

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

PROJETO ENGENHARIA
DESENVOLVIMENTO DO SOFTWARE:
CÁLCULO DOS FATORES DE RECUPERAÇÃO AVANÇADA DE
PETRÓLEO À PARTIR DE TESTES DE FLUXO EM MEIO POROSO
TRABALHO DA DISCIPLINA PROGRAMAÇÃO PRÁTICA

LAÍSA RIGUETI BRASIL - Versão 1:
Prof. André Duarte Bueno

MACAÉ - RJ
JULHO - 2019

Sumário

1	Introdução	1
1.1	Escopo do Problema	1
1.2	Objetivos	2
2	Especificação	3
2.1	Especificação do Software – Descrição dos Requisitos	3
2.1.1	Nome do sistema/produto e componentes	3
2.1.2	Especificação	4
2.1.3	Requisitos funcionais	4
2.1.4	Requisitos não funcionais	5
2.2	Casos de uso do software	5
2.2.1	Diagrama de casos de uso geral	5
2.2.2	Diagrama de casos de uso específico	6
3	Elaboração	7
3.1	Análise de Domínio	7
3.2	Conceitos Teóricos	9
3.2.1	Conceitos Fundamentais e Propriedades das Rochas e Fluidos . . .	10
3.2.2	Teste Laboratorial para utilização do Software	13
3.3	Diagrama de pacotes - assuntos	19
4	AOO - Análise Orientada a Objeto	20
4.1	Diagrama de Classes	20
4.1.1	Dicionário de classes	22
4.2	Diagrama de seqüência - eventos e mensagens	22
4.2.1	Diagrama de seqüência geral	22
4.3	Diagrama de comunicação - colaboração	23
4.4	Diagrama de máquina de estado	24
4.5	Diagrama de atividades	24
5	Projeto	26
5.1	Projeto orientado a objeto - POO	28

5.2	Diagrama de implantação	28
6	Implementação	30
6.1	Código fonte	30
7	Teste	46
8	Documentação	52
8.1	Documentação do usuário	52
8.1.1	Como instalar o software	52
8.2	Documentação para desenvolvedor	53
8.2.1	Dependências	53
8.2.2	Como gerar a documentação usando doxygen	54

Capítulo 1

Introdução

No presente trabalho de engenharia, desenvolve-se um projeto computacional em linguagem orientada a objeto C++ que tem como principal objetivo o gerenciamento de informações e realização de cálculos para o monitoramento de frações de recuperação de óleo a partir de um experimento laboratorial que realiza testes de injeção com métodos de recuperação avançada de petróleo.

Levando em consideração que no Laboratório de Engenharia e Exploração de Protróleo (LENEP) da Universidade Estadual do Norte Fluminense (UENF) não há softwares para realização dos cálculos que quantificam as propriedades envolvidas neste tipo de experimento, o presente projeto possibilita os cálculos das mesmas, podendo, portanto, ser usado para fins acadêmicos.

1.1 Escopo do Problema

A recuperação avançada ou EOR (*Enhanced Oil Recovery*) é o conjunto de métodos utilizados atualmente para produzir o óleo remanescente, em sua maioria, após os processos de recuperação primária e secundária. Na literatura, encontram-se referenciados alguns dos projetos de grande escala que utilizaram sistemas químicos como método de injeção. Os reservatórios de óleo em todo o mundo representam sistemas biológicos muito complicados para os quais as simulações laboratoriais de atividades microbianas se tornam desafiadoras como uma parte essencial e fundamental para o desenvolvimento em campo. Neste contexto, o projeto focará nos processos de recuperação avançada de petróleo melhorada microbiologicamente (*Microbial Enhanced Oil Recovery, MEOR*) a partir das propriedades dos biossurfactantes.

O conhecimento das propriedades das rochas e fluidos é essencial para o desenvolvimento de qualquer tecnologia para aumentar o fator de recuperação do petróleo, como por exemplo, as definições de porosidade, molhabilidade, permeabilidade, razão de mobilidade, tensão interfacial, tensoativos, fator de recuperação, dentre outros. Desta forma, uma das finalidades do programa será calcular à partir das propriedades físicas e químicas

dos fluidos de injeção e das amostras de rochas, a permeabilidade absoluta e o volume poroso. Além disso, o usuário poderá incluir as informações dos volumes produzidos através dos experimentos de fluxo em meios porosos para avaliar o potencial de recuperação com o método adotado.

O programa buscará auxiliar o pesquisador/usuário com a compilação dos dados, otimizando tempo, minimizando possíveis erros de cálculos das frações recuperadas e mantendo um software otimizado que poderá ser utilizado para diferentes tipos de fluidos, rochas e métodos de recuperação.

1.2 Objetivos

Os objetivos deste trabalho são:

- Objetivo geral:
 - Esse projeto tem como objetivo utilizar a programação para facilitar as pesquisas laboratoriais de novos alunos pesquisadores do setor de Engenharia de Petróleo, criando um software capaz de automatizar os cálculos envolvidos no projeto.
- Objetivos específicos:
 - Calcular a permeabilidade absoluta de amostras rochosas;
 - Obter gráfico da permeabilidade absoluta da amostra (*plugs*);
 - Calcular o volume poroso de amostras rochosas;
 - Calcular as frações de recuperação de óleo a partir do método utilizado: Secundário e/ou Avançado;
 - Possibilitar a inclusão de informações de novos fluidos e amostras (*plugs*);
 - Obter gráficos com as curvas de produção de água e de óleo através da recuperação secundária e avançada a partir da entrada de dados pelo usuário (volumes produzidos por minuto);
 - Obter as saturações iniciais e finais dos fluidos injetados no meio poroso;
 - Obter o fator de recuperação total de óleo através dos métodos de recuperação utilizados.

Capítulo 2

Especificação

Neste capítulo do projeto de engenharia, apresenta-se a especificação do software a ser modelado e desenvolvido para aplicação em testes de recuperação avançada de óleo, um processo altamente visado no setor de petróleo a fim de garantir otimização e eficiência nos métodos adotados.

2.1 Especificação do Software – Descrição dos Requisitos

Nesta seção são descritas as principais características, além dos requisitos para a utilização do software desenvolvido.

2.1.1 Nome do sistema/produto e componentes

Na Tabela 2.1, apresenta-se as características do software:

Tabela 2.1: Características do software

NOME	Software para Cálculo dos Fatores de Recuperação de Óleo para Métodos EOR
COMPONENTES PRINCIPAIS	Sistema para cálculo das propriedades do meio poroso e fatores de recuperação total de petróleo a partir de testes de injeção com métodos de recuperação avançada de petróleo
MISSÃO	Calcular volume poroso e porosidade de amostras de rochas; Calcular permeabilidade absoluta de amostras; Gerar gráficos dos volumes produzidos durante os testes de injeção; Calcular os fatores de recuperação secundária e avançada; Calcular o fator de recuperação total.

2.1.2 Especificação

O projeto a ser desenvolvido consiste de um programa que deverá realizar cálculos de alguns parâmetros de amostras rochosas, como volume poroso e porosidade, além de propriedades do meio poroso, como a permeabilidade absoluta a partir da equação de Darcy. Conforme realização de testes de injeção de fluidos neste meio poroso, será calculado os volumes de fluidos produzidos e os fatores de recuperação de petróleo com métodos avançados.

Na dinâmica de execução do software, o usuário deverá entrar com os dados da amostra medidos em laboratório, como diâmetro, comprimento, peso da amostra seca, peso da amostra molhada e com os dados em arquivo *ascii* com extensão *.txt* contendo os volumes de óleo recuperados por minuto. Os cálculos, bem como os gráficos da permeabilidade e volumes produzidos durante os métodos de injeção serão fornecidos ao usuário a partir do terminal. Os dados com as suas respectivas unidades estão listados abaixo:

- **Dados/Atributos relativos à amostra (plug):**

Medidas de diâmetro para cálculo da média - mm;

Medidas de comprimento para cálculo da média - mm;

Peso da amostra seca - g;

Peso da amostra molhada (saturada com água salina) – g;

- **Dados/Atributos relativos aos fluidos de injeção:**

Vazões de injeção para medida da permeabilidade – mL/min;

Pressões adquiridas para medida da permeabilidade – psia;

- **Dados/Atributos relativos ao teste de injeção:**

Vazão de Injeção para teste de recuperação – mL/min;

Viscosidade do fluido de injeção – cP. Tempos de medidas – min.;

Volumes produzidos – mL.

2.1.3 Requisitos funcionais

Apresenta-se a seguir os requisitos funcionais:

RF-01: O usuário deverá ser capaz de incluir valores de parâmetros e propriedades das rochas, fluidos e testes laboratoriais para execução dos cálculos;

RF-02: O programa deverá permitir o carregamento de dados a partir de um arquivo de disco criado pelo usuário;

RF-03: Os resultados deverão ser exportados como textos e/ou gráficos.

2.1.4 Requisitos não funcionais

Apresenta-se a seguir os requisitos não funcionais.

RNF-01: Os cálculos devem ser feitos utilizando-se formulações matemáticas conhecidas da literatura;

RNF-02: O programa deverá ser multi-plataforma, podendo ser executado em Windows, GNU/Linux, Mac, OsX.

2.2 Casos de uso do software

A Tabela 2.2 apresenta um caso de uso do sistema, bem como os diagramas de casos de uso.

Tabela 2.2: Caso de uso geral do sistema

Nome do caso de uso:	Cálculo dos Fatores de Recuperação de Óleo para Métodos EOR
Resumo/Descrição:	Cálculo das propriedades do meio poroso e fatores de recuperação total de petróleo a partir de testes de injeção com métodos de recuperação avançada de petróleo
Etapas:	<ol style="list-style-type: none">1. Inserir dados de entrada2. Calcular permeabilidade absoluta da amostra3. Inserir dados em formato .txt4. Plotar os gráficos de volumes produzidos5. Calcular fator de recuperação secundário6. Calcular fator de recuperação avançado7. Calcular fator de recuperação total e apresentar os valores calculados no terminal.
Cenários Alternativos:	Inserir valores negativos ou incompatíveis com a ordem de grandeza do problema.

2.2.1 Diagrama de casos de uso geral

O diagrama de caso de uso geral da Figura 1 exhibe o usuário interagindo com o software para obter os fatores de recuperação de petróleo a partir do método EOR. Neste caso de uso geral, o usuário insere os dados de entrada, calcula a permeabilidade absoluta da amostra, insere um arquivo .txt para possibilitar o plot dos gráficos de produção de água e óleo, plota os gráficos e insere os dados dos volumes produzidos para cálculo das frações de recuperação. O usuário pode então fazer a análise dos resultados obtidos.

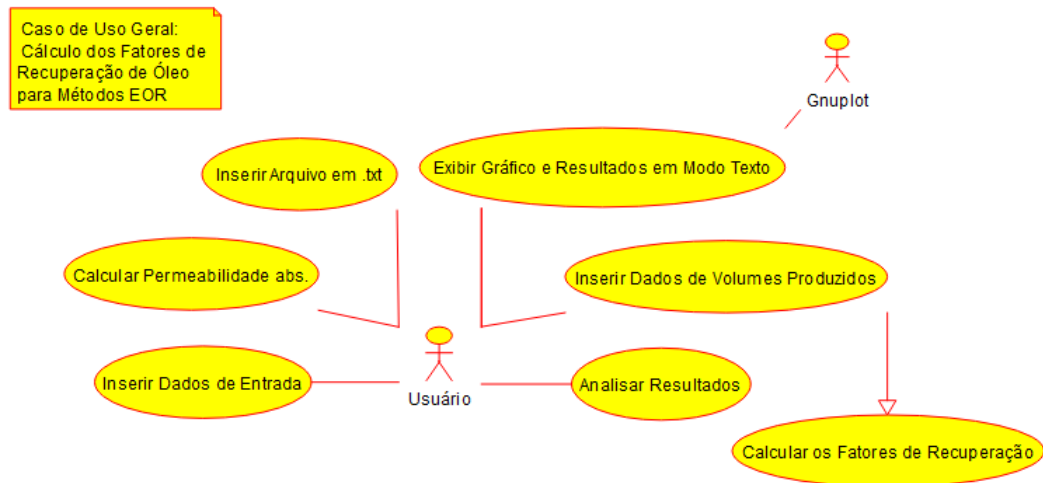


Figura 2.1: Diagrama de caso de uso - Caso de uso geral

2.2.2 Diagrama de casos de uso específico

O diagrama de caso de uso específico da Figura 2 é um detalhamento do caso de uso para calcular os fatores de recuperação, ele mostra a interação usuário com o software para calcular a recuperação de óleo no meio poroso, usando o método EOR.

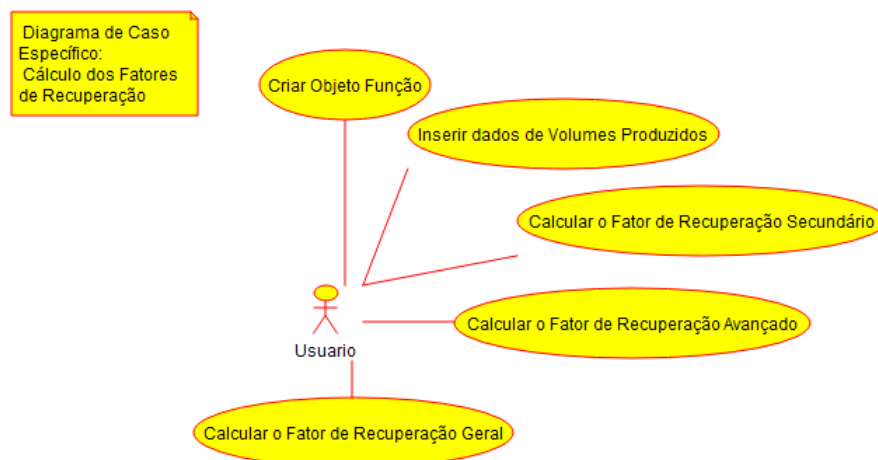


Figura 2.2: Diagrama de caso de uso - Caso de uso específico: calcular fatores de recuperação

Capítulo 3

Elaboração

Depois da definição dos objetivos, da especificação do software e da montagem dos primeiros diagramas de caso de uso, a etapa do projeto de engenharia envolve a identificação e o estudo dos conceitos relacionados ao sistema a ser desenvolvido, isto é, a análise de domínio e a identificação de pacotes. Na elaboração fazemos uma análise dos requisitos, ajustando os requisitos iniciais de forma a desenvolver um sistema útil e adequado, que atenda às necessidades do usuário além de permitir seu reuso e futura extensão.

3.1 Análise de Domínio

O programa que será desenvolvido está relacionado a área da engenharia de reservatório, na qual utiliza-se métodos de recuperação avançada para produção de óleo remanescente.

Desde o início das atividades da indústria de petróleo e gás, a produção de hidrocarbonetos foi baseada na manutenção da energia que ocasionalmente provinha do próprio reservatório a ser explorado. Essa quantidade de energia é denominada “energia primária”, sendo sua grandeza determinada pelo volume e pela natureza dos fluidos existentes na acumulação, bem como pelos níveis de pressão e de temperatura presentes no reservatório. As principais forças motrizes neste período são o gás em solução, o aporte de aquífero, a expansão da capa de gás e segregação gravitacional [Rosa et al., 2006]. Devido às forças viscosas e capilares presentes no meio poroso, ocorre uma dissipação da energia primária no processo de produção [Thomas, 2001] refletindo no decréscimo da pressão do reservatório durante sua vida produtiva, e consequente, na redução da produtividade dos poços. A eficiência da recuperação primária é geralmente baixa ($\sim 15\%$) e dependendo do mecanismo de produção, das características do reservatório e das propriedades dos fluidos acumulados, tal eficiência pode inclusive ser nula [Mothé et al., 2006].

