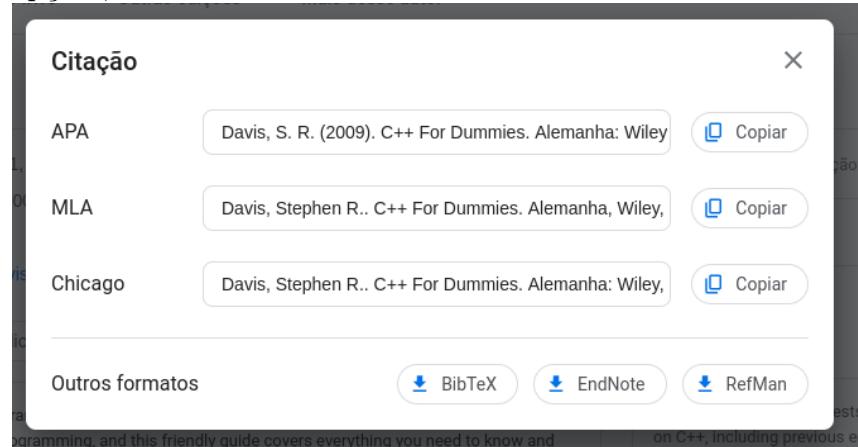
**Stephen R. Davis**  
Autor

Stephen R. Davis is the bestselling author of several books on C++, including previous editions of C++ For Dummies. He has been programming for more than 30 years and presently works for L-3 Communications in the area of Homeland Defense.

[Pesquisar Stephen R. Davis](#)

**Mais desse autor**

- ii. A seguir pressione no botão "Criar Citação" ele abra um diálogo com as opções, selecione "outros formatos BibTex".



- iii. Vai baixar para seu computador a referencia. Ou seja, é tão bacana que em alguns casos nem precisa digitar a citação!

```

Novo Abrir Salvar Salvar como Desfazer Refazer
home > andreduartebueno > Downloads > C++_For_Dummies.bibtex
[book{davis2009c++},
 title={C++ For Dummies},
 author={Davis, S.R.},
 isbn={9780470524121},
 series={--For dummies},
 url={https://books.google.com.br/books?id=-hMTzvUybLoC},
 year={2009},
 publisher={Wiley}]

```

- iv. Note que também é possível editar diretamente este texto no editor, mas dá mais trabalho.

### B.6.1 Citações no meio do texto

Segundo [Grossens et al., 1993] asldkjasldkajsdlkajsdlsksjd [LyX-Team, 2004a, Karger, 2004] Segundo [Grossens et al., 1993, Knuth, 1986, Steding-Jessen, 2000, e Patrick W. Daly, 1995, Lamport, 1985, LyX-Team, 2004a, Karger, 2004, LyX-Team, 2004b] asldkjasldkajsdlkajsdlsksjd

### B.6.2 Citações no início do texto

```
\citeonline{SL-latex-companion} 1asldkjasldkajsdlkajsdlsksjd  
\citeonline{SL-latex-companion,SL-latex-demo} 2asldkjasldkajsdlkajsdlsksjd
```

### B.6.3 Citações tipo apud

Segundo \apud{SL-latex-companion}{SL-latex-demo} 3asldkjasldkajsdlkajsdlsksjd  
\apudsonline{SL-latex-companion}{SL-latex-demo} 4asldkjasldkajsdlkajsdlsksjd....

### B.6.4 Incluir nas referências bibliográficas (fim do documento), mas não citar

asldkjasldkajsdlkajsdlsksjd  
asldkjasldkajsdlkajsdlsksjd

Diversos sites de pesquisa e mesmo sites de pesquisa como o google, costumam ter informações sobre artigos e livros no formato .bib.

Veja a seguir Figura do site books.google.com.br. O usuário localizou o livro na base de dados do google, rolou a página para baixo e encontrou o link para as referencias nos formatos: BIBTeX, EndNote e RefMan. Baixe a versão BIBTeX e copie seu conteúdo para o arquivo bibliografia.bib.

