

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

PROJETO ENGENHARIA
DESENVOLVIMENTO DO SOFTWARE
ANÁLISE DO EFEITO DE PELÍCULA A PARTIR DE CURVAS DE
ÍNDICE DE PRODUTIVIDADE (IPR) PARA POÇOS VERTICAIS EM
ESCOAMENTO MONOFÁSICO DE ÓLEO
TRABALHO DA DISCIPLINA PROGRAMAÇÃO PRÁTICA

Versão 1.0
MATHEUS CARVALHO DUARTE
Prof. André Duarte Bueno

MACAÉ - RJ
Agosto - 2018

Sumário

1	Introdução	1
1.1	Escopo do problema	1
1.2	Objetivos	3
2	Especificação	4
2.1	Nome do sistema/produto	4
2.2	Especificação	4
2.2.1	Requisitos funcionais	5
2.2.2	Requisitos não funcionais	5
2.3	Casos de uso	5
2.3.1	Diagrama de caso de uso geral	6
2.3.2	Diagrama de caso de uso específico	7
3	Elaboração	8
3.1	Análise de domínio	8
3.2	Formulação teórica	9
3.2.1	Equação da Difusividade Hidráulica (E.D.H.)	9
3.2.1.1	Equação da Difusividade Hidráulica - Caso Óleo	9
3.2.2	Soluções Analíticas da Equação da Difusividade	10
3.2.2.1	Regime Transiente	10
3.2.2.2	Regime Pseudopermanente	11
3.2.2.3	Regime Permanente	12
3.2.3	Efeito de Película (<i>skin effect</i>)	12
3.2.3.1	Efeito de Película Total	12
3.2.3.2	Efeito de Película da Zona Alterada - Fórmula de Hawkins (s_d)	13
3.2.3.3	Efeito de Película por Canhoneio Parcial (s_c)	13
3.2.3.4	Efeito de Película por Desvio do Poço (s_{dev})	14
3.3	Identificação de pacotes – assuntos	15
4	AOO – Análise Orientada a Objeto	16
4.1	Diagramas de classes	16

4.1.1	Dicionário de classes	16
4.2	Diagrama de seqüência – eventos e mensagens	18
4.2.1	Diagrama de seqüência geral	18
4.2.2	Diagrama de seqüência específico	18
4.3	Diagrama de comunicação – colaboração	20
4.4	Diagrama de máquina de estado	20
4.5	Diagrama de atividades	21
5	Projeto	22
5.1	Projeto do sistema	22
5.2	Projeto orientado a objeto – POO	23
5.2.0.1	Efeitos do projeto no modelo estrutural	23
5.2.0.2	Efeitos do projeto no modelo dinâmico	24
5.2.0.3	Efeitos do projeto nos atributos	24
5.2.0.4	Efeitos do projeto nos métodos	24
5.2.0.5	Efeitos do projeto nas heranças	25
5.2.0.6	Efeitos do projeto nas associações	25
5.2.0.7	Efeitos do projeto nas otimizações	25
5.3	Diagrama de componentes	26
5.4	Diagrama de implantação	27
6	Implementação	28
6.1	Código fonte	28
7	Teste	60
7.1	Teste 1: Entrada de dados pelo teclado	60
7.2	Teste 2: Entrada de dados por aquivo de disco	63
7.3	Teste 3: Visualizar dados de arquivo de disco	64
7.4	Teste 4: Salvar dados em arquivo de disco	65
7.5	Teste 5: Plotar curvas de IPR	66
7.6	Teste 6: Plotar curvas de IPR de um arquivo de disco	67
7.7	Teste 7: Salvar gráfico	68
7.8	Teste 8: Informações sobre o programa	69
8	Documentação	70
8.1	Documentação do usuário	70
8.1.1	Como rodar o software	70
8.2	Documentação para desenvolvedor	72
8.2.1	Dependências	72
8.2.2	Como gerar a documentação usando doxygen	73

9 Sugestões para Trabalhos Futuros**76**

Capítulo 1

Introdução

No presente projeto de engenharia desenvolve-se o software Análise do Efeito de Película a partir de Curvas de Índice de Produtividade (IPR) em Escoamento Monofásico de Óleo, um software aplicado a engenharia de petróleo e que utiliza o paradigma da orientação a objetos. São implementadas soluções analíticas da E.D.H. (Equação da Difusividade Hidráulica) para poços verticais em escoamento monofásico de óleo, admitindo-se os regimes transiente, pseudopermanente e permanente, assim como equações para o cálculo do efeito de película (*skin*) total. Estas fórmulas são baseadas na premissa em que há um dano/estímulo causado nas proximidades do poço devido seu desvio até a formação alvo, alteração da permeabilidade ou canhoneio parcial do reservatório. A partir dos parâmetros do poço é possível calcular cada tipo de *skin* e plotar um gráfico que compare as curvas de IPR (*Inflow Performance Relationship*) nos seguintes cenários: dano a formação nulo e não-nulo.

1.1 Escopo do problema

A curva de IPR, também conhecida como curva de influxo, curva de pressão disponível ou curva do índice de produtividade, descreve o comportamento da pressão disponível no fundo do poço em função da vazão de fluidos, ambas medidas na profundidade dos canhoneados para um dado instante de tempo da vida do reservatório. Reflete, portanto, o comportamento do fluxo em meios porosos e através desta curva é possível determinar o Índice de Produtividade (IP) do poço. Uma outra aplicabilidade desta curva é monitorar a pressão em que o fluxo no reservatório se torna bifásico, ou seja, produz óleo e gás, considerando que não há produção de água. Esta pressão é conhecida como pressão de bolha e evita-se, em geral, fazer com que o reservatório atinja este ponto, uma vez que a produção de óleo decresce significativamente devido a expansão do gás em solução. Assim, pode-se, por exemplo, julgar como interessante a adição de poços injetores a fim de manter a pressão do poço acima da pressão de bolha, garantindo uma produção de óleo satisfatória.

Para melhor precisão no comportamento da pressão e da vazão ao longo do poço deve-se levar em consideração o efeito de película (*skin effect*) nas proximidades do poço. O fluxo radial para um poço vertical resulta em uma velocidade de fluxo cada vez mais alta em suas proximidades devido ao maior gradiente de pressão. Van Everdingen e Hurst (1949) introduziram o conceito de uma diferença de pressão, Δp_s , ocorrendo ao longo de uma distância infinitesimal no raio do poço proporcional ao *skin* para explicar um perfil de fluxo não ideal.

Matematicamente, o efeito de película de Van Everdingen e Hurst não tem dimensão física e é análogo ao coeficiente de transferência de calor. O *skin* pode ser positivo ou negativo, diminuindo ou aumentando a produtividade, respectivamente. Portanto, torna-se útil determinar maneiras de reduzir um skin positivo ou induzir um efeito negativo, uma vez que este resulta na estimulação do poço.

Efeitos de película podem ser causados por fatores como o desvio do poço até a formação alvo, alteração da permeabilidade, canhoneio parcial do reservatório, pela turbulência do fluxo no meio poroso, entre outros. Em geral, qualquer fenômeno que cause distorção ou restrição das linhas de fluxo em direção ao poço (que pode ser visto como uma distorção na escala da garganta de poros e dos poros) resulta em um efeito de película [Economides et al. 2013].

A fim de se obter estimativas destes efeitos, encontra-se na literatura e artigos científicos, equações que determinam, a partir de parâmetros do poço, o efeito de película causado por cada um dos fatores citados, podendo assim, posteriormente, obter um skin total que irá compor as equações das curvas de IPR, melhorando a predição da produtividade do poço.