Com a necessidade do aumento da produção de óleo e devido à rápida queda de pressão do reservatório, surgiu a necessidade de se utilizar métodos artificiais capazes de suplementar esta energia, conhecidos como métodos de recuperação secundária [Rosa et al., 2006].

A injeção de fluidos no reservatório tem como objetivo fornecer pressões necessárias para deslocar o óleo da rocha reservatório e ocupar o espaço deixado pelo fluido produzido. Os métodos de recuperação secundária conseguem elevar a recuperação média de 15% para 30%. Porém, existe ainda uma parcela do “*Original Oil In-Place*” (OOIP) - volume que se refere àquele estimado antes de iniciada a produção do reservatório - que permanece retida [Bruce et al., 1947]. Os métodos mais comuns e mais baratos são a injeção de água e a reinjeção de gás de produção [Mothé et al., 2006].

Desta forma, cerca de 70% do óleo permanece no reservatório, sendo que sua curva de produção continua a declinar [Carrero et al., 2007]. Isso se deve principalmente à alta viscosidade do óleo residual e à alta tensão interfacial entre o hidrocarboneto e a fase aquosa, que resultam em altas forças capilares que retêm o óleo em pequenos poros na rocha [Lazar et al., 2007]. Como resultado, para recuperar o óleo residual após a aplicação dos métodos convencionais, é indicado a adoção de um método de recuperação avançada de petróleo.

Nas últimas décadas, a produção primária e os métodos de recuperação secundária passaram a ser classificados como métodos convencionais de recuperação, enquanto que os métodos terciários passaram a ser denominados como métodos especiais de recuperação ou métodos de Recuperação Avançada conhecidos como EOR (*Enhanced Oil Recovery*) [Ali et al., 1996, Thomas, 2001, Rosa et al., 2006, Muggeridge et al., 2014].

Os métodos de recuperação terciária são mais indicados para a recuperação de campos maduros, pois buscam recuperar reservatórios que apresentam óleos com alta viscosidade, elevadas tensões interfaciais, ou que já atingiram ou estão perto de atingir a saturação de óleo residual. Neste cenário, a aplicação dos métodos convencionais de recuperação secundária não é suficiente.

Os principais métodos de EOR incluem: i) métodos térmicos, como injeção de vapor, combustão in situ e injeção de água quente; (ii) métodos químicos com a injeção de produtos químicos, como injeção de polímero, de surfactante, injeção alcalina, injeção de microrganismos (*Microbial Enhanced Oil Recovery, MEOR*) e (iii) métodos miscíveis, como injeção de CO₂ e hidrocarbonetos miscíveis.

A eficiência de um cenário de EOR pode ser variável para diferentes reservatórios, dependendo das condições do reservatório e do tipo de óleo. Como resultado, para obter o máximo benefício de um método EOR, as condições do óleo não recuperado, bem como os mecanismos de produção do óleo por meio desses processos, devem ser bem compreendidos e estudados [Venuto, 1989, Rosa et al., 2006, Muggeridge et al., 2014].

Com o declínio de reservas, os métodos de recuperação avançada (EOR) têm se tornado um tema importante na indústria petrolífera. Os objetivos gerais destes métodos são de impulsionar o deslocamento de óleo através do reservatório devido alteração das propriedades físicas e químicas nas interações rocha-fluidos contidos no meio poroso [Lake et al., 1989]. O objetivo principal visa aumentar o fator de recuperação de óleo, ou seja, recuperar a maior fração do OOIP em relação aos métodos convencionais

[Green et al., 1998].

A partir dos inúmeros estudos feitos na aplicação do método, a proporção das melhorias na recuperação do óleo se mostrou influenciada por vários fatores, incluindo características individuais do reservatório como litologia, natureza dos arenitos, porosidade, permeabilidade, temperatura do reservatório, grau API do petróleo bruto, entre outros. Além das condições do fundo do poço e da natureza das formações, composição química e concentração da injeção, o tempo de adaptação e o tempo de aplicação constituem os principais fatores que governam a EOR.

Perspectivas da engenharia de reservatório, com foco em questões como o aumento de escala dos resultados de laboratório para projetos de campo e operações fornecem uma estrutura consistente para comparar os processos EOR para escolha do método adequado. Isso justifica a lógica por trás dos testes de campo em escala piloto que são realizados após o desenvolvimento bem-sucedido dos processos EOR através de testes em laboratório [Sen, 2008]. Uma análise de dados de uma base de dados de informações e cálculos concisos empregando os métodos EOR resultam na avaliação da eficiência técnica e econômica dos processos, além de fornecer uma fonte de informação útil para prever a resposta ao tratamento em qualquer reservatório [Portwood et al., 1995].

Uma das técnicas é o método de recuperação microbiológica ou MEOR (*Microbial Enhanced Oil Recovery*) no qual se utilizam diferentes estratégias de injeção de metabólitos produzidos por microrganismos, dentre eles, que atuem como biossurfactantes, mobilizando o óleo trapeado em reservatórios e aumentando a eficiência de varrido por meio da redução da viscosidade e da tensão superficial óleo/rocha.

Neste projeto, o objetivo dos resultados do software será a avaliação da eficiência de um biossurfactante produzido pela bactéria *Bacillus safensis* em processo de recuperação avançada de petróleo. Para isso, foram realizados testes de fluxo em meio poroso saturado com solução salina e óleo sintético. Injetou-se uma solução salina de 30.000 ppm como método de recuperação secundária e, após atingir a saturação de óleo residual, injetou-se a solução do biossurfactante como método MEOR à uma vazão de 0,5 mL/min. Os fatores de recuperação serão adquiridos à partir da entrada dos dados dos volumes produzidos por minuto e retorno dos cálculos.

3.2 Conceitos Teóricos

O conhecimento das propriedades das rochas e fluidos é essencial para o desenvolvimento de qualquer metodologia para aumentar o fator de recuperação do petróleo. Assim, será mostrado definições de porosidade absoluta e relativa, molhabilidade, permeabilidade absoluta, fator de recuperação, entre outros.

3.2.1 Conceitos Fundamentais e Propriedades das Rochas e Fluidos

O entendimento de conceitos fundamentais relativo a algumas propriedades das rochas e dos fluidos é de fundamental importância no desenvolvimento de um trabalho com aplicação na engenharia de reservatórios.

• Porosidade

A porosidade de uma rocha é definida como a razão entre o volume poroso, capaz de armazenar fluidos, e o volume total da rocha, que é dado pela soma do volume poroso e do volume da parte sólida da rocha. A porosidade mede o volume dos espaços vazios em um meio poroso, independente de estarem ou não interligados. Portanto, a porosidade é um parâmetro petrofísico de grande importância visto que se consegue medir a capacidade de armazenagem de fluidos em um corpo poroso [Rosa et al., 2006].

Assim, a porosidade (ϕ), expresso em porcentagem, pode ser definida como a razão entre o volume poroso (V_p) e o volume total (V_t) da amostra (Equação 3.1), onde o volume total é dado pela soma do espaço poroso e da fase sólida (Equação 3.2).

$$\phi = \frac{V_p}{V_t} \quad (3.1)$$

$$V_{total} = V_p + V_{sólida} \quad (3.2)$$

A porosidade pode ser absoluta, quando representar a relação entre o volume total de vazios e o volume total da rocha (Equação 3.3), ou efetiva quando a relação for entre o volume dos espaços vazios interconectados de uma rocha e o volume total da mesma (Equação 3.4) [Rosa et al., 2006]. Vale ressaltar que a porosidade efetiva é o parâmetro de maior interesse na engenharia de reservatórios, pois representa o espaço ocupado por fluidos que podem ser deslocados através do meio poroso, ou seja, representa o volume máximo de fluidos que pode ser extraído de um meio poroso [Thomas, 2001].

$$\phi = \frac{V_{pinterconectados} + V_{pisolados}}{V_t} \quad (3.3)$$

$$\phi = \frac{V_{pinterconectados}}{V_t} \quad (3.4)$$

• Molhabilidade

O deslocamento de petróleo por meio dos poros da rocha reservatório também é influenciado por outros parâmetros petrofísicos como a molhabilidade, a qual pode ser definida como a capacidade de um fluido de se espalhar em uma superfície sólida na presença de outros fluidos, ou seja, é a tendência da superfície de ter mais afinidade por um fluido em detrimento de outro, também presente no meio poroso. Isto significa que em um fluxo

multifásico, um fluido tem mais afinidade com o meio poroso que outros fluidos presentes [Rosa et al., 2006, Dandekar, 2013]. Desta forma, a molhabilidade quantifica a afinidade que a superfície da rocha apresenta para cada fluido na presença de outros, estando relacionada com as forças intermoleculares que atuam entre a superfície e as moléculas dos líquidos presentes. Em reservatórios de petróleo, encontram-se basicamente duas fases líquidas, formadas pelo óleo e a água [Rosa et al., 2006].

A molhabilidade de uma rocha-reservatório varia de acordo com a interação dos fluidos com a superfície da rocha reservatório, podendo ser mais fortemente molhável ao óleo ou à água [Rosa et al., 2006, Agbalaka et al., 2008]. Além disso, a molhabilidade pode variar a partir do contato óleo-água dependendo da saturação de água, ou devido às diferenças na composição da rocha ou mineralogia, sendo amplamente reconhecida como uma importante parte da descrição do reservatório [Buckley, 1999].

A molhabilidade tem impacto fundamental nas características petrofísicas das rochas e uma das formas de se considerar os impactos deste parâmetro na produção e recuperação de petróleo são através das curvas de permeabilidade relativa [Donaldson et al., 1985]. Donaldson e Thomas (1971), em análises de laboratórios, verificaram um aumento da permeabilidade relativa à água e uma diminuição da permeabilidade relativa ao óleo quando a molhabilidade ao óleo era mais acentuada. Isto mostra que quanto mais o meio poroso for molhável ao óleo, maior a saturação de água posicionada no centro dos poros competindo com o óleo nos caminhos mais permeáveis, reduzindo a permeabilidade relativa ao óleo, e aumentando a permeabilidade relativa à água.

• Permeabilidade

A permeabilidade é uma das características petrofísicas mais importantes de um reservatório, sendo a capacidade da rocha de permitir o escoamento de fluidos. Uma rocha pode ter alta porosidade e apresentar baixa permeabilidade, caso os poros não sejam bem conectados, ou seja, para que o reservatório seja produtivo não basta um alto valor de porosidade, a rocha deve possuir a capacidade de permitir o deslocamento de fluidos através dela [Rosa et al., 2006]. Este parâmetro é um dos que tem mais influência na determinação da capacidade de produção de hidrocarbonetos acumulados. A permeabilidade (k) é uma propriedade dinâmica, definida como a capacidade de um dado meio poroso se deixar atravessar por um fluido [Rosa et al., 2006]. Ela é uma função da posição e pressão, e varia fortemente com o tamanho dos poros e sua distribuição em determinado local [Lake et al., 1989].

O conceito de permeabilidade aparece na lei que governa o deslocamento dos fluidos através de meios porosos, conhecida como a Lei de Darcy, sendo medida em milidarcy (md). Existem dois tipos de permeabilidades, a permeabilidade absoluta, quando o reservatório está saturado com um único fluido, e a permeabilidade efetiva, quando existem dois ou mais fluidos coexistindo dentro de uma mesma rocha [Albuquerque et al., 2007, Dandekar, 2013]. O desenvolvimento da expressão que permite encontrar a permeabi-

lidade absoluta de um meio poroso é utilizado até os dias atuais na indústria do petróleo. O experimento original de Darcy investigou o fluxo de água através da areia e concluiu que um fluxo linear com vazão de injeção (q) é função da condutividade hidráulica (k), da área da seção transversal (A), do diferencial de pressão da entrada para a saída (Δp) e do comprimento do meio poroso (L), conforme mostra a Equação 3.5 [Rosa et al., 2006, Lake et al., 1989].

$$q = \frac{\kappa A \Delta P}{\mu L} \quad (3.5)$$

A permeabilidade absoluta pode ser calculada isolando k na equação de Darcy conforme Equação 3.6.

$$\kappa = \frac{\mu L q}{A \Delta P} \quad (3.6)$$

Sendo:

K = Permeabilidade

A = Área da seção transversal

μ = Viscosidade

L = Comprimento do meio poroso

Desta forma, durante um fluxo, se todas as variáveis são conhecidas, menos a permeabilidade, torna-se possível encontrá-la. A forma para a Equação 3.6 é utilizada para um fluxo linear. Em poços, com fluxo radial, modifica-se a geometria para definir a permeabilidade levando em consideração o raio externo do reservatório, o raio do poço, a pressão externa do reservatório, e a pressão medida no poço em uma determinada altura do reservatório [Rosa et al., 2006].

A permeabilidade efetiva (k_e), quando dois ou mais fluidos saturam o meio poroso [Rosa et al., 2006], sempre apresentará valores menores do que o valor da permeabilidade absoluta da rocha. O cálculo das permeabilidades efetivas à água e ao óleo (k_w e k_o) também pode ser realizado usando a Lei de Darcy conforme Equações 3.7 e 3.8 [Rosa et al., 2006].

$$\kappa_w = \frac{\mu_w L q_w}{A \Delta P} \quad (3.7)$$

$$\kappa_o = \frac{\mu_o L q_o}{A \Delta P} \quad (3.8)$$

A razão entre a permeabilidade efetiva de determinado fluido no meio poroso e a permeabilidade absoluta é denominada permeabilidade relativa (k_r), a qual pode ser representada pelas Equações 3.9 e 3.10 para um sistema bifásico óleo-água.

$$\kappa_{rw} = \frac{\kappa_w}{\kappa} \quad (3.9)$$

$$\kappa_{ro} = \frac{\kappa_o}{\kappa} \quad (3.10)$$

Este parâmetro sofre efeitos da variação da saturação dos fluidos, da molhabilidade da rocha, da estrutura dos poros da rocha, da tensão de confinamento, do teor de argila da rocha, da migração de finos, da temperatura, além das variações de tensão interfacial, viscosidade e velocidade do fluxo [Dandekar, 2013].

O aumento da saturação de um fluido molhante no meio poroso em relação a outro fluido chama-se de embebição, e, por outro lado, quando existe uma redução de saturação do fluido que molha preferencialmente a rocha em relação a outro fluido, tem-se uma drenagem [Donaldson et al., 1985]. Assim, no processo de embebição, é necessário que haja uma determinada saturação da fase molhante no início do fluxo, chamada saturação de água conata ou saturação irreduzível (S_{wi}). Da mesma forma ocorre no processo de drenagem, e essa saturação é denominada saturação de óleo residual (S_{or}). A saturação da fase não molhante atinge seu valor máximo a saturações menores que 100%, o que indica que nem todo o meio poroso interligado irá contribuir ao fluxo desta fase [Núñez, 2011].

• Fator de recuperação de óleo

O fator de recuperação de óleo, FR, é a razão entre o volume de óleo produzido do reservatório ($V_{ÓleoProduzido}$) e o volume total de óleo dentro do reservatório ($V_{ÓleoTotal}$), conforme Equação 3.11. Os valores de FR variam de 0 a 1, ou podem ser expressos em porcentagem [Rosa et al., 2006].

$$FR = \frac{V_{ÓleoProduzido}}{V_{ÓleoTotal}} \quad (3.11)$$

3.2.2 Teste Laboratorial para utilização do Software

Neste tópico são apresentados os procedimentos experimentais adotados para caracterizar o biossurfactante e realizar o teste de recuperação avançada por meio da injeção de biossurfactante a fim de verificar a recuperação adicional de óleo pelo uso do método *MEOR*. O fluxograma da Figura 3.2 apresenta o protocolo experimental adotado, o qual serviu como base para criação das classes do programa.

