

UNIVERSIDADE ESTADUAL DO NORTE FLUMINENSE  
LABORATÓRIO DE ENGENHARIA E EXPLORAÇÃO DE PETRÓLEO  
CENTRO DE CIÊNCIA E TECNOLOGIA

PROJETO DE ENGENHARIA  
DESENVOLVIMENTO DO SOFTWARE  
FERRAMENTA PARA GERAÇÃO DA CURVA DE FLUXO FRACIONÁRIO  
DISCIPLINA LEP-0144: Projeto de Software Aplicado à Engenharia  
Setor de Modelagem Matemática Computacional

Versão 1:  
Fabiane da Silva Barros  
Prof. André Duarte Bueno

MACAÉ - RJ  
Novembro - 2025

# Sumário

<b>1</b>	<b>Concepção</b>	<b>6</b>
1.1	Metodologia . . . . .	6
1.2	Nome do Sistema/Produto . . . . .	6
1.3	Especificação . . . . .	6
1.4	Requisitos . . . . .	8
1.4.1	Requisitos funcionais . . . . .	8
1.4.2	Requisitos não funcionais . . . . .	8
1.5	Casos de Uso . . . . .	9
1.5.1	Diagrama de caso de uso geral . . . . .	9
1.5.2	Diagrama de caso de uso específico . . . . .	9
<b>2</b>	<b>Elaboração</b>	<b>11</b>
2.1	Análise de domínio . . . . .	11
2.2	Formulação teórica . . . . .	12
2.2.1	Implementação do Modelo Tabular (Interpolação Linear) . . . . .	12
2.2.2	Implementação do Modelo Analítico (Correlações de Corey) . . . . .	12
2.3	Identificação de pacotes – assuntos . . . . .	13
2.4	Diagrama de pacotes – assuntos . . . . .	14
<b>3</b>	<b>AOO – Análise Orientada a Objeto</b>	<b>15</b>
3.1	Diagramas de classes . . . . .	15
3.1.1	Dicionário de classes . . . . .	15
3.2	Diagrama de sequência – eventos e mensagens . . . . .	16
3.2.1	Diagrama de sequência geral . . . . .	16
3.2.2	Diagrama de sequência específico . . . . .	16
3.3	Diagrama de comunicação – colaboração . . . . .	17
3.4	Diagrama de estado . . . . .	18
3.5	Diagrama de atividades . . . . .	19
<b>A</b>	<b>Formato do Arquivo de Entrada</b>	<b>24</b>
A.1	Exemplo de Arquivo de Entrada (Modelo Tabelado) . . . . .	24
A.2	Exemplo de Arquivo de Entrada (Modelo Corey) . . . . .	25

<b>B Guia de Compilação e Execução</b>	<b>26</b>
B.1 Dependências . . . . .	26
B.2 Compilação (Exemplo g++) . . . . .	26
B.3 Execução . . . . .	26

# Lista de Figuras

1.1	Metodologia utilizada no desenvolvimento do sistema . . . . .	7
1.2	Diagrama de caso de uso – Caso de uso geral . . . . .	10
1.3	Diagrama de caso de uso específico – Título . . . . .	10
2.1	Diagrama de Pacotes . . . . .	14
3.2	Diagrama de sequência . . . . .	16
3.3	Diagrama de sequência . . . . .	17
3.4	Diagrama de comunicação . . . . .	18
3.5	Diagrama de máquina de estado . . . . .	19
3.6	Diagrama de atividades . . . . .	20
3.1	Diagrama de classes . . . . .	21

# Lista de Tabelas

1.1	Caso de uso 1 . . . . .	9
-----	-------------------------	---

# Capítulo 1

## Concepção

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

### 1.1 Metodologia

A Figura 1.1 apresenta a metodologia a ser utilizada no desenvolvimento do sistema.

### 1.2 Nome do Sistema/Produto

Nome	Ferramenta para Geração da Curva de Fluxo Fracionário
<b>Componentes principais</b>	1. Módulo de Entrada (Leitor de arquivo de texto) 2. Módulo de Dados (Modelos de Permeabilidade) 3. Módulo de Cálculo (Equação de Buckley-Leverett) 4. Módulo de Saída (Gerador de .csv e interface com Gnuplot)
<b>Missão</b>	Missão, Calcular e visualizar a curva de fluxo fracionário ( $f_w$ vs. $S_w$ ) para auxiliar na análise de deslocamento bifásico em meios porosos.

### 1.3 Especificação

A especificação desta solução de engenharia define uma ferramenta computacional analítica, de caráter didático e profissional, implementada como um executável de linha de comando. O sistema irá abstrair a complexidade do modelo matemático de fluxo fracionário, exigindo do usuário apenas um