1.2 Objetivos

Os objetivos deste projeto de engenharia são:

- Objetivo geral:
 - Utilizar equações propostas na literatura e em artigos científicos para o cálculo do efeito de película total.
 - Plotar curvas de IPR para poços verticais em escoamento monofásico de óleo a partir de soluções analíticas da E.D.H. considerando regime transiente, pseudopermanente e permanente a partir do software externo Gnuplot.
- Objetivos específicos:
 - Calcular o efeito de película causado pela alteração da permeabilidade do reservatório nas proximidades do poço a partir da Fórmula de Hawkins (1956).
 - Calcular o efeito de película causado pelo canhoneio parcial do reservatório a partir das fórmulas propostas por Papatzacos (1987).
 - Calcular o efeito de película causado pelo desvio do poço a partir das fórmulas propostas por Besson (1990).

Capítulo 2

Especificação

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

2.1 Nome do sistema/produto

Nome	Análise do Efeito de Película a partir de Curvas de Índice de Produtividade (IPR) em Escoamento Monofásico de Óleo.
Componentes principais	Sistema para cálculo de efeitos de película (<i>skin</i>).
Missão	Uma ferramenta de suporte educacional para cursos de Elevação e Escoamento, para auxiliar no estudo dos efeitos de película de diferentes tipos e análise de curvas de IPR para poços verticais de óleo.

2.2 Especificação

Este programa de engenharia tem por objetivo analisar curvas de IPR para poços verticais em escoamento monofásico de óleo nos seguintes cenários: efeito de película nulo e não-nulo. Através de soluções analíticas da E.D.H. (Equação da Difusividade Hidráulica) em regime transiente, pseudopermanente e permanente, o sistema utiliza o software externo Gnuplot para gerar um gráfico da pressão de poço em função da vazão de fluidos.

O programa deve ler os parâmetros do poço armazenados no disco ou inseridos pelo usuário como dados, e a partir deles, calcular os efeitos de película (*skin*) causados pelo desvio do poço até a formação alvo, alteração da permeabilidade em suas proximidades e canhoneio parcial do reservatório através de equações propostas na literatura e em artigos científicos.

O software será desenvolvido utilizando o conceito de programação orientada a objeto e interface em modo texto.

2.2.1 Requisitos funcionais

Apresenta-se a seguir os requisitos funcionais.

RF-01	O usuário deverá ter liberdade para escolher todos os dados de entrada.
RF-02	Deve permitir o carregamento e leitura de arquivos criados pelo software.
RF-03	Deve permitir a escolha do regime do reservatório (transiente, pseudopermanente ou permanente).
RF-04	O software plotará os resultados em um gráfico. O gráfico poderá ser salvo como imagem ou ter seus dados exportados como texto.

2.2.2 Requisitos não funcionais

RNF-01	Os cálculos devem ser feitos utilizando-se as equações citadas na seção 1.2.
RNF-02	O programa deverá ser multi-plataforma, podendo ser executado em <i>Windows</i> , <i>GNU/Linux</i> ou <i>Mac</i> .

2.3 Casos de uso

Nesta seção, apresenta-se uma tabela que especifica um caso de uso do sistema, bem como diagramas de casos de uso.

Tabela 2.1: Exemplo de caso de uso

Nome do caso de uso:	Determinação das curvas de Índice de Produtividade (IPR).
Resumo/descrição:	Determinação das curvas de IPR em poços verticais em escoamento monofásico de óleo, no regime transiente, pseudopermanente ou permanente.
Etapas:	<ol style="list-style-type: none"> 1. Escolher entrada de dados via teclado ou arquivo de disco. 2. Inserir ou carregar parâmetros do poço. 3. Selecionar o regime do reservatório 4. Calcular cada efeito de película. 5. Calcular o efeito de película total. 6. Analisar tabela comparativa entre vazão de fundo considerando skin nulo e não-nulo. 7. Gerar gráficos. 8. Analisar os resultados gráficos. 9. Salvar a imagem ou dados em disco.
Cenários alternativos:	Um cenário alternativo seria uma entrada errada do usuário (por exemplo, a pressão inicial do reservatório menor que a pressão final). O software apresentará uma mensagem de erro informando a incoerência do dado.

2.3.1 Diagrama de caso de uso geral

O diagrama de caso de uso geral da Figura 2.1 mostra o usuário interagindo com o software para obter um gráfico da curva do Índice de Produtividade (IPR).

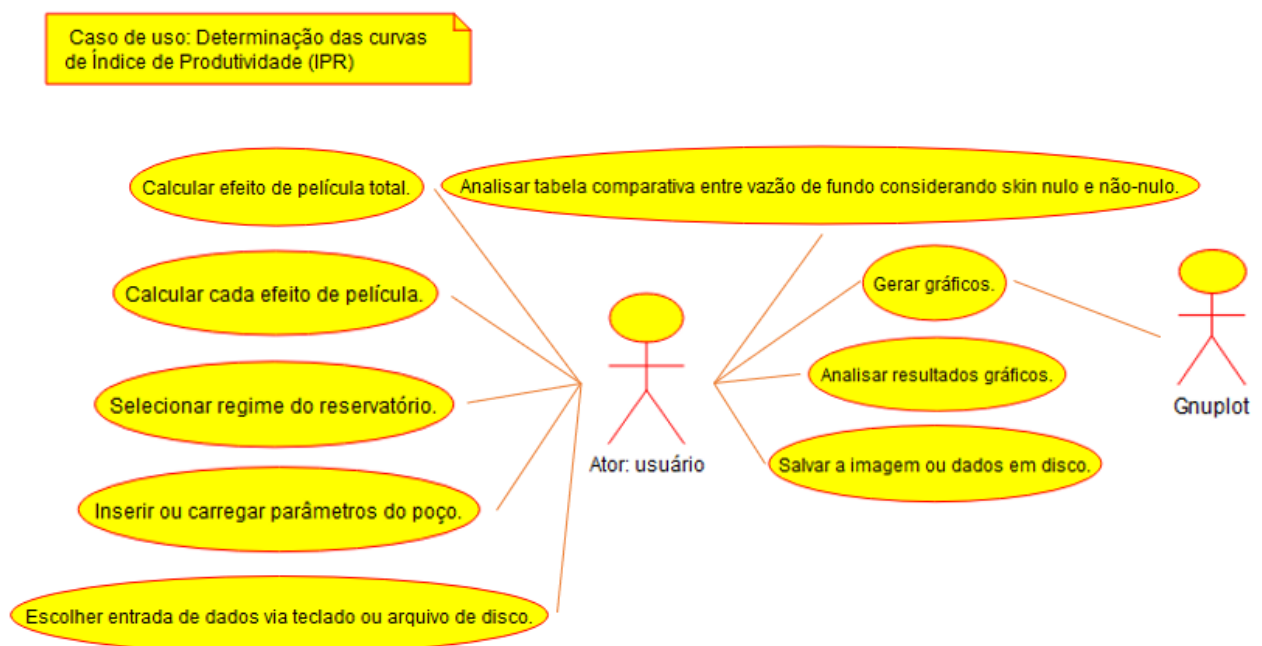


Figura 2.1: Diagrama de caso de uso – Caso de uso geral

2.3.2 Diagrama de caso de uso específico

O caso de uso “Determinação das curvas de Índice de Produtividade (IPR)” descrito na Figura 2.1 e na Tabela 2.1 é detalhado na Figura 2.2 para o caso específico de um reservatório no regime permanente. Neste caso de uso específico, o usuário insere os dados via teclado, seleciona o regime do reservatório e após o cálculo dos efeitos de película, o software mostra a tabela comparando a vazão considerando skin nulo e não-nulo. O usuário opta por plotar gráfico de Pressão *versus* Vazão com os resultados obtidos utilizando o software externo Gnuplot e salva a imagem e os dados em disco.