• Medidas das viscosidades dos fluidos de injeção

As viscosidades do óleo, solução salina e biossurfactante foram aferidas no viscosímetro Fann Viscosimeter Model 35 A, que apresenta acurácia de $\pm 0,5$ cP.

O óleo sintético utilizado para o teste foi uma mistura de óleo cru (Bacia de Santos) com n-hexano (50/50).

• Seleção e caracterização da amostra de rocha

Utilizou-se uma amostra de arenito Berea (nomeado como BEREa) (Figura 3.1) em formato cilíndrico (plug).



Figura 3.1: Ilustração do plug de arenito Berea utilizado no teste de recuperação

O plug foi seco em estufa e submetido a vácuo para remoção do ar. O primeiro passo foi inferir as medidas de comprimento e diâmetro, onde realizou-se três medidas utilizando-se um paquímetro com precisão de $\pm 0,01$ mm. O resultado utilizado foi a média dos três valores encontrados, a fim de minimizar a margem de erro.

Após aferida a dimensão do plug, ele foi transferido para uma estufa onde permaneceu durante 24 horas, com temperatura constante de 60°C (padrão utilizado em laboratório para secagem da amostra), com o objetivo de assegurar que toda a umidade ou possível água residual evaporasse.

Após a etapa de secagem o plug foi pesado com uma balança de precisão milesimal Adventure OHAUS. Aferido o peso, o plug foi colocado em um becker, onde ficou totalmente imerso em solução salina (30.000 ppm de NaCl) para etapa de saturação. Nesta etapa, utilizou-se uma bomba à vácuo BUCHI Vac V-500 acoplada a um dessecador com secadora, durante 4 horas, mantendo uma pressão de 2 mBar. Terminado o tempo para a saturação completa do testemunho, foi realizada novamente a sua pesagem na balança analítica para viabilizar o cálculo da porosidade.



Figura 3.2: Fluxograma experimental

• Determinação da Porosidade

Para calcular a porosidade do plug, utilizou-se o método da maturação à vácuo, que consiste em calcular a porosidade a partir da massa do plug seco (m_{seca}) e do plug saturado (m_{sat}) com um líquido (neste trabalho utilizou-se solução salina). Com estes valores aferidos, além do volume total da amostra (Vol_{Total}) e da determinação da densidade do fluido (ρ_w), foi possível calcular a porosidade (ϕ) através das equações 3.12 e 3.13 [Dandekar, 2013]:

$$Vol_{total} = \frac{\pi D^2 L}{4} \quad (3.12)$$

$$\phi = \frac{m_{sat} - m_{seca}}{Vol_{total} \rho_w} \quad (3.13)$$

• Sistema de injeção de fluidos

O sistema de injeção é composto de diversos equipamentos, que podem ser vistos na Figura 3.3. Para a injeção dos fluidos de recuperação em um *core holder* (câmara de injeção) de fluxo direto com destaque para o *core holder* hidrostático utilizado no experimento houve a junção dos seguintes equipamentos à câmara: garrafas de transferência utilizadas para transferir o óleo, solução salina e solução com biossurfactante até a linha de injeção do core holder e bomba inteligente HPLC operada com vazão constante (sendo a vazão máxima de trabalho deste equipamento de 50 mL/min). Para o experimento foi utilizado um transdutor de pressão com a função de medir o diferencial de pressão total na rocha e um manômetro para medir a pressão de confinamento aplicada ao sistema. Utilizou-se também linhas de Nylon de 1/8" para conectar a bomba HPLC às garrafas de transferência e às garrafas no *core holder*. Os core holders do tipo biaxial são definidos como equipamentos que têm pressão radial e axial comum aplicada à amostra de rocha. Estes tipos de equipamentos são rotineiramente utilizados para permeabilidade a gás e a líquido, além de outras experiências de injeção de fluidos no meio poroso, como foi realizado neste trabalho.

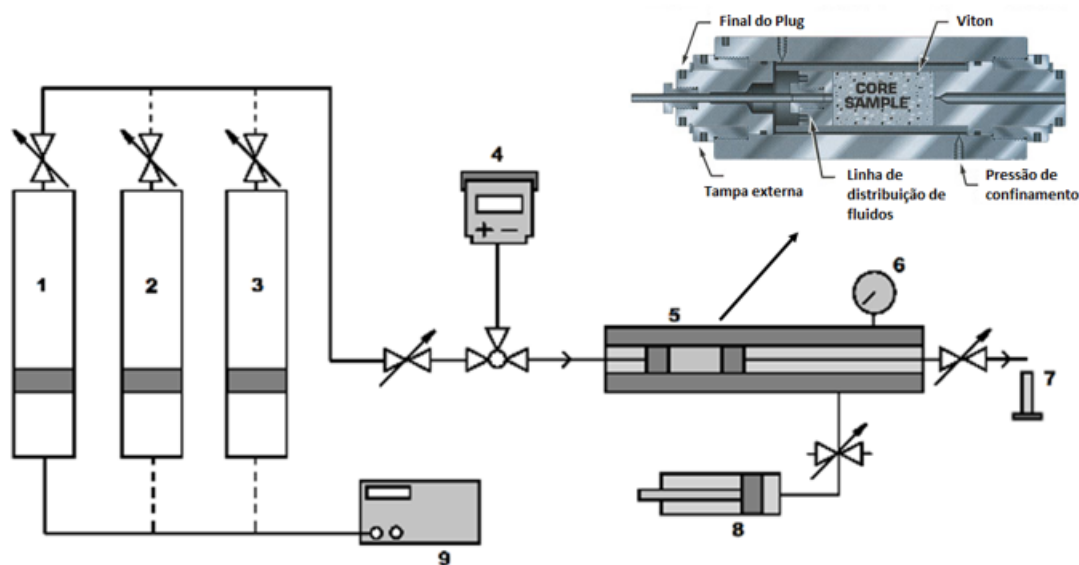


Figura 3.3: Ilustração esquemática do sistema de injeção de fluidos utilizado no experimento de recuperação de petróleo. Em (1),(2) e (3) tem-se as garrafas de transferência onde foram armazenados solução de biossurfactante, óleo e solução salina, respectivamente; (4) transdutor de pressão diferencial com faixa de medida de pressão 0 a 1000 psi; (5) célula de confinamento (*core holder*) do tipo hidrostático; (6) Manômetro do *core holder* para aferir pressão de confinamento; (7) proveta para armaenamento dos fluidos produzidos e (9) bomba HPLC

O procedimento consistiu em posicionar a amostra de rocha saturada no *core colder* e os fluidos em suas devidas garrafas de transferência (óleo, biossurfactante e solução salina). Após montagem foi possível injetar os fluidos para saturação da amostra e obtenção das recuperações convencional (injeção de solução salina) e avançada (injeção de solução com biossurfactante).

• Determinação da permeabilidade absoluta da amostra

A variação da pressão, medida através dos transdutores, permite determinar os parâmetros de interação fluido-rocha, como a permeabilidade. A permeabilidade da amostra é medida em regime permanente, devido à facilidade de medição e cálculos. Todas as fases foram injetadas a vazões constantes até o equilíbrio ser atingido e o diferencial de pressão medido. A permeabilidade absoluta de cada fase foi calculada através da Lei de Darcy [Engelke, 2012].

A Lei de Darcy indica que há direta relação entre a vazão e o gradiente de pressão através de um meio poroso. No *core holder* são gerados diferentes fluxos q (vazão de injeção), medindo-se ΔP (diferencial de pressão). Primeiramente, procura-se determinar a permeabilidade absoluta da rocha com a saturação da mesma em solução salina. Então, iniciou-se a injeção, medindo o diferencial de pressão para diferentes vazões após o fluxo atingir o regime permanente, quando o diferencial de pressão estabilizou. A partir desses dados, foi construído um gráfico de vazão x pressão, em que a inclinação da reta fornece os valores de permeabilidade a partir das equações 3.5 e 3.6.

• Procedimento de injeção de água como método de recuperação secundária

Após a saturação do *plug* com óleo, foi injetada novamente a solução salina até não se observar mais óleo no efluente produzido, obtendo-se a saturação de óleo residual (S_{or}). Foi determinado também o volume de óleo recuperado pela injeção salina (V_{ors}), que permitiu calcular a saturação de óleo recuperado pela injeção de solução salina (S_{ors}) para determinar o percentual de recuperação secundária, também conhecida como, convencional.

A saturação residual de óleo (S_{or}), dada pelo volume (V_{or}) foi calculada como:

$$S_{or(\%)} = \frac{V_{or}}{V_p} = \left(\frac{OOIP - V_{ors}}{V_p} \right) \times 100 \quad (3.14)$$

A saturação de óleo recuperado pela solução salina é dado por:

$$S_{ors(\%)} = S_{oi} - S_{or} = \frac{V_{ors}}{V_p} \times 100 \quad (3.15)$$

Dessa forma, calcula-se a saturação de óleo residual (S_{or}) como sendo:

$$S_{or(\%)} = S_{oi} - S_{or} = \left(\frac{OOIP - V_{ois}}{V_p} \right) \times 100 \quad (3.16)$$

Sendo o volume de óleo residual após a injeção salina (V_{or}) dado por:

$$V_{or} = OOIP - V_{ors} \quad (3.17)$$

• Procedimento de Injeção de Biossurfactante como método de recuperação avançada

O método de injeção de biossurfactante foi avaliado nesta etapa do teste. Um volume definido da formulação do biossurfactante foi injetado na amostra seguido por um volume definido da solução salina, na qual desloca o biossurfactante formando uma frente de avanço [Bonilla Sanabria et al., 2013]. O procedimento inclui os passos: (1) Abertura da linha de formulação de biossurfactante para permitir a entrada do mesmo no core holder; (2) Injeção de volume definido de biossurfactante na amostra; (3) Abertura da linha de solução salina simultaneamente ao fechamento da válvula da linha de formulação de biossurfactante para permitir a entrada da solução no core holder; (4) Injeção de volume definido de solução salina no plug. Foram realizadas duas injeções alternadas de biossurfactante e solução salina.

O fator de recuperação para a injeção de biossurfactante pode ser calculado a partir dos resultados desta etapa, assim como é possível fazer uma análise comparativa entre as respostas obtidas para a recuperação com injeção convencional de solução salina.

• **Determinação do Fator de Recuperação e avaliação dos ganhos obtidos no experimento de recuperação**

A eficiência na recuperação de óleo obtida pelos métodos de injeção de água e injeção de biossurfactante foi quantificada através do fator de recuperação. O fator de recuperação convencional (FR_{MC}) foi obtido pela injeção da salmoura (solução salina) no plug, obtendo-se a recuperação de certa quantidade de óleo. O FR_{MC} foi quantificado de acordo com as equações 3.18 e 3.19.

$$FR_{MC} = \frac{\text{VolumedeóleoRecuperadoporInjeçãodeSoluçãoSalina}}{\text{VolumedeÓleoInicialnoTestemunho}} \quad (3.18)$$

Que pode ser escrito em função dos volumes obtidos experimentalmente:

$$FR_{MC} = \frac{V_{ors}}{OOIP} \quad (3.19)$$

O método especial de recuperação utilizado teve o objetivo de recuperar uma fração do óleo residual por injeção da solução de biossurfactante. O fator de recuperação (FR_{ME}) neste caso foi obtido segundo as equações 3.20 e 3.21.

$$FR_{ME} = \frac{\text{VolumedeóleoRecuperadoporInjeçãodeBiossurfactante}}{\text{VolumedeÓleoInicialnoTestemunho}} \quad (3.20)$$

$$FR_{MC} = \frac{V_{orb}}{OOIP} \quad (3.21)$$

O Fator de Recuperação Geral pode ser obtido através da fórmula 3.22 abaixo:

$$FR_G = \frac{\text{VolumeTotaldeóleoRecuperado}}{\text{VolumedeÓleoInicialnoTestemunho}} \quad (3.22)$$

3.3 Diagrama de pacotes - assuntos

Com base na análise de domínio do software desenvolvido, foram identificados os seguintes pacotes:

- Meio poroso: é um pacote que possui os dados das propriedades da rocha e do fluido de saturação, que compõem o meio poroso. Sua função é fornecer estas propriedades para o modelo de recuperação;
- Permeabilidade absoluta: é um pacote que envolve todos os conceitos físicos que ocorrem no fluxo de fluido no meio poroso, onde a equação de Darcy, modela o fenômeno de transferência de fluido no meio poroso;
- Métodos matemáticos: é um pacote que reúne as funções matemáticas para cálculo das propriedades do meio poroso (porosidade, volume poroso, saturações, entre outros) e fatores de recuperação de óleo a partir da entrada de dados dos testes como, por exemplo, volume de água inicial (Vwi), volume de água produzida (Vwf), volume de óleo produzido na recuperação secundária (Vof1) e volume de óleo produzido na recuperação avançada (Vof2);
- Gráficos: é um pacote que utiliza o gnuplot para plotar a permeabilidade absoluta e volumes de óleo produzidos por minuto, a partir dos testes de fluxo em meio poroso.

A Figura 3.4 mostra o diagrama de pacotes para o software desenvolvido:

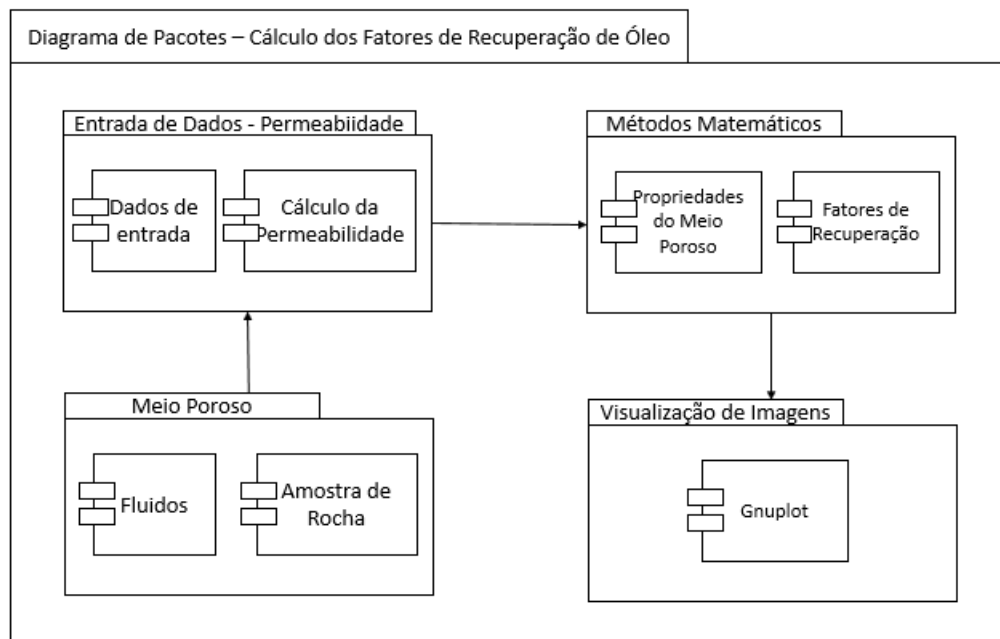


Figura 3.4: Diagrama de Pacotes – Assuntos

Capítulo 4

AOO - Análise Orientada a Objeto

Apresenta-se neste capítulo a Análise Orientada a Objeto - AOO, as relações entre as classes, os atributos, os métodos e suas associações. A análise consiste em modelos estruturais dos objetos e seus relacionamentos, e modelos dinâmicos, apresentando as modificações do objeto com o tempo. O resultado da análise é um conjunto de diagramas que identificam os objetos e seus relacionamentos.

4.1 Diagrama de Classes

O diagrama do software desenvolvido é composto por 4 classes e é apresentado na Figura 4.1.