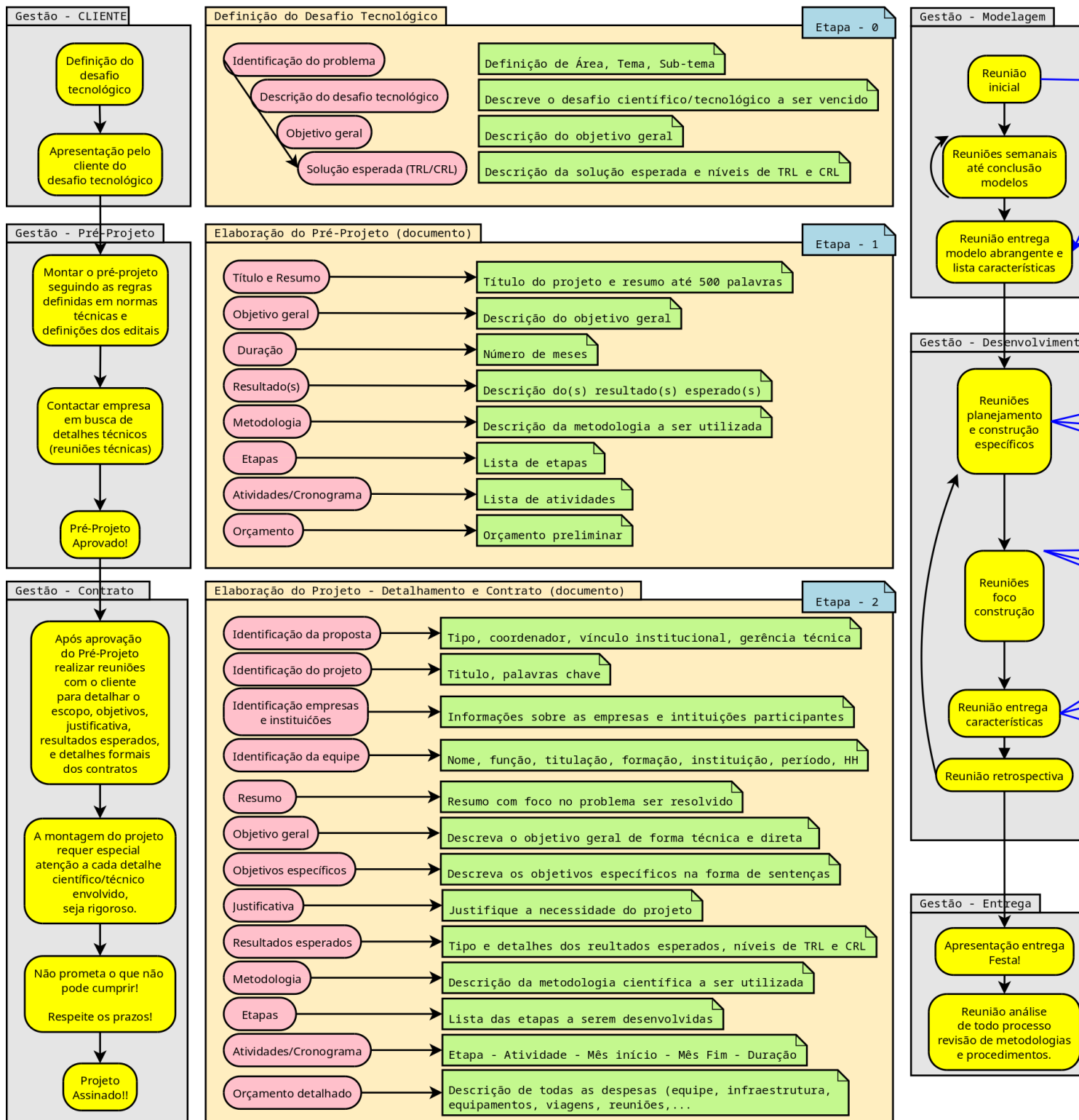


Figura 1.1: Metodologia utilizada no desenvolvimento do sistema

arquivo de configuração em texto simples. Este arquivo deverá conter os parâmetros reológicos dos fluidos (viscosidades) e os dados de permeabilidade relativa, que são suportados em dois formatos: (a) dados tabulares discretos ( $S_w, k_{rw}, k_{ro}$ ), que serão processados via interpolação linear; ou (b) parâmetros para o modelo analítico de Corey [Corey, 1954]). O software processará esses dados, calculará a curva  $f_w$  vs  $S_w$  e gerará dois produtos: um arquivo de dados .csv para análise externa e uma visualização gráfica imediata, invocando o Gnuplot para plotar a curva "S" característica.

## 1.4 Requisitos

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

### 1.4.1 Requisitos funcionais

Apresenta-se a seguir os requisitos funcionais.

<b>RF-01</b>	O sistema deve ler um arquivo de entrada de texto contendo os parâmetros da simulação (viscosidades e dados de permeabilidade).
<b>RF-02</b>	O sistema deve permitir que os dados de permeabilidade relativa sejam fornecidos de duas formas: (a) uma tabela de pontos ( $S_w, k_{rw}, k_{ro}$ ) ou (b) parâmetros para um modelo analítico (Correlações de Corey[Corey, 1954]).
<b>RF-03</b>	O sistema deve calcular os valores de fluxo fracionário ( $f_w$ ) para um intervalo de saturação de água ( $S_w$ ) de 0 a 1.
<b>RF-04</b>	O sistema deve gerar um arquivo de saída em formato de texto (ex: .csv) contendo os pares de dados ( $S_w, f_w$ ) calculados.
<b>RF-05</b>	O sistema deve invocar um processo externo (Gnuplot) para plotar automaticamente o gráfico de $f_w$ vs. $S_w$ .
<b>RF-06</b>	O sistema deve informar ao usuário em caso de erro na leitura do arquivo de entrada (ex: arquivo não encontrado, formato inválido).

### 1.4.2 Requisitos não funcionais

<b>RNF-01</b>	Os cálculos de permeabilidade relativa, quando tabulados, devem usar interpolação linear.
<b>RNF-02</b>	O programa deverá ser multi-plataforma, podendo ser executado em Windows e GNU/Linux.



<b>RNF-03</b>	O software deve ser desenvolvido na linguagem C++ utilizando o paradigma da Orientação a Objetos.
<b>RNF-04</b>	O software deve ser executável via terminal (linha de comando).

## 1.5 Casos de Uso

A Tabela 1.1 mostra a descrição de um caso de uso.

Tabela 1.1: Caso de uso 1

Nome do caso de uso:	Executar Cálculo e Analisar Resultados
Resumo/descrição:	O usuário executa a ferramenta, que lê os dados de entrada, calcula a curva de fluxo fracionário e exibe o gráfico resultante.
Etapas:	<ol style="list-style-type: none"><li>1. Usuário executa o programa via terminal, fornecendo o nome do arquivo de entrada.</li><li>2. O sistema lê e valida o arquivo de entrada.</li><li>3. O sistema calcula a curva de <math>f_w</math> vs. <math>S_w</math>.</li><li>4. O sistema salva os resultados em um arquivo .csv.</li><li>5. O sistema invoca o Gnuplot para exibir o gráfico.</li><li>6. O usuário analisa o gráfico e fecha a janela.</li></ol>
Cenários alternativos:	<p>2a. O arquivo de entrada não existe ou está mal formatado: O sistema exibe uma mensagem de erro clara e encerra.</p> <p>5a. O Gnuplot não está instalado ou não foi encontrado no PATH do sistema: O sistema exibe um aviso, mas informa que o arquivo de dados .csv foi gerado com sucesso.</p>

### 1.5.1 Diagrama de caso de uso geral

O diagrama de caso de uso geral da Figura 1.2 mostra o ator principal, o Usuário (Engenheiro/Estudante), e as interações primárias que ele pode ter com o sistema.

### 1.5.2 Diagrama de caso de uso específico

O caso de uso Executar Cálculo é detalhado na Figura 3.2.2. Ele é composto por várias sub tarefas que são obrigatoriamente incluídas em sua execução.