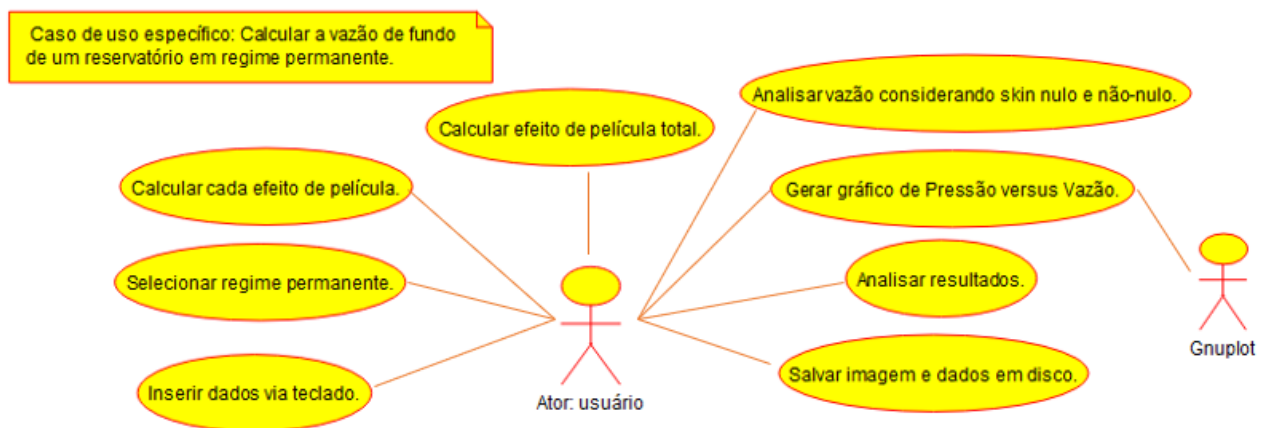


Figura 2.2: Diagrama de caso de uso específico – Calcular vazão de fundo de um reservatório em regime permanente

Capítulo 3

Elaboração

Neste capítulo, depois da definição dos objetivos, da especificação do software e da montagem dos primeiros diagramas de caso de uso, será apresentada a elaboração, que envolve o estudo de conceitos relacionados ao sistema a ser desenvolvido, a análise de domínio e a identificação de pacotes. Assim sendo, será feita uma análise dos requisitos, ajustando-os inicialmente de forma a desenvolver um sistema útil, que atenda às necessidades do usuário e, na medida do possível, permita seu reuso e futura extensão.

3.1 Análise de domínio

O programa a ser desenvolvido trata da análise do efeito de película a partir de curvas de Índice de Produtividade (IPR), uma área da engenharia de petróleo voltada ao estudo e investigação dos principais causadores do dano à formação nas proximidades do poço. Além disso, procura-se saber como mitigá-los, ou ainda, que recursos adotar para provocar o efeito oposto, sem a necessidade de empregar, ao menos inicialmente, a aplicação de métodos de estimulação que viabilizem o aumento da produção de hidrocarbonetos, como a acidificação e o fraturamento hidráulico, por exemplo.

De maneira geral, qualquer procedimento praticado durante o projeto de construção e manutenção de um poço causa alterações das linhas de fluxo, ou seja, o caminho percorrido pelo fluido até o poço. Em sua maioria, este fenômeno afeta negativamente a produção, diminuindo a interconexão entre os poros ou obstruindo as gargantas de poros nas localidades vizinhas. Para ilustrar este problema, pode-se citar as causas do efeito de película pelas seguintes atividades:

- Perfuração: invasão do filtrado de lama, fluido de perfuração e/ou partículas que o constituem na formação;
- Cimentação: penetração do fluido, partículas e/ou cimento na formação;
- Produção/Injeção: injeção de sólidos na formação, migração de finos, precipitação

de materiais insolúveis nos poros, crescimento bacteriano e declínio de pressão (compressão de poros ou desagregação dos grãos);

- Estimulação e *Work-over*: modificação nas propriedades da rocha como a molhabilidade e a permeabilidade.

A elaboração de uma estratégia efetiva, a fim de minimizar o *skin*, requer sua caracterização, determinando-se a origem, a localidade e o impacto do mesmo sobre o fluxo. Medidas reparadoras, em grande parte das vezes, são difíceis de serem executadas e, consequentemente, exigem elevado custo de operação. Além disso, o insucesso no controle do dano à formação pode reduzir a eficiência das demais atividades as quais o poço será submetido. Por conseguinte, a melhor ação a ser empregada é a prevenção, desenvolvendo um pré-planejamento profundo e detalhado a partir de uma análise integrada das etapas de perfuração, completação e produção.

3.2 Formulação teórica

Nesta seção, apresenta-se a formulação teórica dos conceitos fundamentais abordados ao longo deste projeto.

3.2.1 Equação da Difusividade Hidráulica (E.D.H.)

A equação da difusividade hidráulica descreve o fluxo de um fluido em um meio poroso. A partir desta formulação e de uma série de premissas, propõe-se a solução analítica para diferentes casos que exemplificam o comportamento de um reservatório. Esta equação é resultante da combinação de três princípios fundamentais da física: o princípio de conservação da massa (equação da continuidade), o princípio da conservação da quantidade de movimento representado pela equação de transporte (lei de Darcy) e uma equação de estado (compressibilidade do sistema), sendo válido tanto para fluxo monofásico ou multifásico. Mais informações sobre a dedução matemática podem ser encontradas em Rosa et al. (2006).

3.2.1.1 Equação da Difusividade Hidráulica - Caso Óleo

Para a formulação da equação da difusividade hidráulica em sistemas monofásicos para reservatórios de óleo são admitidas as seguintes hipóteses [Rosa, Carvalho e Xavier 2006]:

- Meio poroso homogêneo e isotrópico;
- Fluxo estritamente horizontal e isotérmico;
- Fluxo 3-D (tridimensional) monofásico;
- Sistema com um único fluido (saturação igual a 100%);

- Poço penetrando totalmente a formação;
- Pequenos gradientes de pressão;
- Fluido com compressibilidade pequena e constante, e viscosidade constante;
- Rocha com compressibilidade pequena e constante;
- Forças gravitacionais desprezíveis;
- Fluido e rocha não reagem entre si.

Ao final da dedução e das simplificações a equação da difusividade é apresentada na Eq. 3.1:

$$\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} + \frac{\partial^2 p}{\partial z^2} = \frac{k}{\phi \mu c_t} \frac{\partial p}{\partial t} \quad (3.1)$$

Sendo:

x, y e z : espaço tridimensional $[m]$;

p : pressão $[Pa]$;

t : tempo $[s]$;

k : permeabilidade $[m^2]$;

ϕ : porosidade $[m^3/m^3]$;

μ : viscosidade $[Pa.s]$;

c_t : compressibilidade total $[Pa^{-1}]$.

3.2.2 Soluções Analíticas da Equação da Difusividade

Nesta subseção serão apresentadas as soluções clássicas da equação da difusividade considerando um sistema monofásico de óleo em fluxo radial nos seguintes regimes: transiente, pseudopermanente e permanente. Essas soluções admitem que um reservatório cilíndrico seja produzido com vazão de fundo constante através de um único poço vertical centralizado. Com o objetivo de utilizar estas soluções para obter as curvas de Índice de Produtividade (IPR), as equações serão apresentadas na forma da vazão q $[STB/day]$ em função da pressão do poço $p_{wf}[psi]$.