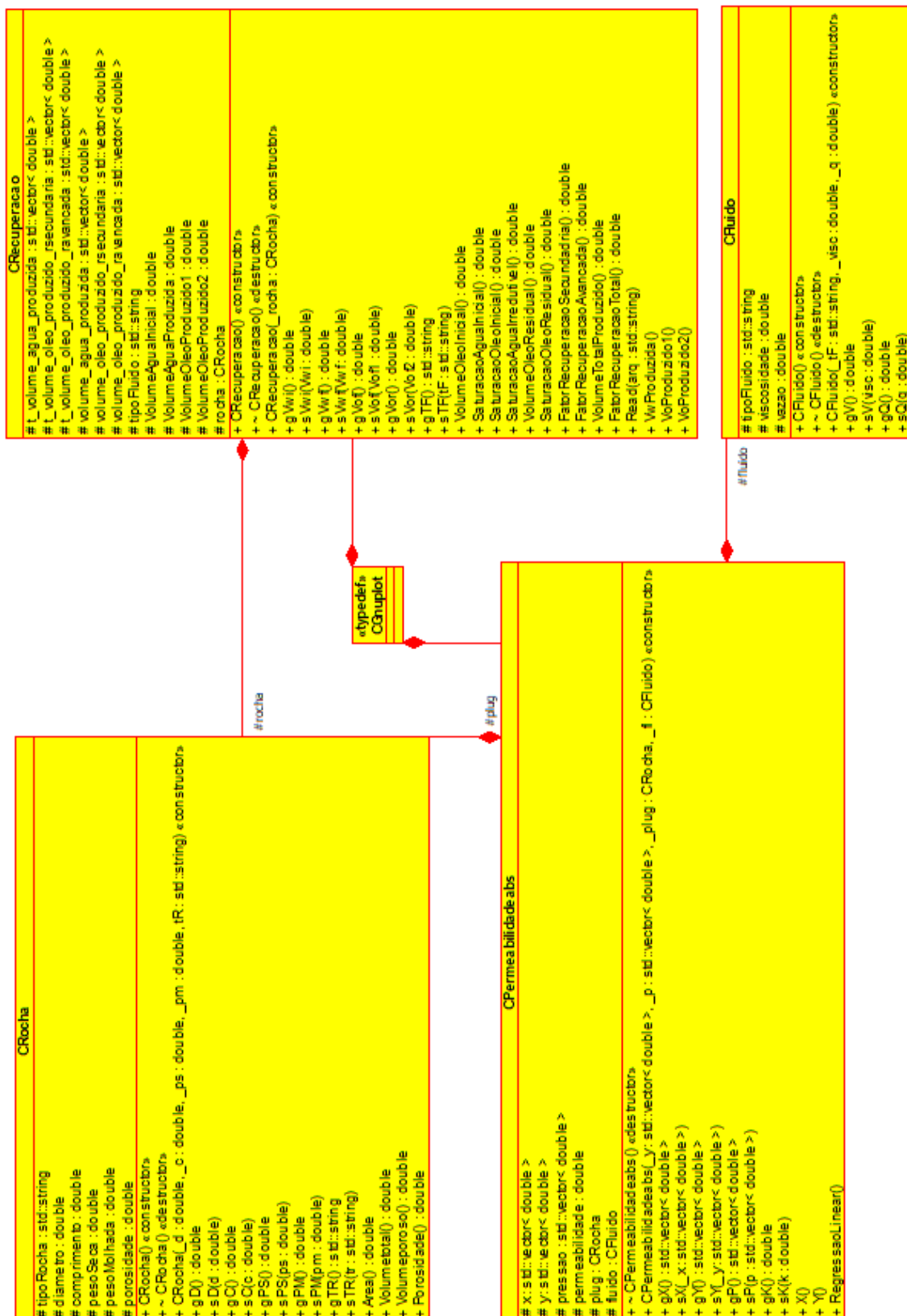


Figura 4.1: Diagrama de Classes

4.1.1 Dicionário de classes

- Classe CRocha: Uma classe que representa a matriz de um meio poroso, contendo suas propriedades físicas características. Nesta, calcula-se a área, o volume total, o volume poroso e a porosidade da amostra de rocha;
- Classe CFluido: Uma classe que representa as características dos fluidos injetados, contendo suas propriedades físico-químicas como viscosidade e vazão de injeção no meio poroso;
- Classe CPermeabilidadeabs: Representa a classe que irá calcular os parâmetros da permeabilidade absoluta à partir da Lei de Darcy (regressão linear);
- Classe CRecuperacao: Classe para cálculo dos fatores de recuperação de óleo à partir de métodos de recuperação secundária e avançada. Nesta classe, será importado um arquivo *.txt* que possibilita a geração dos gráficos de volumes produzidos por minuto. Serão calculados também, as saturações iniciais dos fluidos e volumes produzidos;
- Classe CGnuplot: Possibilita a geração de gráficos usando o software externo Gnuplot.

4.2 Diagrama de seqüência - eventos e mensagens

O diagrama de seqüência enfatiza a troca de eventos e mensagens e sua ordem temporal. Contém informações sobre o fluxo de controle do software.

4.2.1 Diagrama de seqüência geral

Costuma ser montado a partir de um diagrama de caso de uso e estabelece o relacionamento dos atores (usuários e sistemas externos) com alguns objetos do sistema. O diagrama de seqüência pode ser geral, englobando todas as operações do sistema ou específico, que enfatiza uma determinada operação. O diagrama de seqüência geral do software é mostrado na Figura 4.2.

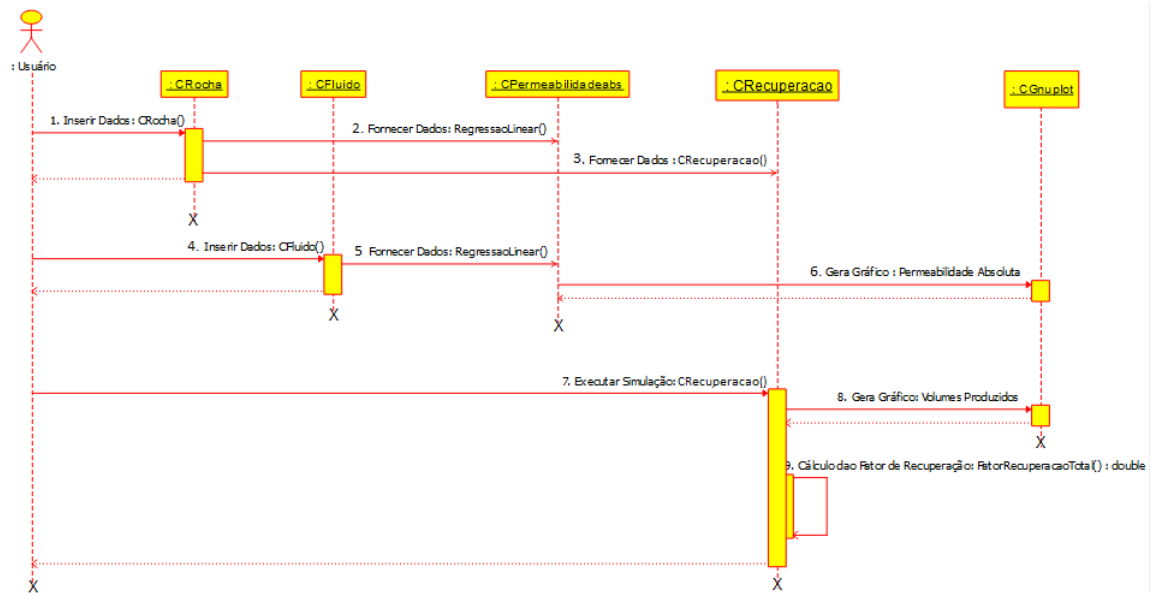


Figura 4.2: Diagrama de Sequências

4.3 Diagrama de comunicação - colaboração

No diagrama de comunicação o foco é a interação e a troca de mensagens e dados entre os objetos. Na Figura 4.3, o diagrama de comunicação mostra a sequência geral do software para o cálculo dos fatores de recuperação. Observe que as trocas de mensagens estão numeradas em sequência temporal.

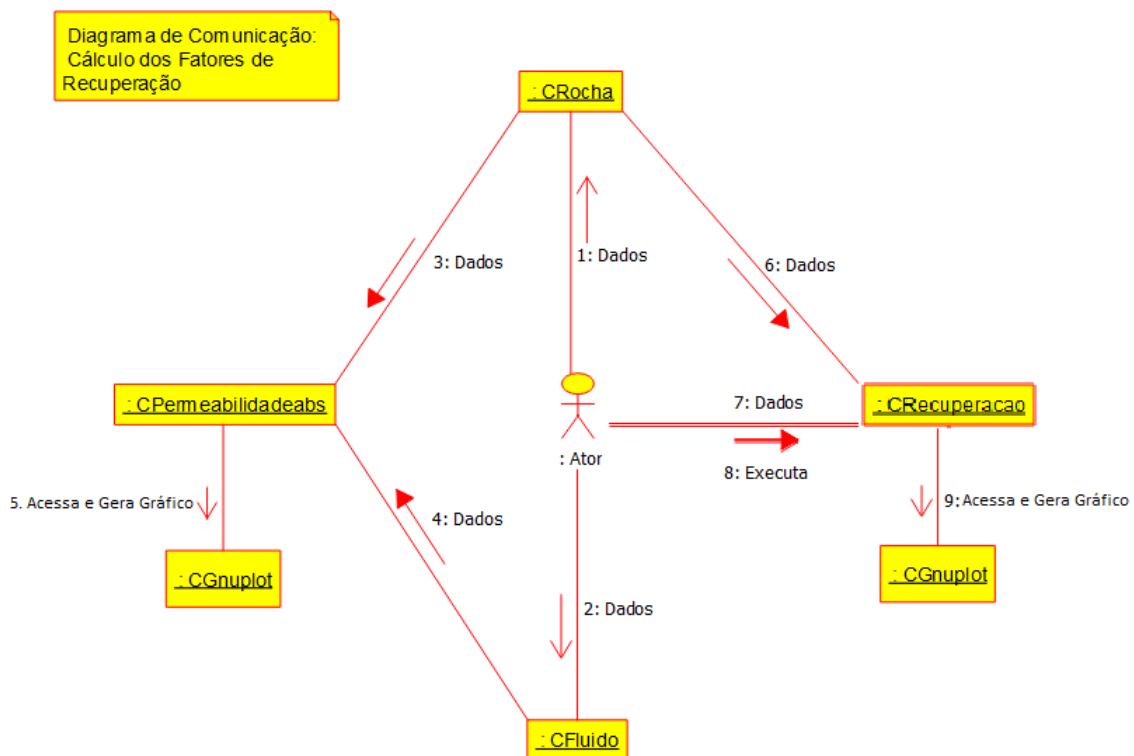


Figura 4.3: Diagrama de Comunicação

4.4 Diagrama de máquina de estado

Um diagrama de máquina 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 objeto. A Figura 4.4 demonstra os estados que a classe Crecuperacao apresenta durante a execução do programa.

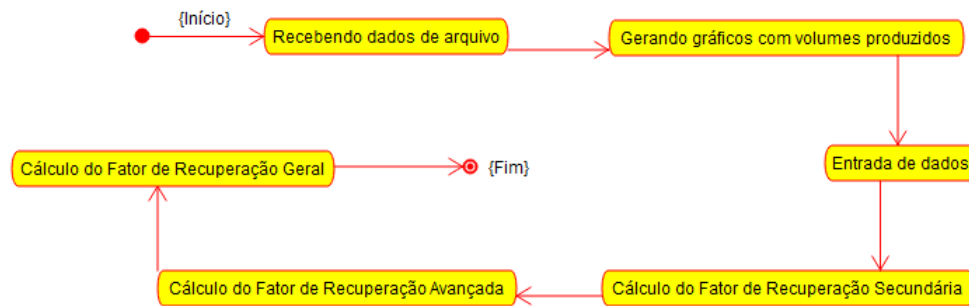


Figura 4.4: Diagrama de Máquina de Estado para a classe CRecuperacao

4.5 Diagrama de atividades

O diagrama de atividades (Figura 4.5) corresponde a uma atividade específica do diagrama de máquina de estado, onde calcula-se o Fator de Recuperação Geral.

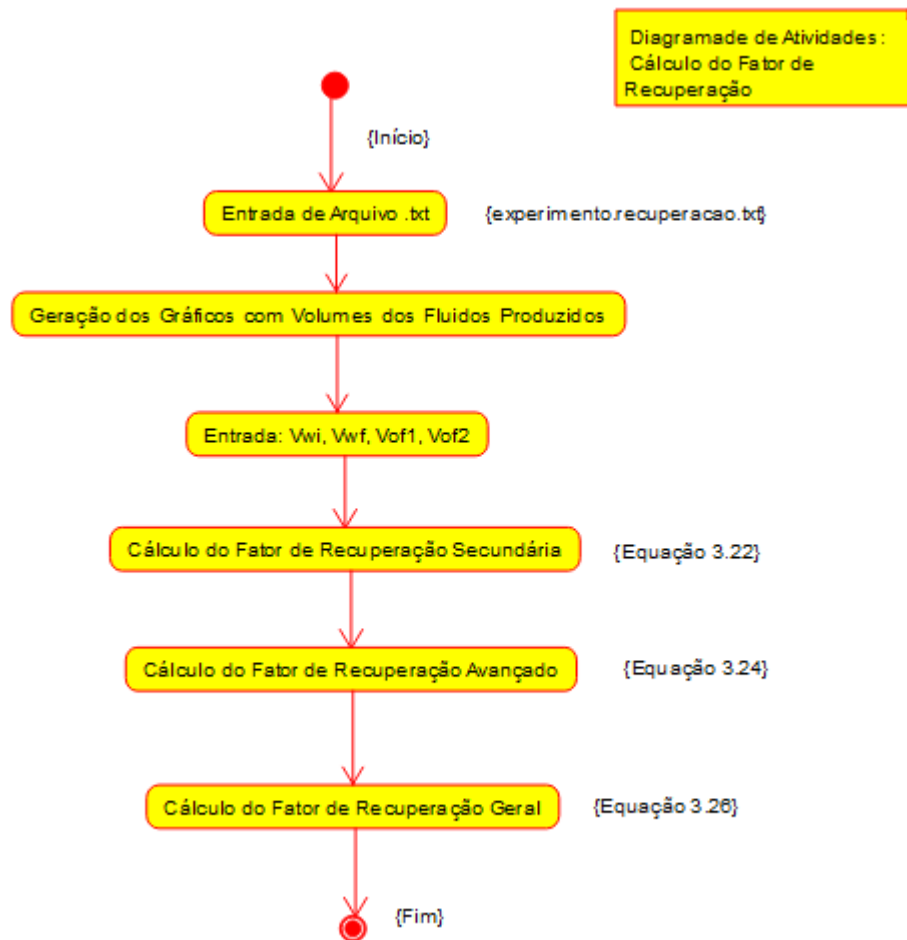


Figura 4.5: Diagrama de Atividades para o Método CRecupera-cao::FatorRecuperacaoTotal()

Capítulo 5

Projeto

Neste capítulo do projeto de engenharia serão apresentadas questões associadas ao projeto do sistema, incluindo protocolos, recursos, plataformas suportadas, implicações nos diagramas feitos anteriormente, diagramas de componentes e implantação. Avalia-se aqui algumas características do software como:

1. Protocolos

- O programa permite salvar imagens em disco no formato .png;

2. Plataformas

- O programa é multiplataforma, o que permite executá-lo em GNU/Linux, mas será desenvolvido na plataforma Windows;

3. Bibliotecas

- Biblioteca padrão C++;
- Programa externo Gnuplot;
- Programa externo Armadillo

3.1. Uso da Biblioteca Armadillo

Armadillo é uma biblioteca de software de álgebra linear para a linguagem de programação C++. O objetivo é fornecer cálculos de base eficientes e simplificados, enquanto ao mesmo tempo tem uma interface simples e fácil de usar. Neste projeto, a biblioteca foi utilizada para o cálculo da regressão linear a fim de determinar a permeabilidade absoluta da amostra de rocha. Abaixo, segue indicações de como instalar a biblioteca para utilização no *Windows* e *GNU/Linux*.