Figura 1.2: Diagrama de caso de uso – Caso de uso geral

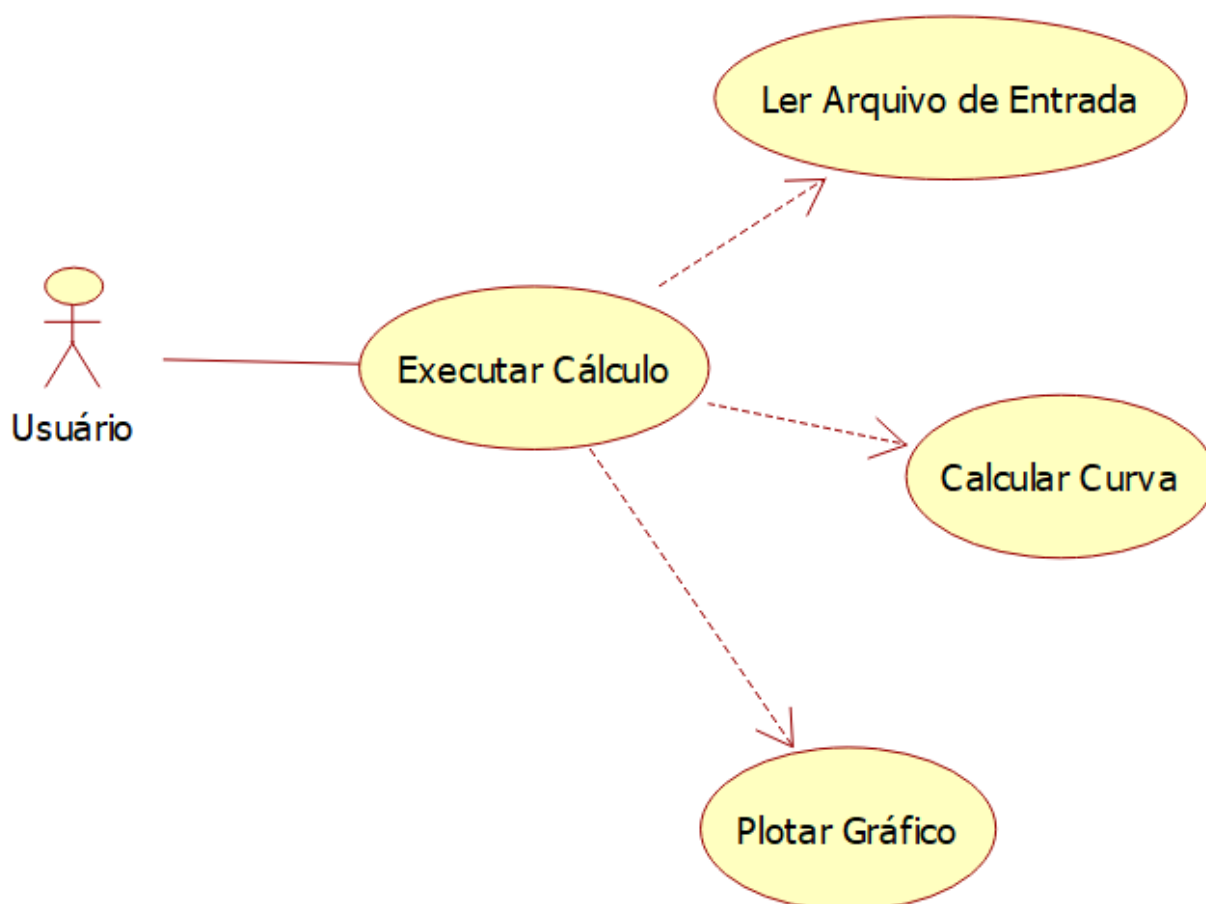


Figura 1.3: Diagrama de caso de uso específico – Título

# Capítulo 2

## Elaboração

Este capítulo detalha o domínio do problema e a lógica de engenharia.

### 2.1 Análise de domínio

O domínio deste projeto situa-se na interseção da Engenharia de Reservatórios com a Modelagem Computacional. A seguir, detalha-se a análise deste domínio e as decisões de arquitetura de software que ela motivou.

- **Justificativa da Plataforma Tecnológica:** A linguagem C++ (padrão C++17) foi selecionada como base tecnológica. A decisão se justifica pelo seu comprovado desempenho computacional, controle de baixo nível de memória e vasta biblioteca padrão (STL). Essa escolha garante que o núcleo de cálculo seja eficiente e, futuramente, possa ser facilmente integrado como uma biblioteca em simuladores de reservatório mais complexos. O desenvolvimento visa a portabilidade entre Windows e GNU/Linux, utilizando ferramentas de compilação padrão (g++, clang).
- **Arquitetura de Dados:** Optou-se por uma arquitetura serverless e sem banco de dados. A persistência dos dados de entrada é gerenciada pelo usuário via arquivos de texto .txt com formato baseado em palavras-chave. Esta decisão mitiga a complexidade de deploy e configuração, alinhando-se à natureza de uma ferramenta de análise rápida. O protocolo de saída (.csv) foi escolhido por sua universalidade, permitindo a interoperabilidade com qualquer software de planilha ou script de pós-processamento.
- **Gerenciamento de Dependências:** O projeto terá uma única dependência externa principal: o Gnuplot. Para manter o software leve e evitar a complexidade de linkagem de bibliotecas gráficas, o Gnuplot será invocado como um processo de sistema externo (system()). A classe Gnuplot do nosso projeto atuará como uma Facade, encapsulando essa chamada.
- **Modelo de Controle:** O fluxo de execução do software é puramente sequencial e síncrono (Leitura -> Cálculo -> Plotagem). Dada a baixa complexidade computacional do modelo 1D, não há necessidade de implementação de concorrência ou processamento paralelo, o que simplifica o design.

## 2.2 Formulação teórica

O núcleo deste projeto é a implementação da equação de fluxo fracionário ( $f_w$ ) de Buckley-Leverett[Buckley conforme definido no documento do Desafio Tecnológico :

$$f_w = \frac{1}{1 + \frac{k_{ro}(S_w)}{\mu_o} \cdot \frac{\mu_w}{k_{rw}(S_w)}}$$

O desafio computacional reside em como obter os valores de  $k_{ro}(S_w)$  e  $k_{rw}(S_w)$  para qualquer valor de saturação de água  $S_w$ , uma vez que os dados de entrada podem vir de duas formas distintas: um modelo tabular ou um modelo analítico.

### 2.2.1 Implementação do Modelo Tabular (Interpolação Linear)

Quando os dados são fornecidos como uma tabela de pontos (lidos do arquivo de texto), o programa precisa calcular os valores de  $k_r$  para saturações que estão entre os pontos da tabela. Para isso, usaremos a Interpolação Linear.

- Algoritmo:

1. O software lerá o arquivo de entrada e armazenará os  $N$  pontos de dados em três vetores (ou listas) paralelos: `vec_Sw[]`, `vec_Krw[]`, `vec_Kro[]`.
2. Quando a função `calcularFw(Sw_desejada)` for chamada:
3. O programa irá buscar no `vec_Sw[]` os dois valores,  $S_{w,i}$  e  $S_{w,i+1}$ , que "cercam" o valor de  $S_w$  desejado (ou seja,  $S_{w,i} \leq S_w \leq S_{w,i+1}$ ).
4. A interpolação linear é aplicada para encontrar o  $k_{rw}$  e  $k_{ro}$  correspondentes:

- $k_{rw} = k_{rw,i} + (S_{w,desejada} - S_{w,i}) \times \frac{k_{rw,i+1} - k_{rw,i}}{S_{w,i+1} - S_{w,i}}$
- $k_{ro} = k_{ro,i} + (S_{w,desejada} - S_{w,i}) \times \frac{k_{ro,i+1} - k_{ro,i}}{S_{w,i+1} - S_{w,i}}$

5. Estes valores interpolados de  $k_{rw}$  e  $k_{ro}$  são então usados na equação principal de  $f_w$ .
6. Tratamento de Pontas: Se  $S_{w,desejada}$  for menor que o primeiro valor da tabela, os valores do primeiro ponto ( $k_{rw,0}, k_{ro,0}$ ) serão usados. Se for maior que o último ponto, os valores do último ponto ( $k_{rw,N}, k_{ro,N}$ ) serão usados (sem extrapolação).