3.2.2.1 Regime Transiente

Caracteriza-se por um regime transiente, um reservatório com meio poroso plano, horizontal e infinito lateralmente. Fisicamente, significa que o limite externo do reservatório ainda não sentiu a queda de pressão causada pela produção do poço.

$$q = \frac{k_H h (p_i - p_{wf})}{162.6 B_o \mu_o (\log(t) + \log(\frac{k_H}{\phi \mu_o c_t r_w^2}) - 3.23 + 0.87 s_t)} \quad (3.2)$$

$$t_{pss} = 1200 \frac{\phi \mu_o c_t r_e^2}{k_H} \quad (3.3)$$

Sendo:

- p_i : pressão inicial do reservatório [psi];
- r_e : raio do reservatório [ft];
- p_{wf} : pressão do poço [psi];
- r_w : raio do poço [ft];
- q : vazão [STB/day];
- t : tempo [hr];
- t_{pss} : tempo final do regime transiente [hr];
- k_H : permeabilidade horizontal [mD];
- h : espessura do reservatório [ft];
- ϕ : porosidade;
- B_o : fator volume de formação do óleo [bbl/STB];
- μ_o : viscosidade do óleo [cp];
- c_t : compressibilidade total [psi^{-1}].
- s_t : efeito de película (*skin*) total.

3.2.2.2 Regime Pseudopermanente

Caracteriza-se por um regime pseudopermanente, o fluxo de um reservatório em que não há alimentação no limite externo como, por exemplo, uma falha no limite do reservatório.

$$q = \frac{k_H h (\bar{p} - p_{wf})}{141.2 B_o \mu_o \left(\ln \left[\frac{r_e}{r_w} \right] - \frac{3}{4} + s_t \right)} \quad (3.4)$$

Sendo:

- \bar{p} : pressão média do reservatório [psi];
- r_e : raio do reservatório [ft];
- p_{wf} : pressão do poço [psi];
- r_w : raio do poço [ft];
- q : vazão [STB/day];
- k_H : permeabilidade horizontal [mD];
- h : espessura do reservatório [ft];
- B_o : fator volume de formação do óleo [bbl/STB];
- μ_o : viscosidade do óleo [cp];
- s_t : efeito de película (*skin*) total.

3.2.2.3 Regime Permanente

Caracteriza-se por um regime permanente, o fluxo de um reservatório em que há alimentação no limite externo como, por exemplo, um aquífero adjacente.

$$q = \frac{k_H h (p_e - p_{wf})}{141.2 B_o \mu_o (\ln \left[\frac{r_e}{r_w} \right] + s_t)} \quad (3.5)$$

Sendo:

p_e : pressão constante na fronteira do reservatório [psi];

r_e : raio do reservatório [ft];

p_{wf} : pressão do poço [psi];

r_w : raio do poço [ft];

q : vazão [STB/day];

k_H : permeabilidade horizontal [mD];

h : espessura do reservatório [ft];

B_o : fator volume de formação do óleo [bbl/STB];

μ_o : viscosidade do óleo [cp];

s_t : efeito de película (*skin*) total.

3.2.3 Efeito de Película (*skin effect*)

Nesta subsecção, é apresentado o efeito de película total, assim como seus componentes e a estimativa da contribuição de cada elemento.

3.2.3.1 Efeito de Película Total

O efeito de película total, s_t , para um poço vertical ou inclinado é composto por uma série de componentes. Em geral, eles podem ser somados e representados da seguinte forma:

$$s_t = (s_{comp})_d + s_c + s_{dev} + \sum s_{pseudo} \quad (3.6)$$

Onde:

$(s_{comp})_d$: combinação do efeito de película causado pela penetração do canhoneado e pela alteração da permeabilidade nas proximidades do poço. Para um poço aberto, este componente se resume ao *skin* obtido pela Fórmula de Hawkins (1956);

s_c : efeito de película devido canhoneio parcial (quando a zona canhoneada é menor que a espessura do reservatório);

s_{dev} : efeito de película causado pelo desvio do poço, ou seja, sua inclinação;

$\sum s_{pseudo}$: efeito de película que leva em consideração todos os “*pseudoskins*”. Este tipo de efeito representa os skins que são dependentes do tipo do fluxo (monofásico ou multifásico) ou da vazão de produção.

1. Obs.: O efeito de película causado pela penetração do canhoneado é de difícil determinação, uma vez que os parâmetros que devem ser levados em consideração não podem ser medidos de forma clara como, por exemplo, o quanto que o canhoneado perfurou a formação. A proposição de equações para sua estimativa ainda é bastante discutida e questionada na área científica. Além disso, sua influência é relativamente menor se comparado ao efeito causado pela alteração da permeabilidade. Assim sendo, neste trabalho, será assumido que $(s_{comp})_d$ é aproximadamente o $skin$ obtido pela Fórmula de Hawkins (1956), mesmo para os casos em que o poço não é aberto.
2. Obs.: Este trabalho se restringirá aos efeitos de películas principais, desconsiderando “*pseudoskins*”. Trabalhos futuros poderão adicioná-los. Portanto, admite-se que o $skin$ total será definido para a implementação do código como:

$$s_t = s_d + s_c + s_{dev} \quad (3.7)$$

3.2.3.2 Efeito de Película da Zona Alterada - Fórmula de Hawkins (s_d)

A fórmula proposta por Hawkins (1956) descreve o efeito de película causado pela alteração da permeabilidade em uma zona próxima ao poço, tornando-se conveniente usá-la por relacionar a permeabilidade da região alterada e o dano à formação. A dedução desta fórmula é mostrada em Economides et al. (2013) e é apresentada a seguir:

$$s_d = \left(\frac{k}{k_s} - 1 \right) \ln \left(\frac{r_s}{r_w} \right) \quad (3.8)$$

s_d : efeito de película causado pela alteração da permeabilidade nas proximidades do poço;

k_s : permeabilidade da zona alterada [mD];

k : permeabilidade do reservatório [mD];

r_s : raio da zona alterada [ft];

r_w : raio do poço [ft].

3.2.3.3 Efeito de Película por Canhoneio Parcial (s_c)

Para casos em que há canhoneio parcial, ou seja, espessura canhoneada menor que a do reservatório, o efeito de película s_c é obtido a partir da equação proposta por Papatzacos (1987) mostrada a seguir:

$$s_c = \left(\frac{1}{h_{pD}} - 1 \right) \ln \left(\frac{\pi}{2r_D} \right) + \frac{1}{h_{pD}} \ln \left[\frac{h_{pD}}{2 + h_{pD}} \left(\frac{A - 1}{B - 1} \right)^{1/2} \right] \quad (3.9)$$

$$A = \frac{1}{h_{1D} + \frac{h_{pD}}{4}} \quad (3.10)$$

$$B = \frac{1}{h_{1D} + \frac{3h_{pD}}{4}} \quad (3.11)$$

$$r_D = \frac{r_w}{h} \left(\frac{k_V}{k_H} \right)^{1/2} \quad (3.12)$$

$$h_{1D} = \frac{h_1}{h} \quad (3.13)$$

$$h_{pD} = \frac{h_p}{h} \quad (3.14)$$

s_c : efeito de película devido canhoneio parcial;

h : espessura do reservatório [ft];

h_p : espessura da zona canhoneada [ft];

h_{pD} : espessura adimensional da zona canhoneada;

h_1 : espessura da zona perfurada e não-canhoneada [ft];

h_{1D} : espessura adimensional da zona perfurada e não-canhoneada;

r_w : raio do poço [ft];

r_D : raio adimensional do poço;

k_H : permeabilidade horizontal [mD];

k_V : permeabilidade vertical [mD];

A, B : constantes.