(a) No Windows:

- i. Entrar no site <http://arma.sourceforge.net/download.html> e fazer o *download* da versão mais atualizada;

- ii. Instalar os pacotes associados: *dnf install cmake openblas-devel lapack-devel arpack-devel SuperLU-devel*;
- iii. Entrar no Ambiente de Desenvolvimento Integrado;
- iv. Abrir o programa no *Visual Studio 2019*;
- v. Na aba “Ferramentas”, selecionar “Gerenciador de Pacotes NuGet”;
- vi. Selecionar “Gerenciador Pacotes do NuGet para a Solução...”;
- vii. Em “Procurar” digite o nome da biblioteca “Armadillo”, selecione e instale.

(b) No GNU/Linux:

- i. Entrar no site *http://arma.sourceforge.net/download.html* e fazer o *download* da versão mais atualizada;
- ii. Instalar os pacotes associados: *dnf install cmake openblas-devel lapack-devel arpack-devel SuperLU-devel*;
- iii. Baixar o arquivo do Armadillo: *http://sourceforge.net/projects/arma/files/armadillo-9.600.4.tar.xz*;
- iv. Descompactar;
- v. Entrar no diretório criado e configurar para a máquina que será utilizada (*./configure*);
- vi. - Compilar: Ler *http://arma.sourceforge.net/faq.html* e adicionar a flag *-larmadillo* para fazer a ligação do programa com a biblioteca do Armadillo;
- vii. Instalar como root para todos os usuários da máquina (*make install*);

4. Ambiente de desenvolvimento

- As listagens foram desenvolvidas com o software Microsoft Visual Studio, na plataforma Windows;
- Foi utilizado um arquivo .txt com os dados do experimento realizado em laboratório nomeado “experimento.recuperacao.txt”. Exemplo com o modelo da planilha é apresentado na Figura 5.1.

Embebição (Inj. De água) - Secundária						Drenagem (Inj. De óleo) - Secundária						Injeção de	
Biossurfactante (Banco) - Avançada													
T [min]	Vtotal [cm3]	Vw [cm3]	Vo [cm3]	P [psia]		T [min]	Vtotal [cm3]	Vw [cm3]	Vo [cm3]	P [psia]		Hora	Vtotal [cm3]
Vw [cm3]	Vo [cm3]	P [psia]				Vw [cm3]	Vo [cm3]	P [psia]					
0.0	0.0	0.0	0.0	0.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.7	
5.00	0.5	0.00	0.00	35.4	5.00	2.4	0.6	0.00	2.7	5.00	0.1	0.1	0.00
10.00	2.4	0.00	0.61	64.00	10.00	3.5	1.7	0.00	6.6	10.00	0.5	0.4	0.1
15.00	4.9	0.00	3.11	55.9	15.00	4.9	3.1	0.00	10.1	15.00	1.2	0.85	0.35
20.00	4.95	0.00	3.16	56.00	20.00	6.3	4.5	0.00	13.9	20.00	1.8	1.45	0.35
25.00	7.4	2.4	3.21	51.00	25.00	7.8	6.0	0.00	18.00	25.00	2.5	2.14	0.36
27.00	9.8	4.8	3.21	48.9	27.00	8.3	6.5	0.00	20.00	30.00	3.5	3.1	0.4
30.00	12.4	7.3	3.31	47.5	30.00	9.6	6.7	1.1	21.6	35.00	4.7	4.25	0.45
35.00	14.9	9.8	3.31	46.1	35.00	11.6	6.8	3.00	25.4	40.00	7.3	6.8	0.5
40.00	17.3	12.2	3.31	45.2	40.00	13.2	6.8	4.6	28.8	45.00	10.1	9.4	0.7
45.00	19.7	14.6	3.31	44.4	45.00	15.00	6.8	6.4	32.3	50.00	12.8	12.1	0.7
50.00	22.3	17.1	3.41	43.4	50.00	16.8	6.8	8.2	36.00	55.00	15.5	14.7	0.8
55.00	24.9	19.7	3.41	42.9	55.00	18.6	6.8	10.00	39.6	60.00	18.00	17.2	0.8
60.00	27.3	22.1	3.41	42.7	60.00	20.5	6.8	11.9	43.6	65.00	20.00	19.2	0.8
65.00	29.7	24.5	3.41	42.00	65.00	22.6	6.8	14.00	47.5	70.00	22.2	21.4	0.8
70.00	32.1	26.8	3.51	41.3	70.00	24.5	6.8	15.9	50.4	75.00	23.9	23.05	0.85
75.00	34.7	29.4	3.51	40.8	75.00	26.6	6.8	18.00	52.9	80.00	24.9	24.05	0.85
80.00	37.1	31.8	3.51	40.3	80.00	28.7	6.8	20.1	56.3	85.00	27.7	26.85	0.85
85.00	39.5	34.2	3.51	39.7	85.00	30.8	6.8	22.2	59.4	90.00	29.9	29.04	0.86
90.00	42.6	37.3	3.51	39.2	90.00	32.8	6.8	24.2	62.6	95.00	31.2	30.34	0.86
95.00	44.8	39.5	3.51	38.9	95.00	34.8	6.8	26.2	66.0	100.00	32.6	31.74	0.86
100.00	46.8	41.5	3.51	38.3	100.00	37.00	6.8	28.4	69.5	105.00	35.2	34.34	0.86
105.00	49.3	44.00	3.51	38.1	105.00	39.00	6.8	30.4	72.9	110.00	37.8	36.94	0.86

Figura 5.1: Diagrama de Implantação.

5.1 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. Além disso, envolve a otimização da estrutura de dados e dos algoritmos, a minimização do tempo de execução, de memória e de custos. Como o projeto não alterou os diagramas apresentados na análise orientada a objeto, não houve necessidade de descrever os itens abaixo relacionados:

- Efeitos do projeto no modelo estrutural;
- Efeitos do projeto no modelo dinâmico;
- Efeitos do projeto nos atributos;
- Efeitos do projeto nos métodos;
- Efeitos do projeto nas heranças;
- Efeitos do projeto nas associações;
- Efeitos do projeto nas otimizações.

5.2 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. Este 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. A Figura 5.2 mostra o diagrama de implantação utilizado.

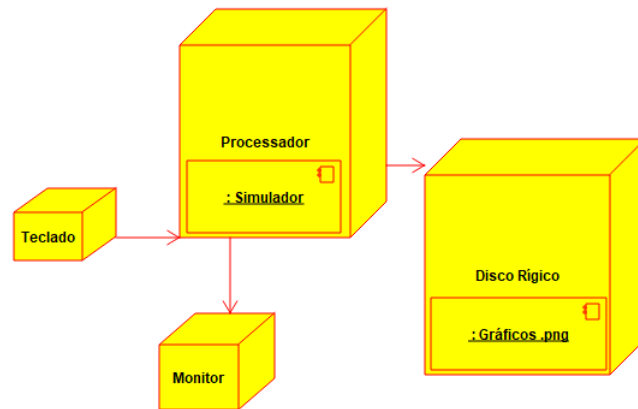


Figura 5.2: Diagrama de Implantação.

Capítulo 6

Implementação

Neste capítulo do projeto de engenharia apresenta-se os códigos fonte que foram desenvolvidos.

6.1 Código fonte

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

Apresenta-se na listagem 6.1 o arquivo com código da classe CRocha.

Listing 6.1: Arquivo de cabeçalho da classe CRocha.h

```
1 #pragma once
2 #ifndef CROCHA_H
3 #define CROCHA_H
4 #include <string>
5 #define PI 3.141592654
6
7
8 ///Representa as propriedades da rocha arenitica, Berea.
9 class CRocha
10 {
11 protected:
12     std::string tipoRocha;
13     double diametro;
14     double comprimento;
15     double pesoSeca;
16     double pesoMolhada;
17     double porosidade;
18
19 public:
20     CRocha() {};
21     ~CRocha() {};
22     ///Construtor Sobrecarregado
```

```

23         CRocha(std::string tR, double _d, double _c, double _ps, double
           _pm);
24
25         //Alterar ou acessar os valores dos parâmetros protegidos ou
           privados. Declaração dos métodos.
26         double gD();
27         void sD(double d);
28
29         double gC();
30         void sC(double c);
31
32         double gPS();
33         void sPS(double ps);
34
35         double gPM();
36         void sPM(double pm);
37
38         std::string gTR();
39         void sTR(std::string tr);
40
41         double Area();
42         double VolumeTotal();
43         double VolumePoroso();
44         double Porosidade();
45     };
46 #endif

```

Apresenta-se na listagem 6.2 o arquivo de implementação da classe CRocha.

Listing 6.2: Arquivo de implementação da classe CRocha.cpp

```

47 #include "CRocha.h"
48 #include <cmath>
49
50 CRocha::CRocha(std::string tR, double _d, double _c, double _ps, double
           _pm) {
51     tipoRocha = tR;
52     diametro = _d;
53     comprimento = _c;
54     pesoSeca = _ps;
55     pesoMolhada = _pm;
56     porosidade = Porosidade();
57 }
58
59 std::string CRocha::gTR() { return tipoRocha; }
60 void CRocha::sTR(std::string tr) { tipoRocha = tr; }
61
62 double CRocha::gD() { return diametro; }
63 void CRocha::sD(double d) { diametro = d; }
64

```

```

65 double CRocha::gC() { return comprimento; }
66 void CRocha::sC(double c) { comprimento = c; }
67
68 double CRocha::gPS() { return pesoSeca; }
69 void CRocha::sPS(double ps) { pesoSeca = ps; }
70
71 double CRocha::gPM() { return pesoMolhada; }
72 void CRocha::sPM(double pm) { pesoMolhada = pm; }
73
74 double CRocha::Area() {
75     return PI * (pow(diametro, 2) / 4.0) * 0.1; //0.1 Representa
        fator para conversão de unidades
76 }
77 double CRocha::VolumeTotal() {
78     return Area()* comprimento* 0.1; //0.1 Representa fator para
        conversão de unidades
79 }
80 double CRocha::VolumePoroso() {
81     return pesoMolhada - pesoSeca;
82 }
83 double CRocha::Porosidade() {
84     return VolumePoroso() / VolumeTotal();
85 }

```

Apresenta-se na listagem 6.3 o arquivo com código da classe CFluido.

Listing 6.3: Arquivo de cabeçalho da classe CFluido.h

```

86 #pragma once
87 #ifndef CFLUIDO_H
88 #define CFLUIDO_H
89
90 #include <string>
91 #include <iostream>
92
93 class CFluido
94 {
95 protected:
96     std::string tipoFluido;
97     double viscosidade;
98     double vazao;
99
100 public:
101     CFluido() {};
102     ~CFluido() {};
103     ///Construtor Sobrecarregado
104     CFluido(std::string _tF, double _visc, double _q);
105     double gV();
106     void sV(double visc);
107     double gQ();

```

```

108         void sQ(double q);
109 };
110 #endif

```

Apresenta-se na listagem 6.4 o arquivo de implementação da classe CFluido.

Listing 6.4: Arquivo de implementação da classe CFluido.cpp

```

111 #include "CFluido.h"
112
113 CFluido::CFluido(std::string _tF, double _visc, double _q) {
114     tipoFluido = _tF;
115     viscosidade = _visc;
116     vazao = _q;
117 }
118 double CFluido::gV() { return viscosidade; }
119 void CFluido::sV(double visc) { viscosidade = visc; }
120 double CFluido::gQ() { return vazao; }
121 void CFluido::sQ(double q) { vazao = q; }

```

Apresenta-se na listagem 6.5 o arquivo com código da classe CPermeabilidadeabs.

Listing 6.5: Arquivo de cabeçalho da classe CPermeabilidadeabs.h

```

122 #pragma once
123 #ifndef CPermeabilidadeabs_h
124 #define CPermeabilidadeabs_h
125 #include <string>
126 #include <iostream>
127 #include <vector>
128 #include "CRocha.h"
129 #include "CFluido.h"
130 #include <armadillo> //biblioteca para regressão linear
131 #include "CGnuplot.h"
132
133 //arma::Col p = arma::Col<float>(std::vector<float>y); conversão STD - >
134 ARMADILLO
135
136 class CPermeabilidadeabs
137 {
138     protected:
139         std::vector<double> x; //Calculado
140         std::vector<double> y; //Dados de entrada
141         std::vector<double> pressao; //Dados de entrada
142         double permeabilidade; //Calculado
143         CRocha amostra; //Dados de entrada - amostra
144         CFluido fluido; //Dados de entrada
145     public:
146         //CPermeabilidadeabs() {};
147         ~CPermeabilidadeabs() {};

```

```

148      ///Construtor Sobrecarregado
149      CPermeabilidadeabs(std::vector<double> _y, std::vector<double>
        _p, CRocha _amostra, CFluido _fl);
150      std::vector<double> gX() { return x; }
151      void sX(std::vector<double> _x) { x = _x; }
152
153      std::vector<double> gY() { return y; }
154      void sY(std::vector<double> _y) { y = _y; }
155
156      std::vector<double> gP() { return pressao; }
157      void sP(std::vector<double> p) { pressao = p; }
158
159      double gK() { return permeabilidade; }
160      void sK(double k) { permeabilidade = k; }
161
162      void X();
163      void Y();
164      void RegressaoLinear();
165 };
166 #endif

```

Apresenta-se na listagem 6.6 o arquivo de implementação da classe CPermeabilidadeabs.

Listing 6.6: Arquivo de implementação da classe CPermeabilidadeabs.cpp

```

167 #include "CPermeabilidadeabs.h"
168
169 CPermeabilidadeabs::CPermeabilidadeabs(std::vector<double> _y, std::
        vector<double> _p, CRocha _amostra, CFluido _fl) {
170     y = _y;
171     pressao = _p;
172     amostra = _amostra;
173     fluido = _fl;
174     X();
175     Y();
176 }
177 void CPermeabilidadeabs::X() {
178     if (y.size() != pressao.size()) { //verificando se o tamanho de
        y é igual da pressão.
179         std::cout << "Vetores de pressao e vazao de tamanhos
        diferentes" << '\n';
180     }
181     else {
182         x.resize(y.size()); //redimensionar
183         for (int i = 0; i < x.size(); ++i) {  //.size() - tamanho
        do vetor (quantidade de valores preenchidos)
184             x[i] = (amostra.Area() * pressao[i]) / (fluido.
        gV() * amostra.gC()) * (0.06804596);
185         std::cout << x[i] << " "; //Valor calculado
        aparece na tela

```

```

186         }
187     }
188     std::cout << '\n';
189 }
190 void CPermeabilidadeabs::Y() {
191     for (int j = 0; j < y.size(); j++) {
192         y[j] = y[j] * 16.66667; // conversão de unidade
193         std::cout << y[j] << "\n";
194     }
195 }
196 void CPermeabilidadeabs::RegressaoLinear() {
197     arma::Col<double> x1 = arma::Col<double>(x); //Conversão dos
        parâmetros x para Armadillo
198     arma::Col<double> y1 = arma::Col<double>(y); //Conversão dos
        parâmetros y para Armadillo
199     auto d = arma::polyfit(x1, y1, 1); //f = ax+b; d é o polinômio
        de ajuste
200     permeabilidade = d.eval()(0); // retirando o coeficiente A do
        pol. de ajuste
201     std::cout << "Permeabilidade_ absoluta_ do_ plug:_" <<
        permeabilidade << "_mD\n";
202     //float b = d.eval()(1); //retirando o coeficiente B do pol. de
        ajuste
203     //std::cout << permeabilidade << " " << d.eval()(1);
204
205     std::vector<double> x2; //criando
206     std::vector<double> y2; //criando
207     for (int i = 0; i < x.size(); i++) {
208         x2.push_back(x[i]);
209         y2.push_back(y[i]);
210     }
211     //Aparecer a equação da reta no gráfico - PlotVector não rodou.
212     std::string eqreta = "y_=" + std::to_string(permeabilidade) + "
        x_+" + std::to_string(d.eval()(1));
213     CGnuplot plot;
214     std::ofstream tmpfile("myfile4.txt");
215     for (int i = 0; i < x2.size(); i++) {
216         tmpfile << x2[i] << "___" << y2[i] << '\n';
217     }
218     tmpfile.close();
219     plot << "set_title \"Permeabilidade_Absoluta_-_BEREA6\"";
220     plot << "set_xlabel \"X\"";
221     plot << "set_ylabel \"Vazão\"";
222     plot << "plot 'myfile4.txt' title \"\" + eqreta + "\"_with_lines"
        ;
223     //plot << "set legend \"Series1\"";
224     std::cout << "Pressione uma tecla para continuar"; std::cin.get
        (); std::cout << '\n';

```

```

225         system("erase myfile4.txt");
226     }

```

Apresenta-se na listagem 6.7 o arquivo com código da classe CRecuperacao.