### Informações bibliográficas

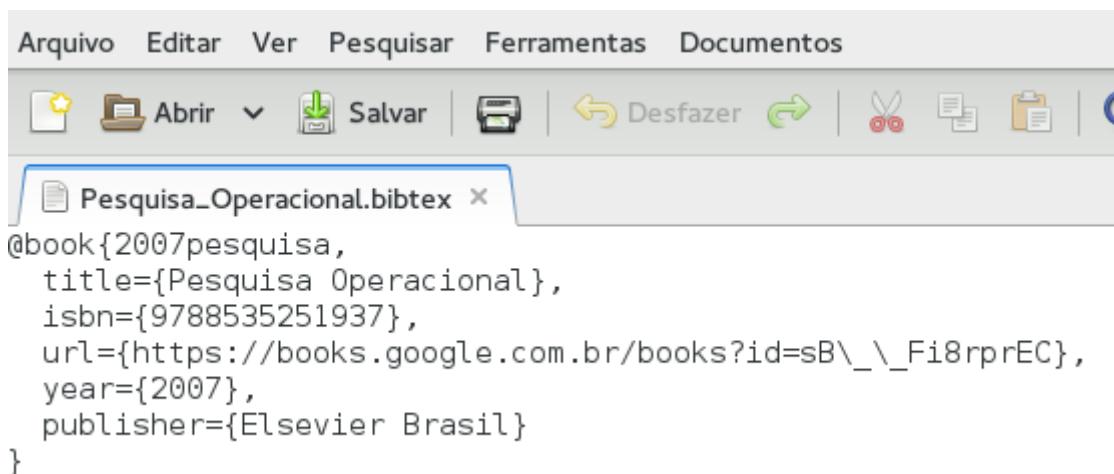
Título	Pesquisa Operacional
Autor	<a href="#">Arenales,marcos/armentano,vinícius/morabit</a>
Editora	Elsevier Brasil, 2007
ISBN	8535251936, 9788535251937
Num. págs.	523 páginas

Exportar citação

[BIBTeX](#)

[EndNote](#)

[RefMan](#)



The screenshot shows a software interface for managing bibliographic references. The menu bar includes 'Arquivo', 'Editar', 'Ver', 'Pesquisar', 'Ferramentas', and 'Documentos'. The toolbar contains icons for opening, saving, undoing, redoing, cutting, copying, and pasting. A window titled 'Pesquisa\_Operacional.bibtex' is open, displaying the following BibTeX code:

```
@book{2007pesquisa,
    title={Pesquisa Operacional},
    isbn={9788535251937},
    url={https://books.google.com.br/books?id=sB\_\_Fi8rprEC},
    year={2007},
    publisher={Elsevier Brasil}
}
```

## B.7 Informações Adicionais

- Manuais do LyX (precisa ler! comece lendo o tutorial)
- <http://chem-e.org/comando-cite-e-citeonline-no-abntex/>
- <http://win.ua.ac.be/~nschloe/content/bibtex-how-cite-website>.
- <http://chem-e.org/comando-apud-e-apudonline-no-abntex/>.
- [http://en.wikibooks.org/wiki/LaTeX/Bibliography\\_Management](http://en.wikibooks.org/wiki/LaTeX/Bibliography_Management)

Tabela B.2: Exemplo de tabela flutuante - coluna com dimensão fixa e rotacionada.

# Índice Remissivo

- Análise orientada a objeto, 27  
AOO, 27  
Associações, 45  
atributos, 44
- Casos de uso, 10  
Ciclo construção, 54  
Ciclos de Planejamento/Detalhamento, 52  
Citações, 73  
Citações no início do texto, 76  
Citações no meio do texto, 76  
Citações tipo apud, 76  
colaboração, 36  
comunicação, 36  
Concepção, 5  
Controle, 41
- Diagrama de colaboração, 36  
Diagrama de componentes, 47  
Diagrama de estado, 37  
Diagrama de execução, 48  
Diagrama de implantação, 48  
Diagrama de sequência, 34
- Efeitos do projeto nas associações, 45  
Efeitos do projeto nas heranças, 45  
Efeitos do projeto nos métodos, 44  
Elaboração, 17  
Especificação, 5  
especificação, 5  
estado, 37  
Eventos, 34
- Heranças, 45  
heranças, 45
- Implementação, 54  
Inclusão de equações, 72  
Inclusão de figuras, 69  
Inclusão de tabelas, 72
- métodos, 44  
Mensagens, 34  
Metodologia utilizada, 3  
modelo, 43, 44  
otimizações, 46
- Plataformas, 41  
POO, 43  
Projeto do sistema, 39  
Projeto orientado a objeto, 43  
Protocolos, 39
- Recursos, 40  
Requisitos, 7  
Requisitos funcionais, 7  
Requisitos não funcionais, 10