### 2.2.2 Implementação do Modelo Analítico (Correlações de Corey)

Quando o arquivo de entrada especifica o uso de um modelo analítico como o de Corey [Corey, 1954], o programa não usará uma tabela, mas sim um conjunto de parâmetros para calcular  $k_r$  diretamente.

- Algoritmo:

1. O software lerá do arquivo de entrada os parâmetros do modelo de Corey:

- $S_{wir}$ : Saturação de água irreduzível.
- $S_{orw}$ : Saturação de óleo residual à injeção de água.
- $k_{rw,max}$  (ou  $k_{rwro}$ ): Permeabilidade relativa à água na saturação de óleo residual.
- $k_{ro,max}$  (ou  $k_{roocw}$ ): Permeabilidade relativa ao óleo na saturação de água irreduzível.
- $n_w$ : Expoente de Corey para a fase água.
- $n_o$ : Expoente de Corey para a fase óleo.

2. Primeiro, calcula-se a saturação de água normalizada ( $S_{w,norm}$ ) para qualquer  $S_w$  de entrada:

$$\bullet S_{w,norm} = \frac{S_w - S_{wir}}{1 - S_{wir} - S_{orw}}$$

3. A  $S_{w,norm}$  é limitada entre 0 e 1 (se  $S_w < S_{wir}$ ,  $S_{w,norm} = 0$ ; se  $S_w > (1 - S_{orw})$ ,  $S_{w,norm} = 1$ ).

4. As permeabilidades relativas são então calculadas usando  $S_{w,norm}$ :

- $k_{rw}(S_w) = k_{rw,max} \times (S_{w,norm})^{n_w}$
- $k_{ro}(S_w) = k_{ro,max} \times (1 - S_{w,norm})^{n_o}$

5. Estes valores calculados de  $k_{rw}$  e  $k_{ro}$  são então usados na equação principal de  $f_w$ .

A estrutura do software em C++ será projetada de forma que a calculadora principal de  $f_w$  não precise saber qual desses métodos está sendo usado, apenas que ela pode pedir os valores de  $k_r$ .

Para uma dedução mais detalhada destas equações e uma discussão aprofundada sobre as propriedades dos fluidos e modelos de permeabilidade, consulte as referências clássicas da engenharia de reservatórios, como Ahmed [Ahmed, 2010] e Lake [Lake et al., 2010].

## 2.3 Identificação de pacotes – assuntos

Identificamos os seguintes pacotes (assuntos) lógicos para a organização do sistema:

- Pacote Core (Núcleo): Contém a classe principal com a lógica de engenharia, ou seja, a CalculadoraFluxoFracionario. Este pacote implementa a equação de Buckley-Leverett.
- Pacote ModelosKr: Contém a interface ICurvasPermeabilidade e suas implementações concretas (CurvasPermeabilidadeTabelada e CurvasPermeabilidadeCorey). Este pacote é responsável por fornecer os dados de permeabilidade relativa para o pacote Core.
- Pacote IO (Entrada/Saída): Contém classes utilitárias responsáveis pela leitura e parsing do arquivo de entrada de texto e pela escrita do arquivo de saída .csv.
- Pacote Visualizacao: Contém a classe utilitária Gnuplot, responsável por formatar os dados de saída e invocar o processo externo do Gnuplot para plotagem.
- Pacote Aplicacao: Contém a classe Simulador (com a função main()), que orquestra todo o processo: lê os dados usando IO, instancia o ModelosKr correto, instancia o Core e, por fim, chama a Visualizacao.

## 2.4 Diagrama de pacotes – assuntos

O Diagrama de Pacotes, apresentado na Figura 2.1, ilustra as dependências entre os assuntos lógicos identificados.