Listing 6.7: Arquivo de cabeçalho da classe CRecuperacao.h

```

229 #pragma once
230 #ifndef CRecuperacao_h
231 #define CRecuperacao_h
232 #include <string>
233 #include <iostream>
234 #include "CRocha.h"
235 #include <fstream>
236 #include <vector>
237 #include "CGnuplot.h"
238
239 class CRecuperacao
240 {
241 protected:
242     //typedef std::vector<double> vetor - apelido para o vetor.
243     std::vector<double> t_volume_agua_produzida;
244     std::vector<double> t_volume_oleo_produzido_rsecundaria;
245     std::vector<double> t_volume_oleo_produzido_ravancada;
246
247     std::vector<double> volume_agua_produzida;
248     std::vector<double> volume_oleo_produzido_rsecundaria;
249     std::vector<double> volume_oleo_produzido_ravancada;
250
251     std::string tipoFluido;
252     double VolumeAguaInicial;
253     double VolumeAguaProduzida;
254     double VolumeOleoProduzido1;
255     double VolumeOleoProduzido2;
256     CRocha rocha;
257
258 public:
259     CRecuperacao() {};
260     ~CRecuperacao() {};
261     ///Construtor Sobrecarregado
262     CRecuperacao(CRocha _rocha);
263
264     double gVwi();
265     void sVwi(double Vwi);
266
267     double gVwf();
268     void sVwf(double Vwf);
269
270     double gVof();
271     void sVof(double Vof1);

```

```

272
273     double gVor();
274     void sVor(double Vof2);
275
276     std::string gTF();
277     void sTF(std::string tF);
278
279     double VolumeOleoInicial();
280     double SaturacaoAguaInicial();
281     double SaturacaoOleoInicial();
282     double SaturacaoAguaIrredutivel();
283     double VolumeOleoResidual();
284     double SaturacaoOleoResidual();
285     double FatorRecuperacaoSecundadria();
286     double FatorRecuperacaoAvancada();
287     double VolumeTotalProduzido();
288     double FatorRecuperacaoTotal();
289     void Read(std::string arq); //método para ler o arquivo
290     void VwProduzida();
291     void VoProduzido1();
292     void VoProduzido2();
293 };
294 #endif

```

Apresenta-se na listagem 6.8 o arquivo de implementação da classe CRecuperacao.

Listing 6.8: Arquivo de implementação da classe CRecuperacao.cpp

```

296 #include "CRecuperacao.h"
297
298 CRecuperacao::CRecuperacao(CRocha _rocha) {
299     rocha = _rocha;
300 }
301
302 double CRecuperacao::gVwi() { return VolumeAguaInicial; }
303 void CRecuperacao::sVwi(double Vwi) { VolumeAguaInicial = Vwi; }
304
305 double CRecuperacao::gVwf() { return VolumeAguaProduzida; }
306 void CRecuperacao::sVwf(double Vwf) { VolumeAguaProduzida = Vwf; }
307
308 double CRecuperacao::gVof() { return VolumeOleoProduzido1; }
309 void CRecuperacao::sVof(double Vof1) { VolumeOleoProduzido1 = Vof1; }
310
311 double CRecuperacao::gVor() { return VolumeOleoProduzido2; }
312 void CRecuperacao::sVor(double Vof2) { VolumeOleoProduzido2 = Vof2; }
313
314 std::string CRecuperacao::gTF() { return tipoFluido; }
315 void CRecuperacao::sTF(std::string tF) { tipoFluido = tF; }
316
317 double CRecuperacao::VolumeOleoInicial() {

```



```

318         return VolumeAguaProduzida;
319     }
320     double CRecuperacao::SaturacaoAguaInicial() {
321         return (VolumeAguaInicial / rocha.VolumePoroso())*100.0;
322     }
323     double CRecuperacao::SaturacaoOleoInicial() {
324         return (VolumeOleoInicial() / rocha.VolumePoroso())*100.0;
325     }
326     double CRecuperacao::SaturacaoAguaIrredutivel() {
327         return ((VolumeAguaInicial - VolumeAguaProduzida) / rocha.
            VolumePoroso())*100.0;
328     }
329     double CRecuperacao::VolumeOleoResidual() {
330         return VolumeOleoInicial() - VolumeOleoProduzido1;
331     }
332     double CRecuperacao::SaturacaoOleoResidual() {
333         return (VolumeOleoResidual() / rocha.VolumePoroso())*100.0;
334     }
335     double CRecuperacao::FatorRecuperacaoSecundaria() {
336         return (VolumeOleoProduzido1 / VolumeOleoInicial())*100.0;
337     }
338     double CRecuperacao::FatorRecuperacaoAvancada() {
339         return (VolumeOleoProduzido2 / VolumeOleoInicial())*100.0;
340     }
341     double CRecuperacao::VolumeTotalProduzido() {
342         return VolumeOleoProduzido1 + VolumeOleoProduzido2;
343     }
344     double CRecuperacao::FatorRecuperacaoTotal() {
345         return (VolumeTotalProduzido() / VolumeOleoInicial())*100.0;
346     }
347     ///Leitura do arquivo de texto: "experimento.recuperacao.txt".
348     void CRecuperacao::Read(std::string arq) {
349         std::ifstream fin(arq);
350         if (!fin.good()) {
351             std::cerr << "Erro na leitura do arquivo";
352         }
353         double d;
354         fin.ignore(5000, '\n'); //Ignorando as primeiras linhas do
            arquivo .txt que contem cabecario
355         fin.ignore(5000, '\n');
356         while (!fin.eof()) {
357             fin >> d; //eu quero esse
358             t_volume_agua_produzida.push_back(d);
359             fin >> d;
360             fin >> d;
361             fin >> d;
362             volume_oleo_produzido_rsecundaria.push_back(d);
363             fin >> d;

```

```

364         fin >> d;
365         t_volume_oleo_produzido_rsecundaria.push_back(d);
366         fin >> d;
367         fin >> d;
368         volume_agua_produzida.push_back(d);
369         fin >> d;
370         fin >> d;
371         fin >> d;
372         t_volume_oleo_produzido_ravancada.push_back(d);
373         fin >> d;
374         fin >> d;
375         fin >> d;
376         volume_oleo_produzido_ravancada.push_back(d);
377         fin >> d;
378         fin.ignore(5000, '\n');
379     }
380     t_volume_agua_produzida.pop_back();
381     t_volume_oleo_produzido_rsecundaria.pop_back();
382     t_volume_oleo_produzido_ravancada.pop_back();
383     volume_agua_produzida.pop_back();
384     volume_oleo_produzido_rsecundaria.pop_back();
385     volume_oleo_produzido_ravancada.pop_back();
386     //fiz isso para tirar o 20.3
387 }
388 void CRecuperacao::VwProduzida() {
389     CGnuplot plot;
390     std::ofstream tmpfile("myfile1.txt");
391     for (int i = 0; i < t_volume_agua_produzida.size(); i++) {
392         tmpfile << t_volume_agua_produzida[i] << "          "
393             << volume_agua_produzida[i] << '\n';
394     }
395     tmpfile.close();
396     plot << "set title \"Vol. Água Produzida Acumulada - BEREÁ\"";
397     plot << "set xlabel \"Tempo (min)\"";
398     plot << "set ylabel \"Volume de Água (ml)\"";
399     plot << "plot 'myfile1.txt' with lines";
400     std::cout << "Pressione uma tecla para continuar"; std::cin.get
401     (); std::cout << '\n';
402     system("erase myfile1.txt");
403 }
404 void CRecuperacao::VoProduzido1() {
405     CGnuplot plot;
406     std::ofstream tmpfile("myfile2.txt");
407     for (int i = 0; i < t_volume_oleo_produzido_rsecundaria.size();
408         i++) {
409         tmpfile << t_volume_oleo_produzido_rsecundaria[i] << "    "
410             << volume_oleo_produzido_rsecundaria[i] << '\n';
411     }

```

```

408     tmpfile.close();
409     plot << "set_title \"Vol. Óleo Produzido por Recuperação
        Secundária\"";
410     plot << "set_xlabel \"Tempo (min)\"";
411     plot << "set_ylabel \"Volume de Óleo (ml)\"";
412     plot << "plot 'myfile2.txt' with lines";
413     std::cout << "Pressione uma tecla para continuar"; std::cin.get
        (); std::cout << '\n';
414     system("erase myfile2.txt");
415 }
416 void CRecuperacao::VoProduzido2() {
417     CGnuplot plot;
418     std::ofstream tmpfile("myfile3.txt");
419     for (int i = 0; i < t_volume_oleo_produzido_ravancada.size(); i
        ++){
420         //std::cout << t_volume_oleo_produzido_ravancada[i] << "
            " << volume_oleo_produzido_ravancada[i] << '\n';
421         tmpfile << t_volume_oleo_produzido_ravancada[i] << "
            " << volume_oleo_produzido_ravancada[i] << '\n';
422     }
423     tmpfile.close();
424     plot << "set_title \"Vol. Óleo Produzido por Recuperação
        Avançada\"";
425     plot << "set_xlabel \"Tempo (min)\"";
426     plot << "set_ylabel \"Volume de Óleo (ml)\"";
427     plot << "plot 'myfile3.txt' with lines";
428     std::cout << "Pressione uma tecla para continuar"; std::cin.get
        (); std::cout << '\n';
429     system("erase myfile3.txt");
430 }

```

Apresenta-se na listagem 6.9 o arquivo de implementação da classe CProjetoBuenoLaisa.

Listing 6.9: Arquivo de implementação da classe ProjetoBuenoLaisa.cpp

```

431 // ProjetoBuenoLaisa.cpp : Este arquivo contem a funcao 'main'.
432
433 #include <iostream>
434 #include <fstream>
435 #include "CFluido.h"
436 #include "CRocha.h"
437 #include "CPermeabilidadeabs.h"
438 #include "CGnuplot.h"
439 #include "CRecuperacao.h"
440 #include <string>
441
442 using namespace std;
443 int main()
444 {

```

```

445     std::cout << "
        #####\n
    ";
446     std::cout << "
        #####\n
    ";
447     std::cout << "LENEP/UENF-ProgramacaoPratica\n";
448     std::cout << "Calculo dos Fatores de Recuperacao de Oleo para
        Metodos EOR\n";
449     std::cout << "Teste de Fluxo em Meio Poroso-Recuperacao
        Avancada\n";
450     std::cout << "Nome: Laisa Rigueti Brasil\n";
451     std::cout << "
        #####\n
    ";
452     std::cout << "
        #####\n
    ";
453     std::cout << "\n";
454     double d1; double d2; double d3;
455     std::cout << "Entre com os dados da Rocha:\n";
456     std::cout << "Entre com tres medidas de diametro em mm:\n";
457     std::cout << "Diametro (1/3): "; cin >> d1; cin.get(); std::cout
        << "\n";
458     std::cout << "Diametro (2/3): "; cin >> d2; cin.get(); std::cout
        << "\n";
459     std::cout << "Diametro (3/3): "; cin >> d3; cin.get(); std::cout
        << "\n";
460     double d_medio = (d1 + d2 + d3) / 3;
461     double c1; double c2; double c3;
462     std::cout << "Entre com tres medidas de comprimento em mm:\n";
463     std::cout << "Comprimento (1/3): "; cin >> c1; cin.get(); std::
        cout << "\n";
464     std::cout << "Comprimento (2/3): "; cin >> c2; cin.get(); std::
        cout << "\n";
465     std::cout << "Comprimento (3/3): "; cin >> c3; cin.get(); std::
        cout << "\n";
466     double c_medio = (c1 + c2 + c3) / 3;
467     double p_mo;
468     std::cout << "\n";
469     std::cout << "Entre com o peso da amostra saturada com o fluido
        em g:\n";
470     std::cout << "Peso molhada: "; cin >> p_mo; cin.get(); std::cout
        << "\n";
471     double p_se;
472     std::cout << "Entre com o peso da amostra seca em g:\n";
473     std::cout << "Peso seca: "; cin >> p_se; cin.get(); std::cout <<
        "\n";

```

```

474     std::cout << "\n";
475     std::cout << "
        #####\n"
        ;
476     std::cout << "Resumo dos dados de entrada da amostra (plug):\n";
477     std::cout << "Diametro_medio:_" << d_medio << "_mm" << '\n';
478     std::cout << "Comprimento_medio:_" << c_medio << "_mm" << '\n';
479     std::cout << "Peso_molhada:_" << p_mo << "_g" << '\n';
480     std::cout << "Peso_seca_" << p_se << "_g" << '\n';
481     CRocha berea6("Arenito", d_medio, c_medio, p_se, p_mo);
        //Construtor Sobrecarregado
482     std::cout << "Area:_" << berea6.Area() << "_mm^2" << '\n';
483     std::cout << "Volume_Total:_" << berea6.VolumeTotal() << "_mm^3"
        << '\n';
484     std::cout << "Volume_Poroso:_" << berea6.VolumePoroso() << "_mL"
        << '\n';
485     std::cout << "Porosidade:_" << berea6.Porosidade() << '\n';
486     std::cout << "
        #####\n"
        ;
487     std::cout << "\n";
488
489     std::cout << "Calcularemos agora da Permeabilidade absoluta do
        meio_poroso:\n";
490     double visc; double q_inj;
491     std::cout << "Entre com os dados do Fluido.\n";
492     std::cout << "Entre com a vazao de injecao em mL/min\n";
493     std::cout << "Vazao de injecao:_" << q_inj << '\n';
494     std::cout << "Entre com a viscosidade do fluido em cP\n";
495     std::cout << "Viscosidade:_" << visc << '\n';
496
497     std::cout << "
        #####\n"
        ;
498     std::cout << "Resumo do dados de entrada do fluido\n";
499     std::cout << "Vazao de injecao:_" << q_inj << '\n';
500     std::cout << "Viscosidade:_" << visc << '\n';
501     std::cout << "
        #####\n"
        ;
502     CFluido brine("Brine", visc, q_inj);
503
504     std::cout << "Entre com as vazoes utilizadas no experimento de
        Perm.Absoluta para criacao do grafico\n";
505     double q1; double q2; double q3;
506     std::cout << "Entre com tres medidas de vazao em mL/min\n";