A Figura 3.1 ilustra os parâmetros de poço necessários para o cálculo de s_c .

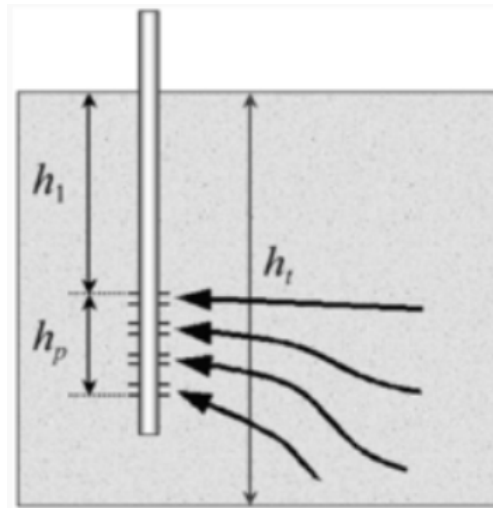


Figura 3.1: Tipos de espessura a serem determinados para o cálculo de s_c

3.2.3.4 Efeito de Película por Desvio do Poço (s_{dev})

O efeito de um poço desviado (inclinado) impacta no aumento de sua produtividade, uma vez que há um maior contato com o reservatório. Portanto, há um estímulo à

formação e este *skin* sempre será negativo. Besson (1990) apresenta equações analíticas para o efeito causado pela inclinação de um poço de ângulo θ para reservatórios isotrópicos ou anisotrópicos.

$$s_{dev} = \ln\left(\frac{4r_w}{L\beta\gamma}\right) + \frac{h}{\gamma L} \ln\left(\frac{\sqrt{Lh}}{4r_w} \frac{2\beta\sqrt{\gamma}}{1 + 1/\gamma}\right) \quad (3.15)$$

$$\gamma = \sqrt{\frac{1}{\beta^2} + \frac{h^2}{L^2}\left(1 - \frac{1}{\beta^2}\right)} \quad (3.16)$$

$$L = \frac{h}{\cos(\theta)} \quad (3.17)$$

$$\beta = \sqrt{\frac{k_V}{k_H}} \quad (3.18)$$

s_{dev} : efeito de película causado pelo desvio do poço;

r_w : raio do poço [ft];

h : espessura do reservatório [ft];

L : comprimento horizontal do reservatório atingido pelo poço [ft];

k_H : permeabilidade horizontal [mD];

k_V : permeabilidade vertical [mD];

θ : ângulo de inclinação do poço.

β, γ : constantes.

3.3 Identificação de pacotes – assuntos

Na Figura 3.2 está representado o diagrama de pacotes, mostrando a dependência entre as partes do sistema.

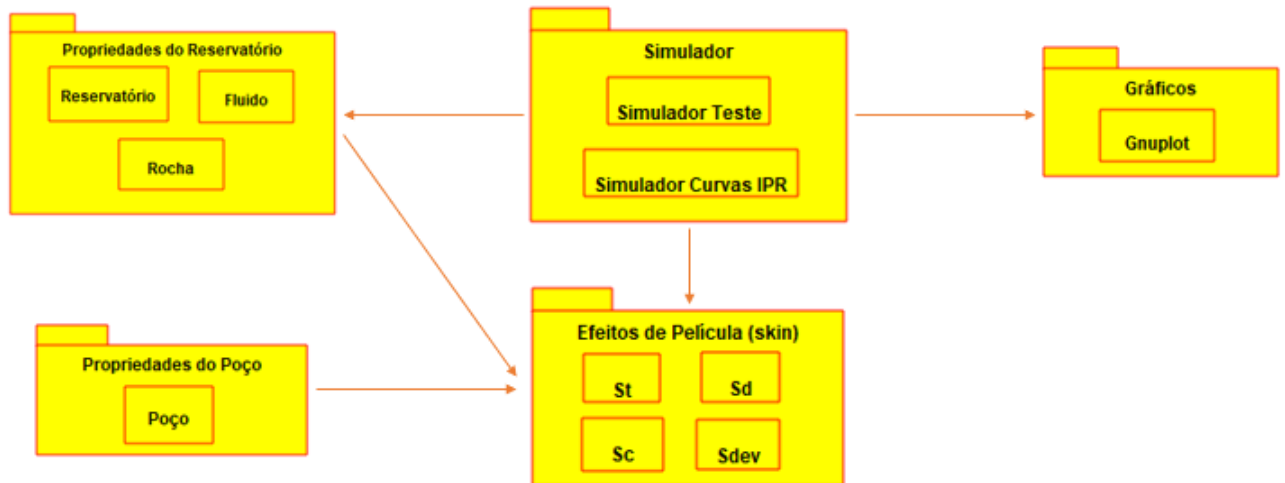


Figura 3.2: Diagrama de pacotes

Capítulo 4

AOO – Análise Orientada a Objeto

A AOO – Análise Orientada a Objeto utiliza algumas regras para identificar os objetos de interesse, as relações entre os pacotes, as classes, os atributos, os métodos, as heranças, as associações, as agregações, as composições e as dependências.

4.1 Diagramas de classes

O diagrama de classes é apresentado na Figura 4.1.

4.1.1 Dicionário de classes

- Classe CSimuladorCurvasIPR: classe responsável pela simulação do software. Esta classe recebe as escolhas do usuário sobre o regime do reservatório, recebe efeito de película total e calcula as vazões de fundo considerando skin nulo e não-nulo.
- Classe CRocha: classe responsável por receber as propriedades da rocha.
- Classe CFluido: classe responsável por receber as propriedades fluido.
- Classe CReservatorio: classe responsável por receber os parâmetros dimensionais do reservatório.
- Classe CPoco: classe responsável por receber os parâmetros dimensionais do poço.
- Classe CSkin: classe-base que representa o conceito geral de efeito de película.
- Classe CSkinDano: classe derivada da classe CSkin que representa o efeito de película devido alteração da permeabilidade na região próxima ao poço.
- Classe CSkinDesvio: classe derivada da classe CSkin que representa o efeito de película devido desvio (inclinação) do poço até a formação alvo.
- Classe CSkinCanhoneio: classe derivada da classe CSkin que representa o efeito de película devido canhoneio parcial do reservatório.