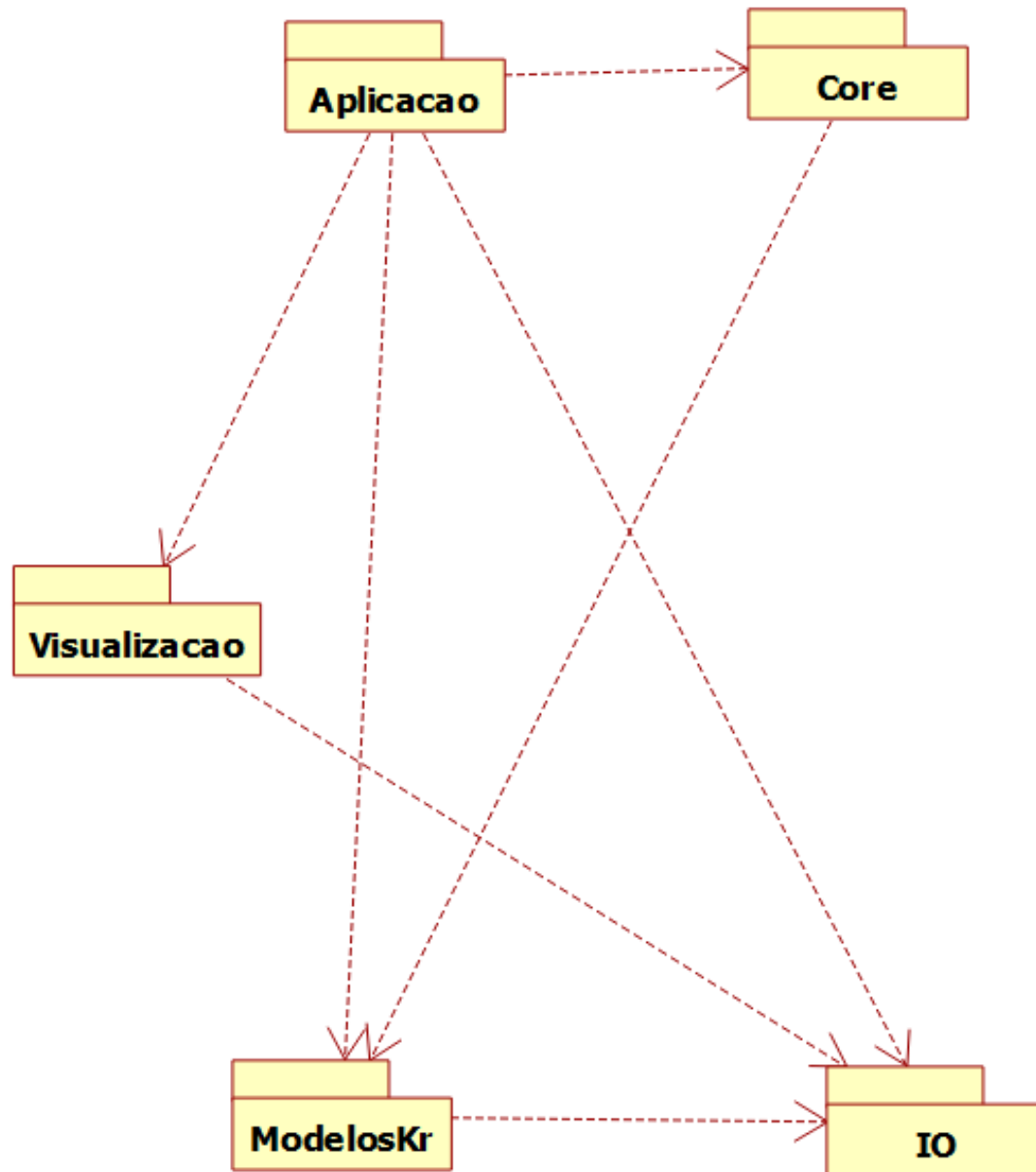


Figura 2.1: Diagrama de Pacotes

# Capítulo 3

## AOO – Análise Orientada a Objeto

### 3.1 Diagramas de classes

O design estático do sistema, apresentado na Figura 3.1, é fundamentado em princípios S.O.L.I.D. de engenharia de software para garantir um código limpo, extensível e de fácil manutenção.

1. Princípio da Inversão de Dependência (D do S.O.L.I.D.): Este é o pilar central da arquitetura. O módulo de alto nível (CalculadoraFluxoFracionario) não depende diretamente dos módulos de baixo nível (...Tabelada, ...Corey). Ambos dependem de uma abstração: a interface ICurvasPermeabilidade.
2. Padrão de Projeto Strategy (Estratégia): A interface ICurvasPermeabilidade implementa o padrão Strategy. Ela define um "contrato" para uma família de algoritmos (os modelos de  $k_r$ ) e permite que eles sejam trocados dinamicamente. A Calculadora é configurada com uma dessas estratégias no momento da sua criação, através de Injeção de Dependência em seu construtor. Isso torna a calculadora completamente agnóstica aos detalhes de como o  $k_r$  é obtido.
3. Padrão de Projeto Facade (Fachada): A classe Gnuplot é uma Facade estática. Ela fornece uma interface unificada e simples (plotarCurva()) para um subsistema complexo (formatação de scripts, salvamento de arquivos temporários e chamada de processos do sistema operacional).

#### 3.1.1 Dicionário de classes

- Classe ICurvasPermeabilidade: (Interface) Define o contrato de serviço para os modelos de permeabilidade relativa. Garante o polimorfismo e a Inversão de Dependência.
- Classe CurvasPermeabilidadeTabelada: (Implementação Concreta) Implementação da Strategy de  $k_r$  via dados tabulares. Encapsula a lógica de interpolação linear.
- Classe CurvasPermeabilidadeCorey: (Implementação Concreta) Implementação da Strategy de  $k_r$  via correlação analítica de Corey [Corey, 1954]. Encapsula a lógica matemática do modelo.

- Classe *CalculadoraFluxoFracionario*: (Domínio de Negócio) Classe principal que implementa o domínio central de negócio (a equação de Buckley-Leverett). Opera em um alto nível de abstração, dependendo apenas da interface *ICurvasPermeabilidade*.
- Classe *Gnuplot*: (Utilitário / Facade) Classe estática que encapsula toda a lógica de interação com o Gnuplot, simplificando a plotagem para o resto da aplicação.
- Classe *Simulador*: (Orquestrador / Ponto de Entrada) Atua como o orquestrador da aplicação. É responsável por "montar" o sistema (ler arquivos, instanciar os objetos corretos usando *new*) e injetar as dependências (*modeloKr* dentro da *Calculadora*). Contém o ponto de entrada *main()*.

## 3.2 Diagrama de sequência – eventos e mensagens

### 3.2.1 Diagrama de sequência geral

O diagrama de sequência geral, Figura 3.2, ilustra o fluxo principal de execução do software, correspondente ao caso de uso "Executar Cálculo e Analisar Resultados" com sucesso. .

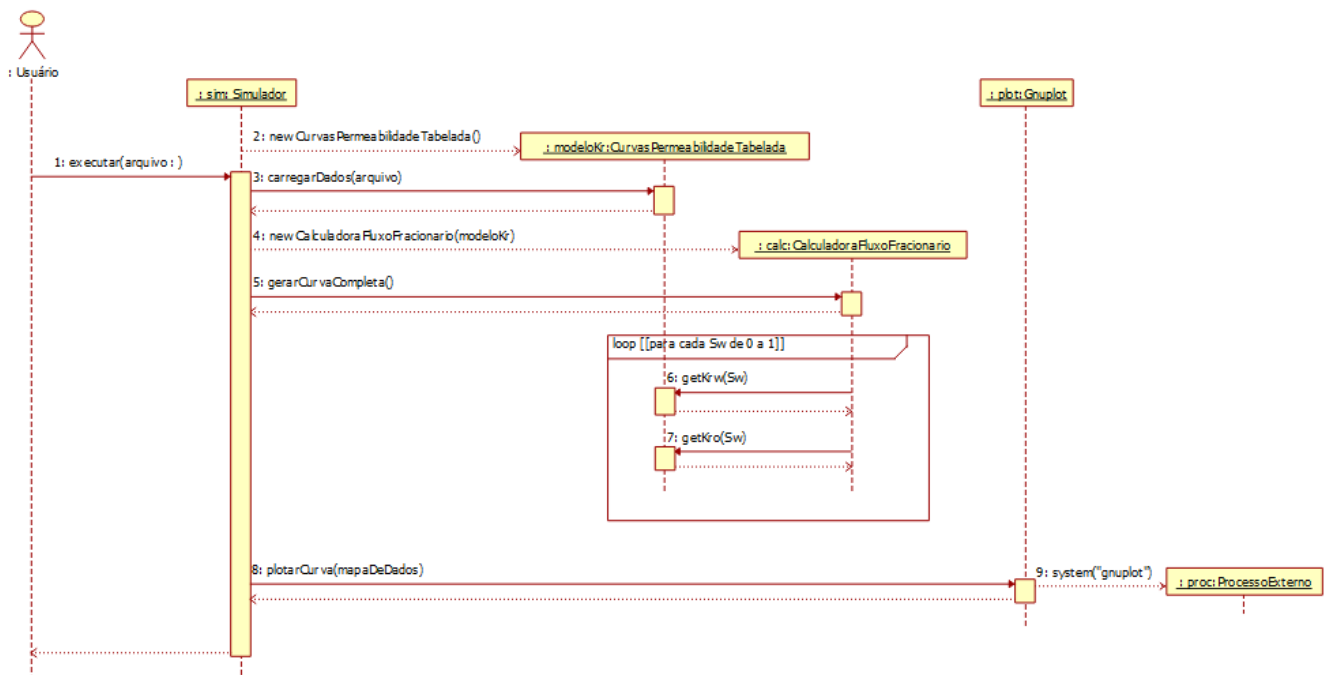


Figura 3.2: Diagrama de sequência

### 3.2.2 Diagrama de sequência específico

O diagrama da Figura 3.3 detalha a atividade específica de *getKrw()* na classe *CurvasPermeabilidadeTabelada*, mostrando a lógica de interpolação.