```

```

507     std::cout << "Vazao_(1/3):_"; cin >> q1; cin.get(); std::cout <<
        '\n';
508     std::cout << "Vazao_(2/3):_"; cin >> q2; cin.get(); std::cout <<
        '\n';
509     std::cout << "Vazao_(3/3):_"; cin >> q3; cin.get(); std::cout <<
        '\n';
510     std::cout << "\n";
511     std::cout << "Entre_com_as_pressoes_utilizadas_no_experimento_de
        _Perm.Absoluta_para_criacao_do_grafico\n";
512     double p1; double p2; double p3;
513     std::cout << "Entre_com_tres_medidas_de_pressao_em_psia\n";
514     std::cout << "Pressao_(1/3):_"; cin >> p1; cin.get(); std::cout
        << '\n';
515     std::cout << "Pressao_(2/3):_"; cin >> p2; cin.get(); std::cout
        << '\n';
516     std::cout << "Pressao_(3/3):_"; cin >> p3; cin.get(); std::cout
        << '\n';
517     std::vector<double> vaz = { q1,q2,q3 };
518     std::vector<double> pre = { p1,p2,p3 };
519     std::cout << "
        #####\n"
        ;
520     CPermeabilidadeabs exp1(vaz, pre, berea6, brine);
521     exp1.RegressaoLinear();
522     std::cout << "
        #####\n"
        ;
523     std::cout << "\n";
524
525     std::cout << "Calcularemos_agora_os_Fatores_de_Recuperacao_de_
        Oleo:\n";
526     CRecuperacao r1 (berea6);
527     std::string arq;
528     std::cout << "Entre_com_o_nome_do_arquivo_contendo_os_dados_do_
        experimento_de_Recuperacao_(.txt):_"; cin >> arq; cin.get();
        std::cout << '\n';
529     r1.Read(arq);
530     while (true) {
531         int resp;
532         std::cout << "Escolha_o_grafico_de_volume_produzido_a_
        ser_gerado:_\n";
533         std::cout << "1_-Volume_de_Agua_Produzida_Acumulada\n";
534         std::cout << "2_-Volume_de_Oleo_Recuperado_com_metodo_
        secundario\n";
535         std::cout << "3_-Volume_Adicional_de_Oleo_Recuperado_
        com_metodo_EOR\n";
536         std::cout << "4_-Sair_da_selecao_do_grafico\n";
537         cin >> resp; cin.get();

```

```

538         if (resp == 1) {
539             r1.VwProduzida();
540         }
541         else if (resp == 2) {
542             r1.VoProduzido1();
543         }
544         else if (resp == 3) {
545             r1.VoProduzido2();
546         }
547         else if (resp == 4) break;
548         else std::cout << "Entrada_invalida\n";
549     }
550     double aux;
551     std::cout << "Entre_com_os_dados_da_Recuperacao:\n";
552     std::cout << "Volume_de_Agua_Inicial_(Recuperacao_Secundaria):_";
553     ; cin >> aux; cin.get(); r1.sVwi(aux); std::cout << '\n';
554     std::cout << "Volume_de_Agua_Produzida_(Recuperacao_Secundaria):_";
555     cin >> aux; cin.get(); r1.sVwf(aux); std::cout << '\n';
556     std::cout << "Volume_de_Oleo_Produzido_(Recuperacao_Secundaria):_";
557     cin >> aux; cin.get(); r1.sVof(aux); std::cout << '\n';
558     std::cout << "Volume_de_Oleo_Produzido_(Recuperacao_Avancada):_"
559     ; cin >> aux; cin.get(); r1.sVor(aux); std::cout << '\n';
560
561     std::cout << "Dados_da_Recuperacao\n";
562     std::cout << "Volume_de_Oleo_Inicial:_" << r1.VolumeOleoInicial
563     () << "mL" << '\n';
564     std::cout << "Saturacao_de_Oleo_Inicial:_" << r1.
565     SaturacaoOleoInicial() << "%" << '\n';
566     std::cout << "Saturacao_de_Agua_Irredutivel:_" << r1.
567     SaturacaoAguaIrredutivel() << "%" << '\n';
568     std::cout << "Fator_Recuperacao_Secundaria:_" << r1.
569     FatorRecuperacaoSecundadria() << "%" << '\n';
570
571     std::cout << "Saturacao_de_Oleo_Residual:_" << r1.
572     SaturacaoOleoResidual() << "%" << '\n';
573     std::cout << "Volume_de_Oleo_Residual:_" << r1.
574     VolumeOleoResidual() << "mL" << '\n';
575     std::cout << "Fator_Recuperacao_Avancada:_" << r1.
576     FatorRecuperacaoAvancada() << "%" << '\n';
577
578     std::cout << "Fator_Recuperacao_Total:_" << r1.
579     FatorRecuperacaoTotal() << "%" << '\n';
580
581     std::ofstream out; //
582     std::ofstream fout("Resultados.txt");
583     fout << "Resultados_do_Software:\n";
584     fout << "\n";
585     fout << "

```

```

#####
n";
574 fout << "Resumo_dos_dados_de_entrada_da_amostra_(plug):\n";
575 fout << "Diametro_medio:_ " << d_medio << "_mm" << '\n';
576 fout << "Comprimento_medio:_ " << c_medio << "_mm" << '\n';
577 fout << "Peso_molhada:_ " << p_mo << "_g" << '\n';
578 fout << "Peso_seca_" << p_se << "_g" << '\n';
579 fout << "
#####
n";
580 fout << "Resumo_das_propriedades_do_meio_poroso_(plug):\n";
581 fout << "Area:_ " << berea6.Area() << "_mm^2" << '\n';
582 fout << "Volume_Total:_ " << berea6.VolumeTotal() << "_mm^3" << '
\n';
583 fout << "Volume_Poroso:_ " << berea6.VolumePoroso() << "_mL" << '
\n';
584 fout << "Porosidade:_ " << berea6.Porosidade() << '\n';
585 fout << "
#####
n";
586 fout << "Resultado_dos_Fatores_de_Recuperacao_Ã_partir_de_
Testes_de_Fluxo_em_Meio_Poroso:\n";
587 fout << "Volume_de_Oleo_Inicial:_ " << r1.VolumeOleoInicial() <<
"_mL" << '\n';
588 fout << "Saturacao_de_Oleo_Inicial:_ " << r1.SaturacaoOleoInicial
() << "%" << '\n';
589 fout << "Saturacao_de_Agua_Irredutivel:_ " << r1.
SaturacaoAguaIrredutivel() << "%" << '\n';
590 fout << "Fator_Recuperacao_Secundaria:_ " << r1.
FatorRecuperacaoSecundadria() << "%" << '\n';
591 fout << "Saturacao_de_Oleo_Residual:_ " << r1.
SaturacaoOleoResidual() << "%" << '\n';
592 fout << "Volume_de_Oleo_Residual:_ " << r1.VolumeOleoResidual()
<< "_mL" << '\n';
593 fout << "Fator_Recuperacao_Avancada:_ " << r1.
FatorRecuperacaoAvancada() << "%" << '\n';
594 fout << "\n";
595 fout << "Fator_Recuperacao_Total:_ " << r1.FatorRecuperacaoTotal
() << "%" << '\n';
596 out.close();
597
598 std::cout << "Pressione_uma_tecla_para_continuar"; std::cin.get
(); std::cout << '\n';
599
600 }

```


Capítulo 7

Teste

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

Como base, foi utilizado um estudo realizado no laboratório de Fluidos do Lenep sobre o uso de um biossurfactante produzido por microbiota de manguezais fornecido pelo centro de pesquisa CPQBA – UNICAMP como fluido de injeção para recuperação avançada de petróleo. A formulação foi direcionada com escolha adequada dos parâmetros petrofísicos com o intuito de se obter um maior fator de recuperação.

O presente trabalho apresenta interface em modo texto. Veja na Figura 7.2 a tela inicial do software. Os dados iniciais de entrada para cálculo das propriedades do meio poroso são incluídos para início do programa:

PARÂMETRO	MEDIDA	UNIDADE
Diâmetro (1/3)	38.300	mm
Diâmetro (2/3)	37.800	mm
Diâmetro (3/3)	38.290	mm
Comprimento (1/3)	41.740	mm
Comprimento (2/3)	41.690	mm
Comprimento (3/3)	41.420	mm
Peso Amostra Seca	100.409	g
Peso Amostra Molhada	110.235	g

Figura 7.1: Dados de entrada.

```
#####
#####
LENEP/UENF - Programacao Pratica
Calculo dos Fatores de Recuperacao de Oleo para Metodos EOR
Teste de Fluxo em Meio Poroso - Recuperacao Avancada
Nome: Laisa Rigueti Brasil
#####
#####

Entre com os dados da Rocha:
Entre com tres medidas de diametro em mm:
Diametro (1/3): 38.3

Diametro (2/3): 37.8

Diametro (3/3): 38.29

Entre com tres medidas de comprimento em mm:
Comprimento (1/3): 41.74

Comprimento (2/3): 41.69

Comprimento (3/3): 41.42

Entre com o peso da amostra saturada com o fluido em g:
Peso molhada: 110.235

Entre com o peso da amostra seca em g:
Peso seca: 100.409
```

Figura 7.2: Entrada de Dados Iniciais.

O programa retornará as propriedades físicas e parâmetros da amostra:

```
#####
Resumo dos dados de entrada da amostra (plug):
Diametro medio: 38.13 mm
Comprimento medio: 41.6167 mm
Peso molhada: 110.235 g
Peso seca 100.409 g
Area: 114.189 mm^2
Volume Total: 475.216 mm^3
Volume Poroso: 9.826 mL
Porosidade: 0.0206769
#####
```

Figura 7.3: Retorno após entrada de dados iniciais da amostra de rocha.

Após cálculo das propriedades, é iniciado o método para cálculo da permeabilidade absoluta da amostra. O usuário entra com os dados do fluido para teste de injeção para recuperação e com os dados medidos em laboratório para criação da reta da permeabilidade absoluta através da Lei de Darcy. Os dados de entrada para cálculo das permeabilidade do meio poroso é indicado abaixo:

Vazão (cm ³ /min)	Diferencial de Pressão (psia)
8	5.7
10	7.5
12	9.3

Figura 7.4: Dados de entrada.

```

#####
Calcularemos agora da Permeabilidade absoluta do meio poroso:
Entre com os dados do Fluido.
Entre com a vazao de injecao em mL/min
Vazao de injecao: 0.5

Entre com a viscosidade do fluido em cP
Viscosidade: 1.0

#####
Resumo do dados de entrada do fluido
Vazao de injecao: 0.5
Viscosidade: 1
#####
BrineEntre com as vazoes utilizadas no experimento de Perm.Absoluta para criacao do grafico
Entre com tres medidas de vazao em mL/min
Vazao (1/3): 8

Vazao (2/3): 10

Vazao (3/3): 12

Entre com as pressoes utilizadas no experimento de Perm.Absoluta para criacao do grafico
Entre com tres medidas de pressao em psia
Pressao (1/3): 5.7

Pressao (2/3): 7.5

Pressao (3/3): 9.3

#####
Dentro do metodo X()
1.06422 1.4003 1.73637
133.333
166.667
200
Permeabilidade absoluta do plug: 99.1854 mD
Pressione qualquer tecla para continuar. . .

```

Figura 7.5: Retorno após entrada de dados da amostra de rocha.

O Gráfico da Permeabilidade é gerado:

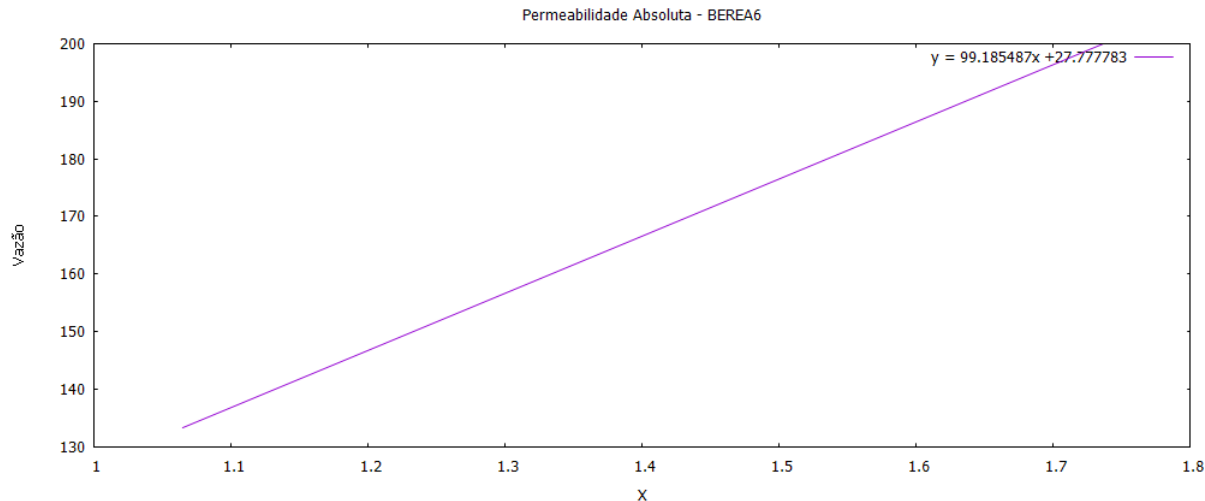


Figura 7.6: Gráfico da Permeabilidade Absoluta.

Após esta etapa, inicia-se o cálculo do fatores de recuperação. O programa solicita o arquivo .txt que contém os dados do experimento. O usuário escolhe o tipo de gráfico que deseja visualizar de acordo com os volumes produzidos durante os testes de injeção.

```
#####
Calcularemos agora os Fatores de Recuperacao de Oleo:
Entre com o nome do arquivo contendo os dados do experimento de Recuperacao (.txt): experimento.recuperacao.txt

Escolha o gráfico de volume produzido a ser gerado:
1 - Volume de Agua Produzida Acumulada
2 - Volume de Oleo Recuperado com metodo secundario
3 - Volume Adicional de Oleo Recuperado com metodo EOR
4 - Sair da selecao do grafico
1
Pressione qualquer tecla para continuar. . .
```

Figura 7.7: Entrada de arquivo e escolha do gráfico dos volumes produzidos.

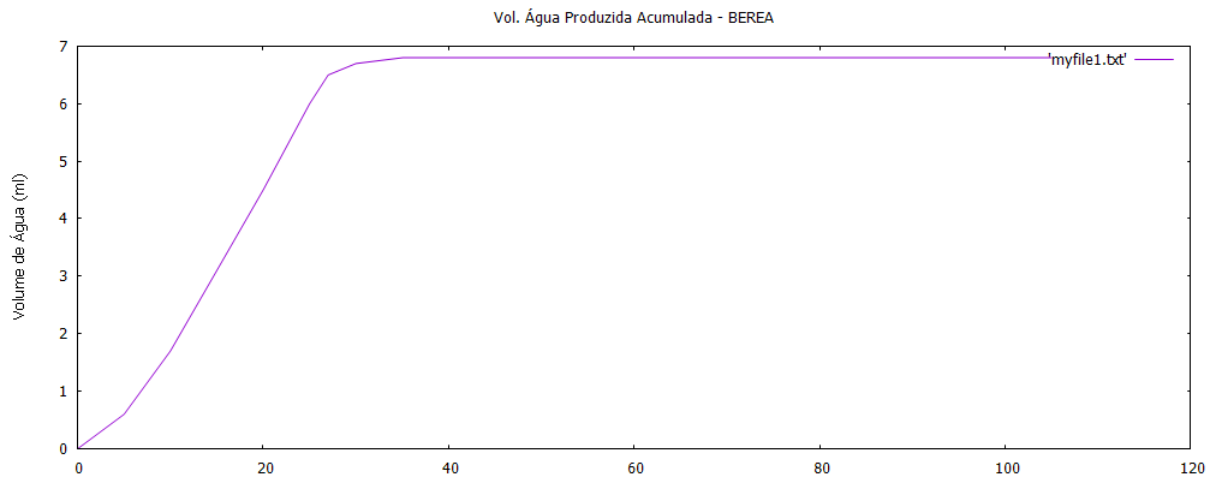


Figura 7.8: Gráfico Opção 1.