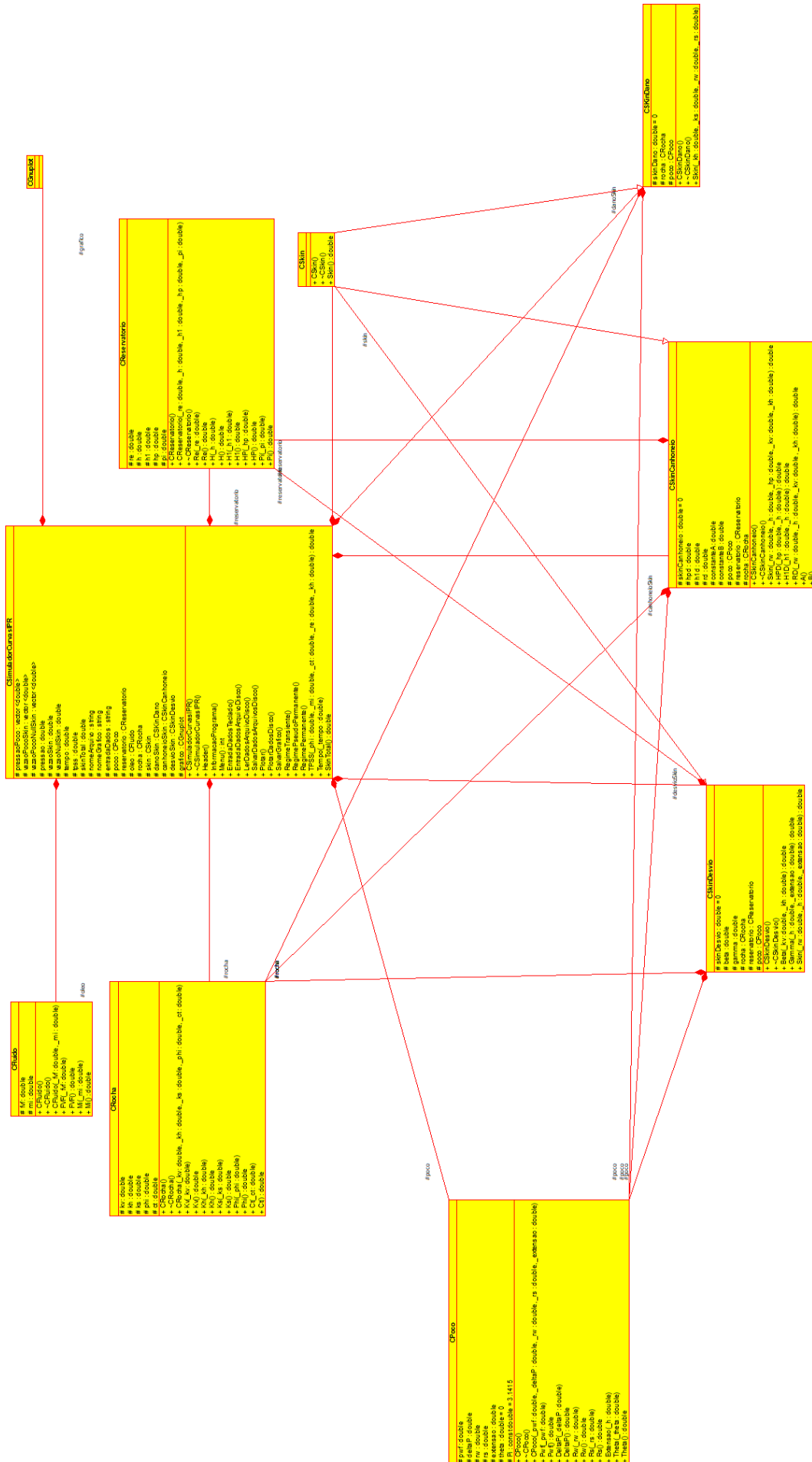


Figura 4.1: Diagrama de classes

- Classe CGnuPlot: classe responsável pela parte gráfica do programa, a partir do uso do 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

A Figura 4.2 exemplifica um diagrama de seqüência geral em que o usuário executa as principais funções do software, podendo optar por realizar a entrada de dados pelo teclado ou arquivo de disco, escolher o regime do reservatório, plotar gráficos, salvar os dados gerados em modo ou em imagem.

4.2.2 Diagrama de seqüência específico

A Figura 4.3 exemplifica um diagrama de seqüência em específico em que o usuário opta por plotar as curvas de IPR a partir de dados gerados previamente e salvos em modo texto.