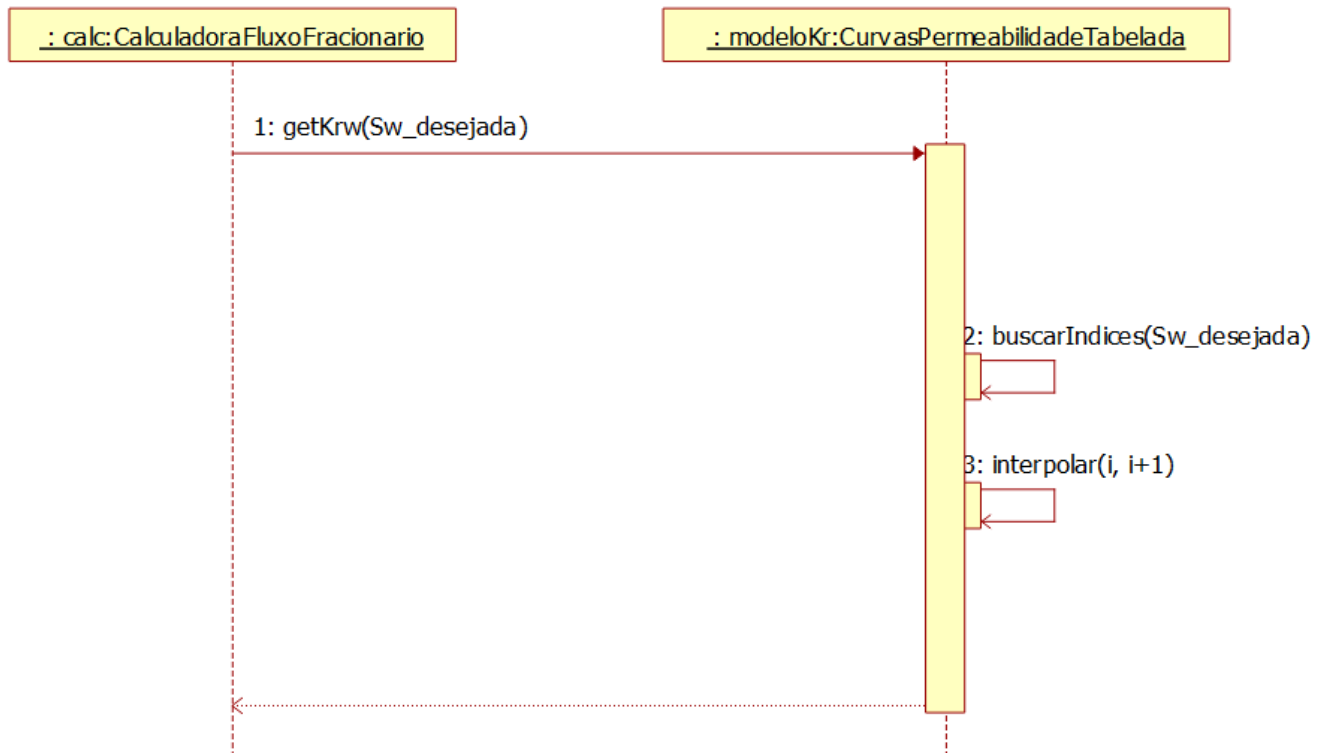


Figura 3.3: Diagrama de sequência

### 3.3 Diagrama de comunicação – colaboração

O diagrama de comunicação da Figura 3.3 apresenta a mesma interação do diagrama de sequência geral, mas foca na colaboração e nas mensagens trocadas entre os objetos, em vez da ordem temporal.

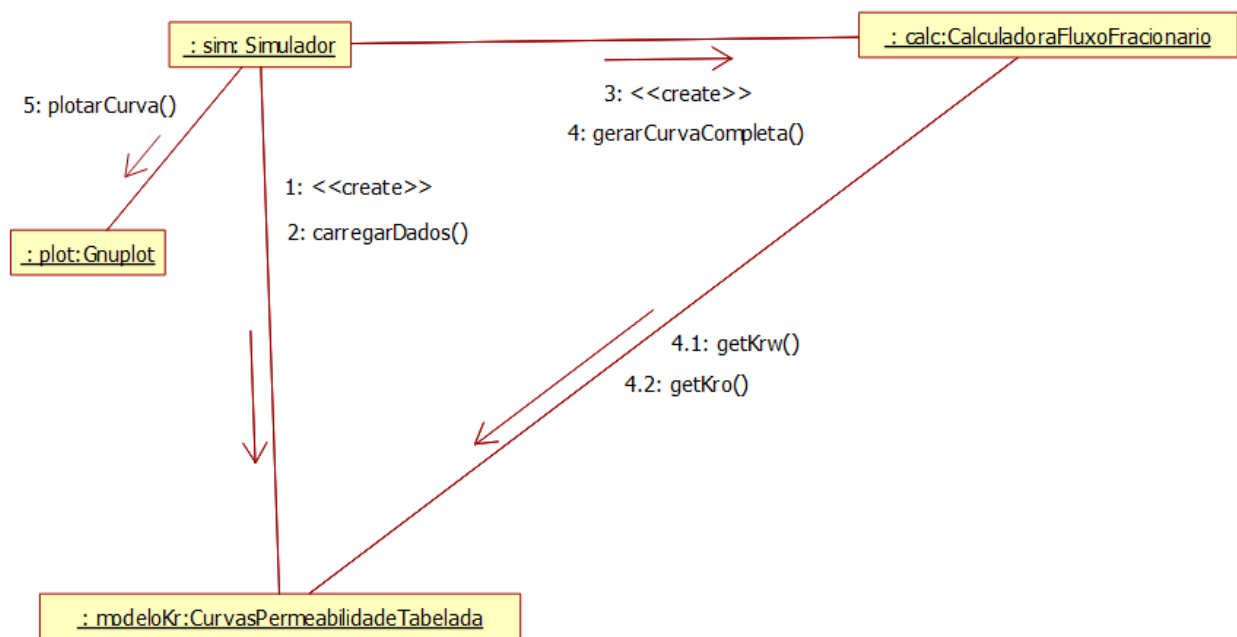


Figura 3.4: Diagrama de comunicação

## 3.4 Diagrama de estado

O diagrama da Figura 3.5 modela os estados dinâmicos de um objeto Simulador ao longo de sua vida, desde a criação até a finalização da execução.