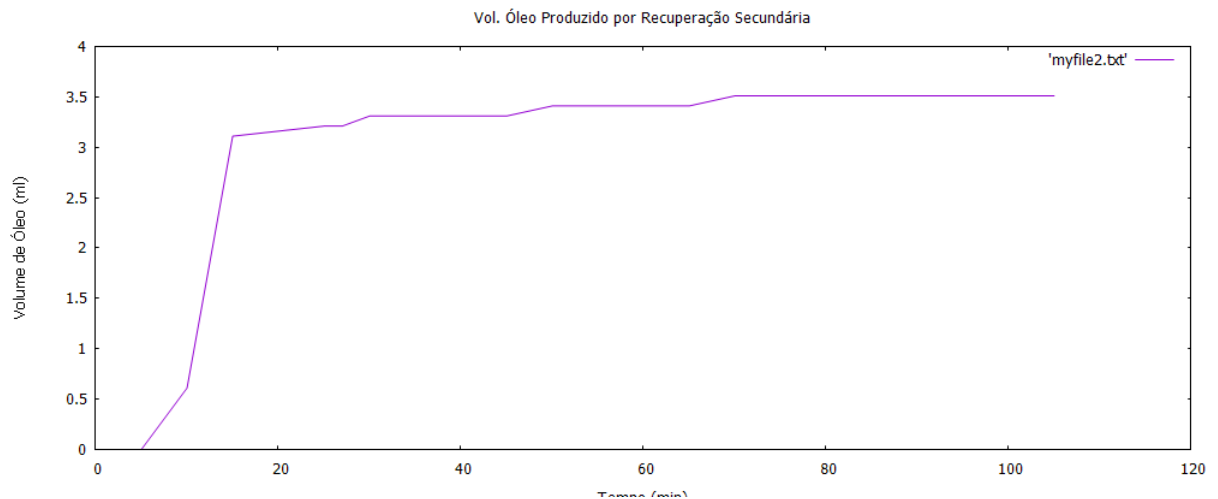


Figura 7.9: Gráfico Opção 2.

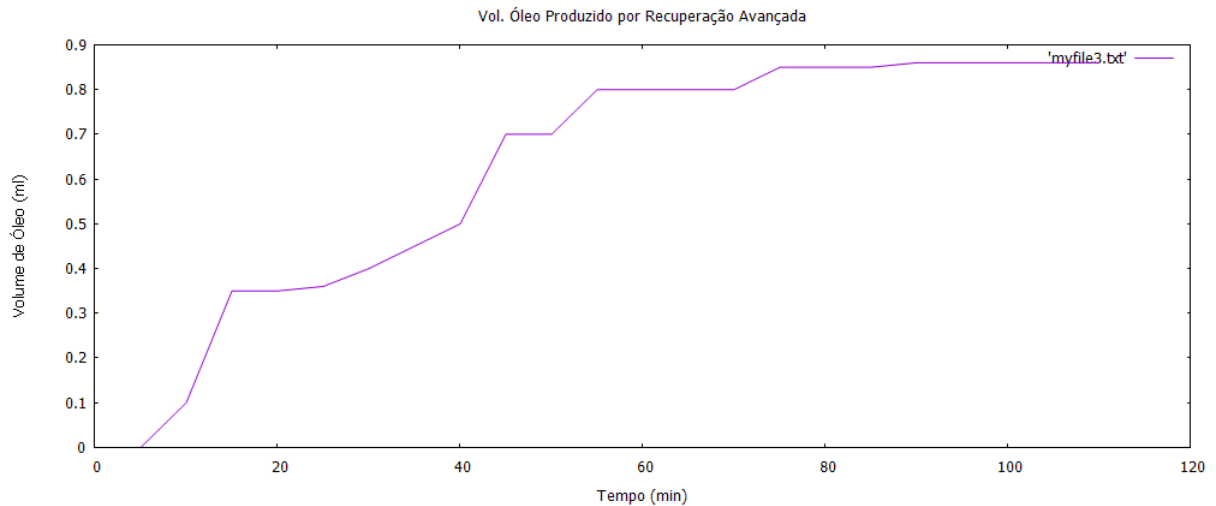


Figura 7.10: Gráfico Opção 3.

Após visualização dos gráficos gerados e análise dos resultados de volumes produzidos, é possível calcular os fatores de recuperação. Para isso, o usuário deverá inserir os volumes produzidos.

```
4 - Sair da selecao do grafico
4
Entre com os dados da Recuperacao:
Volume de Agua Inicial (Recuperacao Secundaria): 9.826

Volume de Agua Produzida (Recuperacao Secundaria): 6.81
Volume de Oleo Produzido (Recuperacao Secundaria): 3.51
Volume de Oleo Produzido (Recuperacao Avancada): 0.86

Dados da Recuperacao
Volume de Oleo Inicial: 6.81 mL
Saturacao de Oleo Inicial: 69.3059 %
Saturacao de Agua Irredutivel: 30.6941 %
Fator Recuperacao Secundaria: 51.5419 %
Saturacao de Oleo Residual: 33.5844 %
Volume de Oleo Residual: 3.3 mL
Fator Recuperacao Avancada: 12.6285 %
Fator Recuperacao Total: 64.1703 %
Pressione qualquer tecla para continuar. . .
```

Figura 7.11: Entrada de dados dos volumes produzidos e retorno dos fatores de recuperação.

Todas as informações necessárias para análise dos resultados e os fatores de recuperação são retornados ao usuário como forma de texto. Um arquivo .txt (com o nome Resultados) é salvo no diretório do projeto. A figura 24 mostra o arquivo salvo com o resumo dos dados.

```
Resultados do Software:

#####
Resumo dos dados de entrada da amostra (plug):
Diametro medio: 38.2 mm
Comprimento medio: 41.6167 mm
Peso molhada: 110.235 g
Peso seca 100.409 g
#####
Resumo das propriedades do meio poroso (plug):
Area: 114.608 mm^2
Volume Total: 476.962 mm^3
Volume Poroso: 9.826 mL
Porosidade: 0.0206012
#####
Resultado dos Fatores de Recuperacao à partir de Testes de Fluxo em Meio Poroso:
Volume de Oleo Inicial: 6.81 mL
Saturacao de Oleo Inicial: 69.3059 %
Saturacao de Agua Irredutivel: 30.6941 %
Fator Recuperacao Secundaria: 51.5419 %
Saturacao de Oleo Residual: 33.5844 %
Volume de Oleo Residual: 3.3 mL
Fator Recuperacao Avancada: 12.6285 %

Fator Recuperacao Total: 64.1703 %
```

Figura 7.12: Arquivo com resultados salvos em .txt no diretório do software.

Capítulo 8

Documentação

Todo projeto de engenharia precisa ser bem documentado. Neste sentido, apresenta-se neste capítulo a documentação de uso do software criado. Esta documentação tem o formato de uma apostila que explica passo a passo como usar o software.

8.1 Documentação do usuário

Descreve-se aqui o manual do usuário, um guia que explica, passo a passo a forma de instalação e uso do software desenvolvido.

8.1.1 Como instalar o software

Abra o terminal, vá para o diretório onde está o projeto, compile o programa e, depois o execute. Logo após, siga os seguintes passos:

1. Entre com os seguintes dados:
 - Três medidas de diâmetro medidas com o auxílio de um paquímetro em laboratório;
 - Três medidas de comprimento medidas com o auxílio de um paquímetro em laboratório;
 - Peso da amostra saturada (molhada) medido com o auxílio de uma balança de precisão em laboratório;
 - Peso da amostra seca medido com o auxílio de uma balança de precisão em laboratório.
2. Após isso serão mostrados no console os resultados das propriedades do meio poroso;
3. Entre com os seguintes dados:
 - Vazão de injeção adotado para o teste de recuperação;

- Viscosidade do fluido que medirá a permeabilidade absoluta;
 - Três medidas de vazão utilizadas no teste para análise da permeabilidade;
 - Três pressões adquiridas conforme vazões adotadas no teste para análise da permeabilidade.
4. Após isso serão mostrados no console o resultado da permeabilidade absoluta da amostra e gráfico da reta da permeabilidade;
 5. Entre com os seguintes dados:
 - Nome do arquivo .txt contendo os dados medidos no laboratório para os testes de recuperação – Volume recuperado por minuto
 6. Após isso serão mostrados no console as opções para visualização dos gráficos com os volumes produzidos em cada etapa de recuperação. O usuário poderá selecionar o gráfico para fazer suas análises;
 7. Entre com os seguintes dados após análise do volume produzido:
 - Volume inicial de água na amostra (igual volume poroso);
 - Volume de água produzida na recuperação secundária;
 - Volume de óleo produzido na recuperação secundária;
 - Volume de óleo produzido na recuperação avançada.
 8. Após isso serão mostrados no console os fatores de recuperação da amostra.

8.2 Documentação para desenvolvedor

Apresenta-se nesta seção a documentação para o desenvolvedor, isto é, informações para usuários que queiram modificar, aperfeiçoar ou ampliar este software.

8.2.1 Dependências

Para compilar o software é necessário atender as seguintes dependências:

- No sistema operacional Windows:
 - Instalar o compilador apropriado;
 - Recomenda-se o Microsoft Visual Studio 2019.
- No sistema operacional GNU/Linux:
 - Instalar o compilador g++ da GNU disponível em <http://gcc.gnu.org>.

- Para instalar no GNU/Linux use o comando `yum install gcc`.
- 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`.
- Biblioteca `CGnuplot`; os arquivos para acesso a biblioteca `CGnuplot` devem estar no diretório com os códigos do software.

8.2.2 Como gerar a documentação usando doxygen

A documentação do código do software deve ser feita usando o padrão JAVADOC, 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` (documentação em anexo neste documento).



Figura 8.1: Documentação criada no Doxygen.

Referências Bibliográficas

- [Agbalaka et al., 2008] Agbalaka, C. C., Dandekar, A. Y., Patil, S. L., Khataniar, S., Hemsath, J., et al. (2008). The effect of wettability on oil recovery: A review. In *SPE Asia Pacific Oil and Gas Conference and Exhibition*. Society of Petroleum Engineers. 11
- [Albuquerque et al., 2007] Albuquerque, H. S., Vale, T. Y. F., Dantas, T. d. C., Avelino de Neto, A., Santanna, V. C., and Coelho, T. A. M. (2007). Estudo da eficiência de sistemas microemulsionados na recuperação avançada de petróleo. *Anais do 4^o PDPETRO*. 11
- [Ali et al., 1996] Ali, S., Thomas, S., et al. (1996). The promise and problems of enhanced oil recovery methods. *Journal of Canadian Petroleum Technology*, 35(07). 8
- [Bonilla Sanabria et al., 2013] Bonilla Sanabria, F. C. et al. (2013). Avaliação da injeção de surfactantes como método de recuperação avançada em reservatórios de arenito. 18
- [Bruce et al., 1947] Bruce, W., Welge, H., et al. (1947). The restored-state method for determination of oil in place and connate water. In *Drilling and Production Practice*. American Petroleum Institute. 8
- [Buckley, 1999] Buckley, J. S. (1999). Evaluation of reservoir wettability and its effect on oil recovery. 11
- [Carrero et al., 2007] Carrero, E., Queipo, N. V., Pintos, S., and Zerpa, L. E. (2007). Global sensitivity analysis of alkali–surfactant–polymer enhanced oil recovery processes. *Journal of Petroleum Science and Engineering*, 58(1-2):30–42. 8
- [Dandekar, 2013] Dandekar, A. Y. (2013). *Petroleum reservoir rock and fluid properties*. CRC press. 11, 13, 15
- [Donaldson et al., 1985] Donaldson, E. C., Chilingarian, G. V., and Yen, T. F. (1985). *Enhanced oil recovery, I: fundamentals and analyses*. Elsevier. 11, 13
- [Engelke, 2012] Engelke, B. S. (2012). *Determinação das Curvas de Permeabilidade Relativa no Escoamento de Emulsões e Óleo*. PhD thesis, PUC-Rio. 17

- [Green et al., 1998] Green, D. W., Willhite, G. P., et al. (1998). *Enhanced oil recovery*, volume 6. Henry L. Doherty Memorial Fund of AIME, Society of Petroleum Engineers Richardson, TX. 9
- [Lake et al., 1989] Lake, L. W. et al. (1989). Enhanced oil recovery. 8, 11, 12
- [Lazar et al., 2007] Lazar, I., Petrisor, I., and Yen, T. (2007). Microbial enhanced oil recovery (meor). *Petroleum Science and Technology*, 25(11):1353–1366. 8
- [Mothé et al., 2006] Mothé, C., Correia, D., De Franca, F., and Riga, A. (2006). Thermal and rheological study of polysaccharides for enhanced oil recovery. *Journal of thermal analysis and calorimetry*, 85(1):31–36. 7, 8
- [Muggeridge et al., 2014] Muggeridge, A., Cockin, A., Webb, K., Frampton, H., Collins, I., Moulds, T., and Salino, P. (2014). Recovery rates, enhanced oil recovery and technological limits. *Phil. Trans. R. Soc. A*, 372(2006):20120320. 8
- [Núñez, 2011] Núñez, V. R. G. (2011). *Análise do Processo de Recuperação de Óleo por Injeção de Emulsão*. PhD thesis, PUC-Rio. 13
- [Portwood et al., 1995] Portwood, J. et al. (1995). A commercial microbial enhanced oil recovery technology: evaluation of 322 projects. In *SPE Production Operations Symposium*. Society of Petroleum Engineers. 9
- [Rosa et al., 2006] Rosa, A. J., de Souza Carvalho, R., and Xavier, J. A. D. (2006). *Engenharia de reservatórios de petróleo*. Interciência. 7, 8, 10, 11, 12, 13
- [Sen, 2008] Sen, R. (2008). Biotechnology in petroleum recovery: the microbial eor. *Progress in energy and combustion Science*, 34(6):714–724. 9
- [Thomas, 2001] Thomas, J. E. (2001). *Fundamentos de engenharia de petróleo*. Interciência. 7, 8, 10
- [Venuto, 1989] Venuto, P. B. (1989). Tailoring eor processes to geologic environments. *World Oil;(USA)*, 209(5). 8

Índice Remissivo

A

Análise de Domínio, 7
AOO - Análise Orientada a Objeto, 20

C

Casos de uso do software, 5
Conceitos Fundamentais e Propriedades
das Rochas e Fluidos, 10
Conceitos Teóricos, 9

D

Diagrama de atividades, 24
Diagrama de casos de uso específico, 6
Diagrama de casos de uso geral, 5
Diagrama de Classes, 20
Diagrama de comunicação - colaboração,
23
Diagrama de implantação, 28
Diagrama de máquina de estado, 24
Diagrama de pacotes - assuntos, 19
Diagrama de sequência - eventos e men-
sagens, 22
Diagrama de sequência geral, 22
Dicionário de classes, 22

E

Elaboração, 7
Escopo do Problema, 1
Especificação, 3, 4
Especificação do Software – Descrição dos
Requisitos, 3

I

Implementação, 30

N

Nome do sistema/produto e componen-
tes, 3

O

Objetivos, 2

P

Projeto, 26

R

Requisitos funcionais, 4
Requisitos não funcionais, 5

T

Teste, 46
Teste Laboratorial para utilização do Soft-
ware, 13