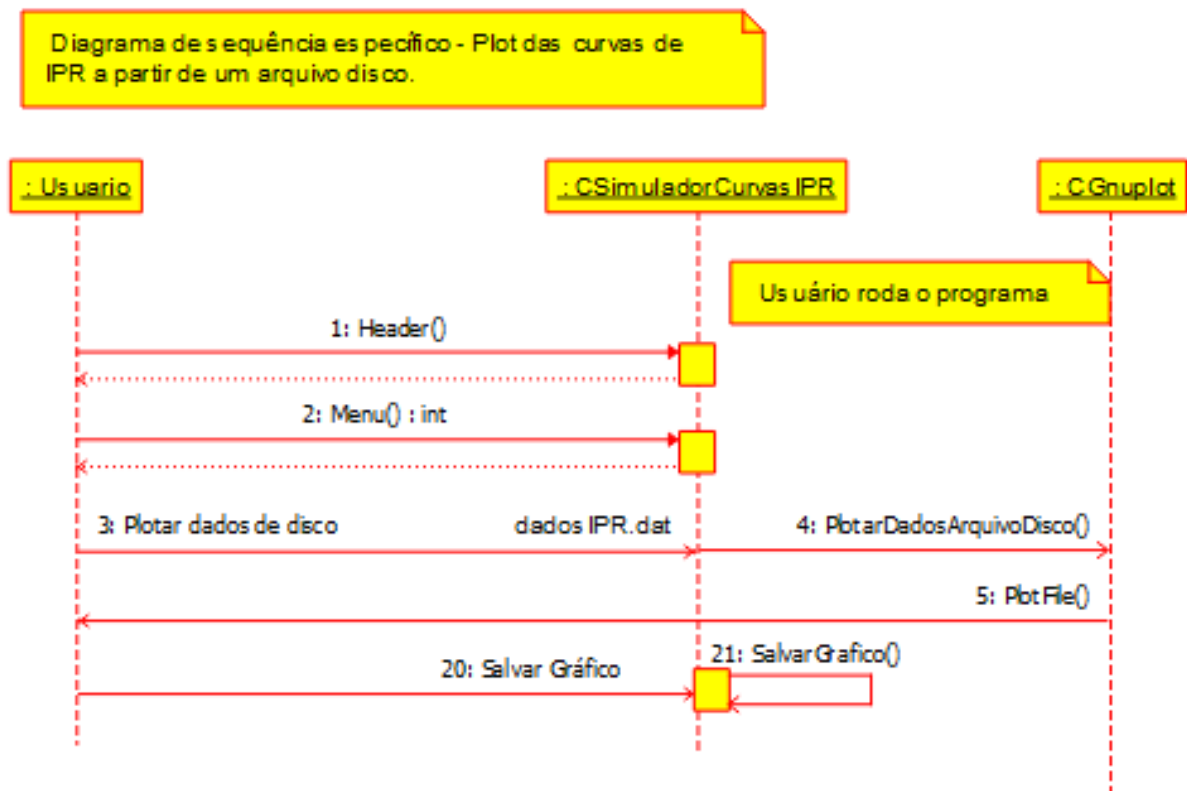


Figura 4.3: Diagrama de seqüência específico

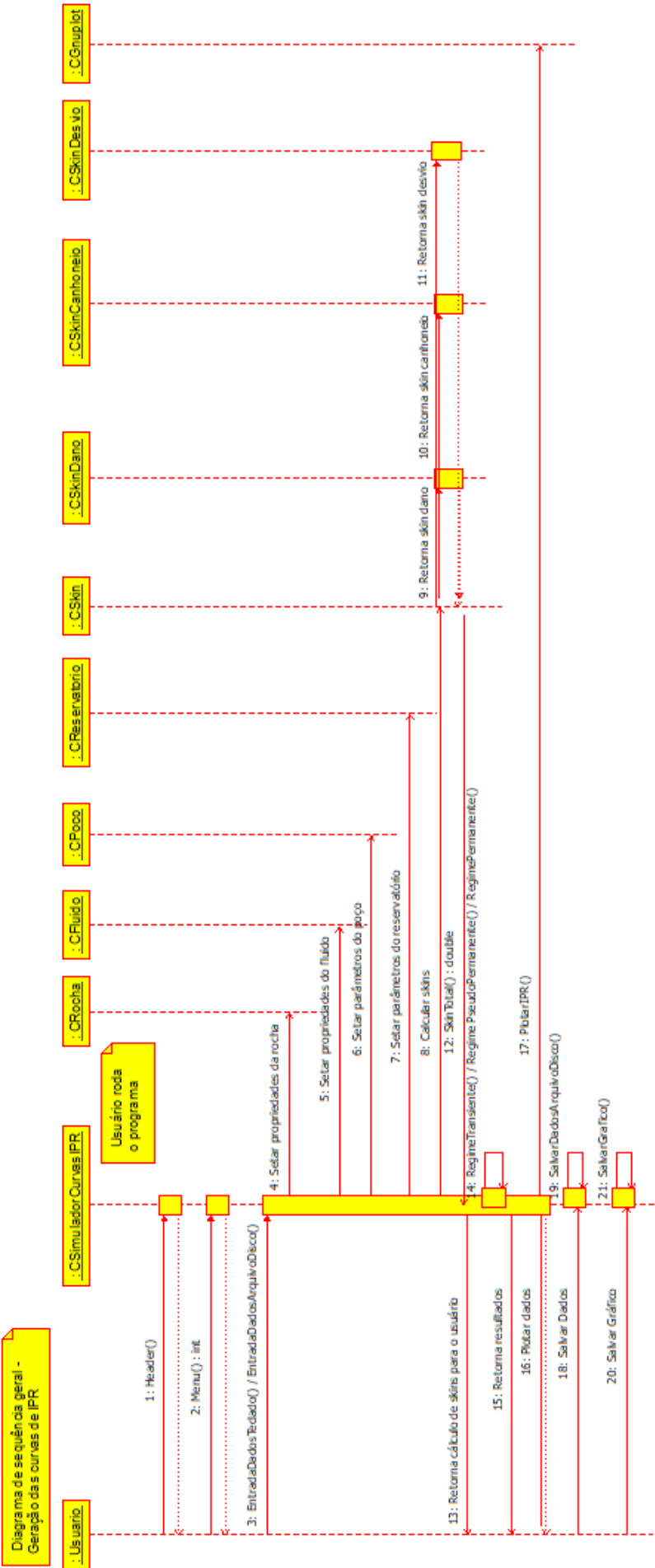


Figura 4.2: Diagrama de sequência geral

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.

Veja na Figura 4.3 o diagrama de comunicação mostrando a sequência de passos executadas pelo software desde a entrada de dados feita pelo usuário até a obtenção dos gráficos.

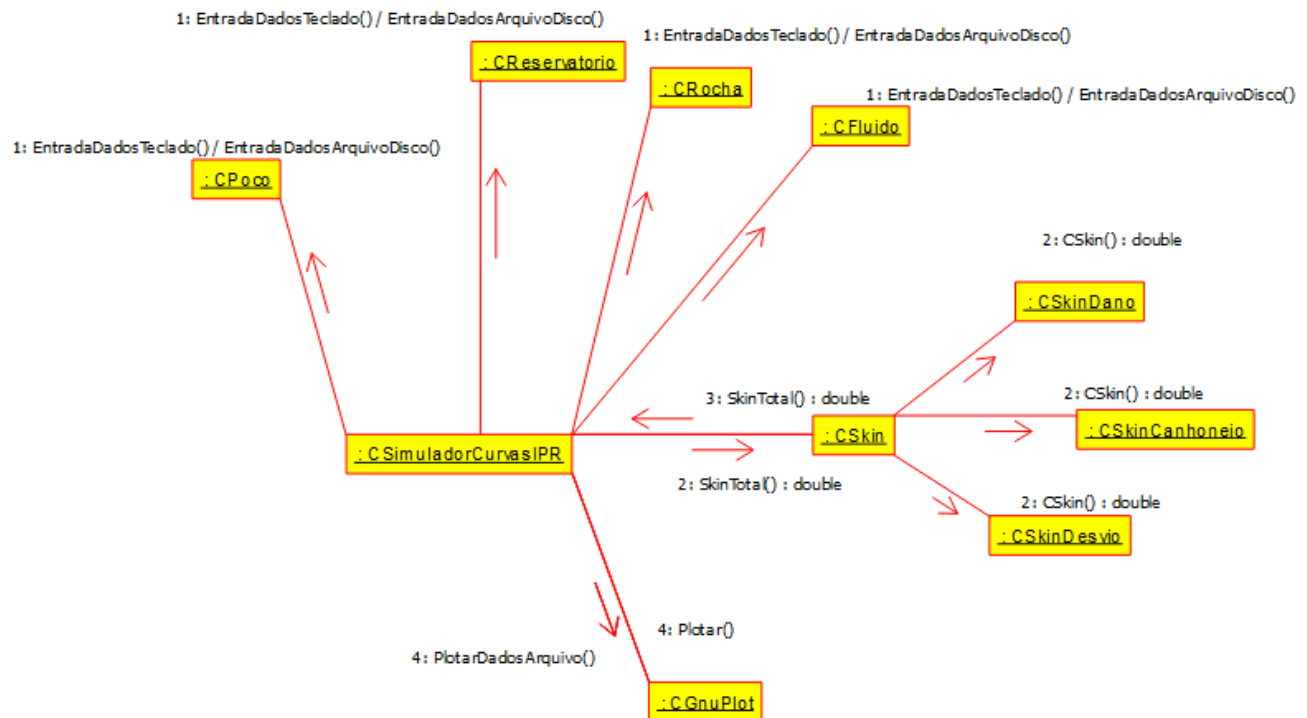


Figura 4.4: Diagrama de comunicação

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.

Veja na Figura 4.5 o diagrama de máquina de estado para os objetos da classe CSimuladorCurvasIPR para o caso de um reservatório em regime permanente.

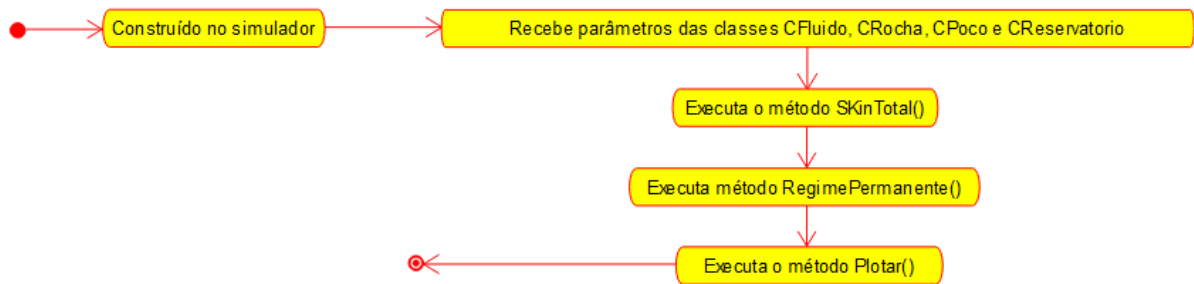


Figura 4.5: Diagrama de máquina de estado

4.5 Diagrama de atividades

Veja na Figura 4.6 o diagrama de atividades correspondente ao procedimento do plot de curvas de IPR considerando skin nulo e não-nulo.