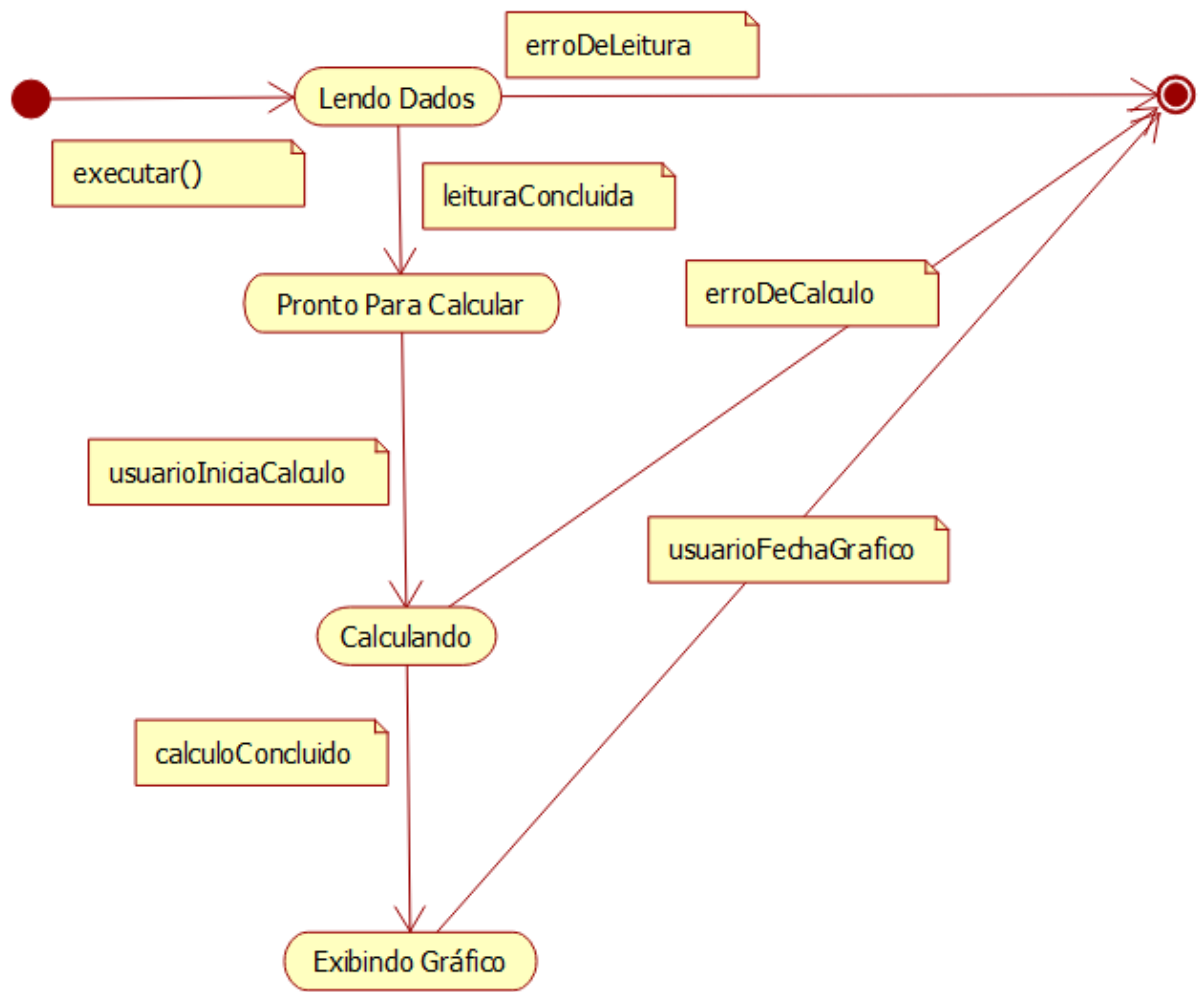


Figura 3.5: Diagrama de máquina de estado

## 3.5 Diagrama de atividades

O diagrama de atividades da Figura 3.6 detalha o algoritmo do método `getKrw()` na classe `Curvas-PermeabilidadeTabelada`, como descrito na Formulação Teórica. Este é um cálculo específico e crucial.

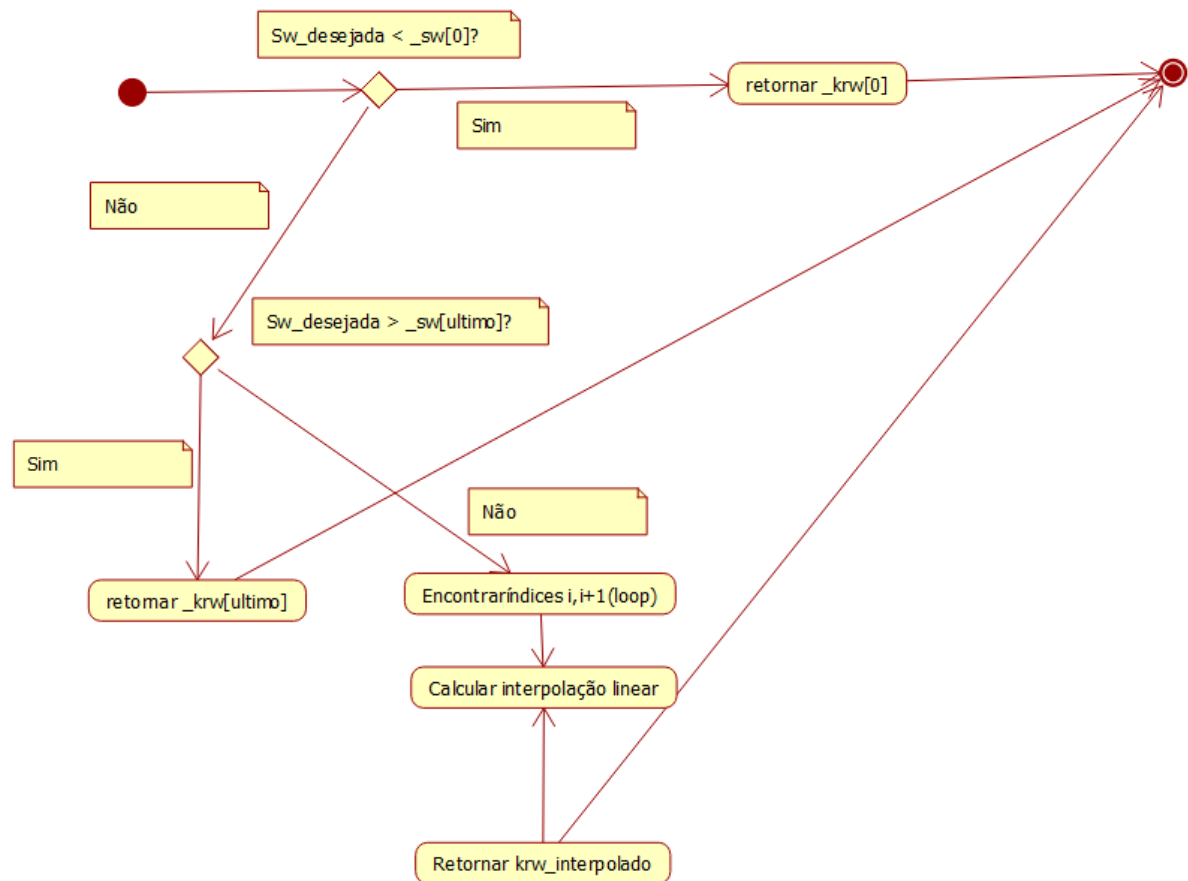


Figura 3.6: Diagrama de atividades

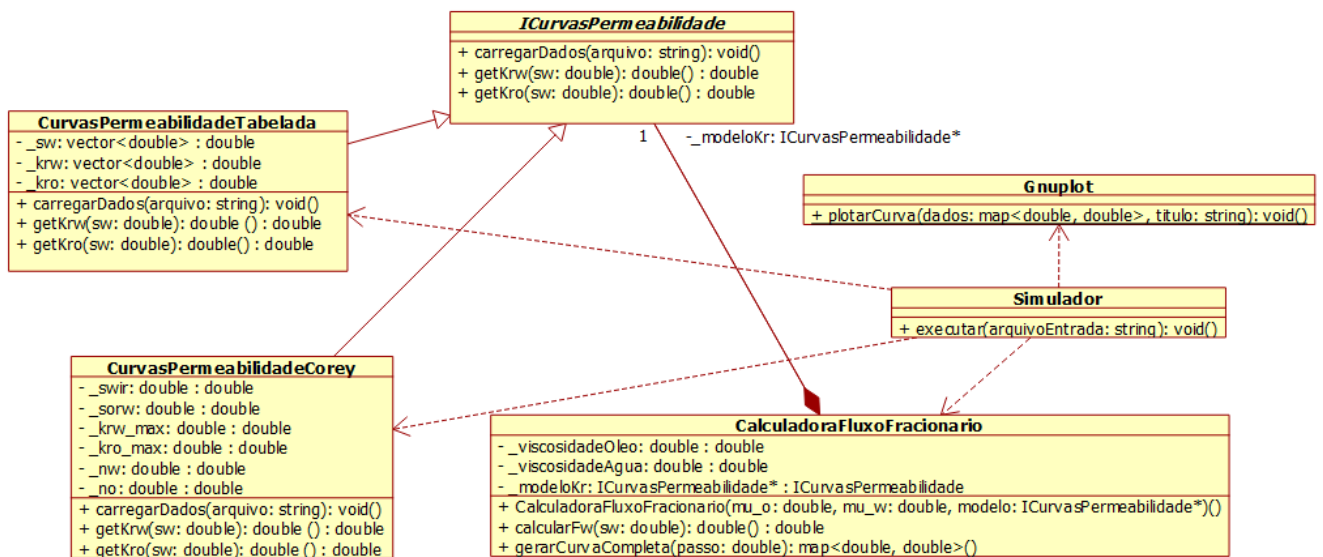


Figura 3.1: Diagrama de classes



# Referências Bibliográficas

- [Ahmed, 2010] Ahmed, T. (2010). *Reservoir Engineering Handbook*. Gulf Publishing Company, 4th edition. 13
- [Buckley and Leverett, 1942] Buckley, S. E. and Leverett, M. C. (1942). Mechanism of Fluid Displacement in Sands. *Transactions of the AIME*, 146:107–116. 12
- [Corey, 1954] Corey, A. T. (1954). The Interrelation Between Gas and Oil Relative Permeabilities. *Producers Monthly*, 19(1):38–41. 8, 12, 15
- [Lake et al., 2010] Lake, L. W. et al. (2010). *Fundamentals of Enhanced Oil Recovery*. Society of Petroleum Engineers. 13

# Apêndice A

## Formato do Arquivo de Entrada

O software utiliza um arquivo de entrada de texto .txt baseado em palavras-chave. O parser identifica a palavra-chave (ex: VISC\_OLEO) e lê o valor na mesma linha.

A palavra-chave MODELO\_KR define qual modelo de permeabilidade será usado: TABELADO ou COREY.

### A.1 Exemplo de Arquivo de Entrada (Modelo Tabelado)

```
# Exemplo de arquivo de entrada para modelo tabelado
# Comentários são ignorados
```

```
VISC_OLEO 1.5
```

```
VISC_AGUA 0.8
```

```
MODELO_KR TABELADO
```

```
# Definição da tabela Sw, Krw, Kro
```

```
# Os dados são lidos até a palavra-chave FIM_DADOS
```

```
DADOS_KR_INICIO
```

```
0.20 0.00 0.90
```

```
0.30 0.05 0.75
```

```
0.40 0.12 0.50
```

```
0.50 0.20 0.30
```

```
0.60 0.30 0.15
```

```
0.70 0.40 0.05
```

```
0.80 0.50 0.00
```

```
FIM_DADOS
```



## A.2 Exemplo de Arquivo de Entrada (Modelo Corey)

```
# Exemplo de arquivo de entrada para modelo Corey  
# Comentários são ignorados
```

```
VISC_OLEO 2.0
```

```
VISC_AGUA 1.0
```

```
MODELO_KR COREY
```

```
# Parâmetros da correlação de Corey
```

```
COREY_SWIR      0.15
```

```
COREY_SORW      0.20
```

```
COREY_KRW_MAX   0.5
```

```
COREY_KRO_MAX   0.9
```

```
COREY_NW        2.0
```

```
COREY_NO        2.5
```

# Apêndice B

## Guia de Compilação e Execução

### B.1 Dependências

Compilador C++: Um compilador moderno com suporte a C++17 (ex: g++ ou clang).

Gnuplot: O executável gnuplot deve estar instalado e acessível no PATH do sistema.

### B.2 Compilação (Exemplo g++)

```
g++ -std=c++17 -o fw_calc main.cpp Simulador.cpp Gnuplot.cpp CalculadoraFluxoF
```

### B.3 Execução

```
./fw_calc meu_arquivo_de_entrada.txt
```

Se a execução for bem-sucedida, uma janela do Gnuplot será aberta e um arquivo output.csv será gerado.

# Índice Remissivo

## **A**

Análise orientada a objeto, 15

AOO, 15

## **C**

Casos de uso, 9

colaboração, 17

comunicação, 17

Concepção, 6

## **D**

Diagrama de colaboração, 17

Diagrama de estado, 18

Diagrama de sequência, 16

## **E**

Elaboração, 11

Especificação, 6

especificação, 6

estado, 18

Eventos, 16

## **M**

Mensagens, 16

## **R**

Requisitos, 8

Requisitos funcionais, 8

Requisitos não funcionais, 8