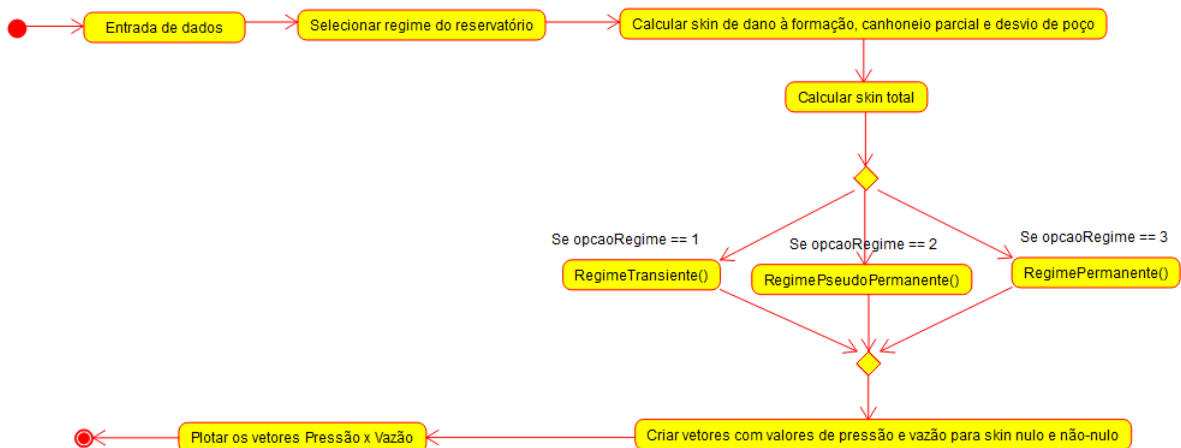


Figura 4.6: Diagrama de atividades

Capítulo 5

Projeto

Neste capítulo do projeto de engenharia veremos questões associadas ao projeto do sistema, incluindo protocolos, recursos, plataformas suportadas, implicações nos diagramas feitos anteriormente, diagramas de componentes e implantação.

5.1 Projeto do sistema

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

Deve-se definir padrões de documentação, padrões para o nome das classes, padrões de retorno e de parâmetros em métodos, características da interface do usuário e características de desempenho.

O projeto do sistema é a estratégia de alto nível para resolver o problema e elaborar uma solução.

1. Protocolos

- Neste projeto o software irá se comunicar com o componente externo Gnuplot, que gerará os gráficos escolhidos pelo usuário.
- Será efetuada a entrada de dados via arquivo de texto, formato ascii, com a extensão .dat ou através do teclado.

2. Recursos

- O presente programa utilizará o HD, o processador, o teclado, a memória, a tela e os demais componentes internos do computador.

3. Plataformas

- A linguagem usada para o programa será C++, portanto este será multiplataforma, podendo ser executado no Windows, Mac OS X e Gnu/Linux.
- O software utilizará a biblioteca externa CGnuplot que permitirá o acesso ao programa Gnuplot. Uma cópia dos arquivos .h e .cpp está disponibilizada.

4. Ambiente de desenvolvimento

- O ambiente para montar a interface de desenvolvimento - IDE será o software CodeBlocks (Windows), Kate (GNU/Linux) e o Xcode (Mac OS X). O compilador será o gcc/g++.

5.2 Projeto orientado a objeto – POO

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

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

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

5.2.0.1 Efeitos do projeto no modelo estrutural

- Aqui estão estabelecidos as dependências e restrições do programa.
 - O programa utiliza o HD, o processador e o teclado do computador;
 - O software pode ser executado nas plataformas GNU/Linux, Windows ou Mac;
 - Em casa um dos sistemas operacionais, existe a necessidade de instalação do software Gnuplot para o funcionamento do programa;
 - O código possui comentários com explicações dos algoritmos a serem executados.

5.2.0.2 Efeitos do projeto no modelo dinâmico

- Revisar os diagramas de sequência e de comunicação considerando a plataforma escolhida.
 - Foi realizado no diagrama de sequência uma generalização mais ampla do escopo do problema, fornecendo ao usuário maior interação com o software. Por exemplo, o usuário poderá carregar e visualizar arquivos de disco e plotá-los sem a necessidade de executar todos os métodos exigidos no primeiro acesso.
 - O diagrama de comunicação foi melhorado, possuindo uma sequência lógica melhor representada e mais intuitiva.
- Verificar a necessidade de se revisar, ampliar e adicionar novos diagramas de máquinas de estado e de atividades.
 - Os diagramas de máquina de estado e atividade foram revisados e corrigidos para que estivessem coerentes com a lógica do código implementado. Não foi necessária a adição de novos diagramas, uma vez que os cenários alternativos são análogos aos representados nos diagramas obtidos.

5.2.0.3 Efeitos do projeto nos atributos

- Atributos novos podem ser adicionados a uma classe, como, por exemplo, atributos específicos de uma determinada linguagem de softwareção (acesso a disco, ponteiros, constantes e informações correlacionadas).
 - Atributos dimensionais inicialmente presentes nas classes relacionados aos efeitos de película foram implementados nas classes CRocha, CPoco e CRocha, a fim de se facilitar o entendimento e agrupamento destes parâmetros.
 - O atributo fin foi criado para possibilitar a leitura dos parâmetros ou plotar dados de pressão *versus* vazão a partir de um arquivo de disco, assim como o atributo fout foi criado para a saída de dados em modo texto.

5.2.0.4 Efeitos do projeto nos métodos

- Em função da plataforma escolhida, verifique as possíveis alterações nos métodos. O projeto do sistema costuma afetar os métodos de acesso aos diversos dispositivos (exemplo: hd, rede).
 - Em virtude de usar leitura de disco, um método de inserção de dados através do teclado foi adicionado a fim de que o usuário possa digitar os dados desejados e não ficar preso somente a leitura do disco.

- Pelo mesmo motivo mencionado no item acima, um método de plotar dados de um arquivo de disco foi adicionado a fim de que o usuário possa carregar dados de maneira mais dinâmica e prática.

5.2.0.5 Efeitos do projeto nas heranças

- Reorganização das classes e dos métodos (criar métodos genéricos com parâmetros que nem sempre são necessários e englobam métodos existentes).
 - Foi realizado uma reformulação das classes, separando-as em classes menores e conceitos independentes.
- Revise as heranças nos diagramas de classes.
 - Heranças no diagrama de classes foram atribuídas a partir da classe-base CSkin e suas derivadas CSkinDano, CSkinCanhoneio e CSkinDesvio.

5.2.0.6 Efeitos do projeto nas associações

- Novas associações foram criadas a fim de se mostrar a relação entre as classes CRocha, CFluido, CReservatorio e CPoco com as classes CSkinDano, CSkinCanhoneio e CSkinDesvio.

5.2.0.7 Efeitos do projeto nas otimizações

- A ordem da execução pode ser invertida.
 - Inicialmente pensou-se em solicitar ao usuário que informasse primeiramente o regime do reservatório, uma vez que os parâmetros requeridos podem variar. Entretanto, ao observar que apenas o regime transiente necessita de parâmetros extras, a ordem de execução foi invertida, criando um padrão na entrada de dados tanto pelo teclado quanto por arquivo de disco. Os demais parâmetros exigidos são solicitados após os atributos comuns a todos os regimes.

As dependências dos arquivos e bibliotecas podem ser descritas pelo diagrama de componentes, e as relações e dependências entre o sistema e o hardware podem ser ilustradas com o diagrama de implantação.

5.3 Diagrama de componentes

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

Veja na Figura 5.1 o diagrama de componentes. A geração de objetos depende dos arquivos de extensão .h e .cpp. O programa executável a ser gerado depende das bibliotecas, dos arquivos .h e .cpp e dos arquivos de entrada.



Figura 5.1: Diagrama de componentes

5.4 Diagrama de implantação

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

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

Veja na Figura 5.2 o diagrama de implantação. Os dados de poço e do reservatório foram obtidos a partir de testes em células PVT ou teste de pressão, podendo ser salvos em arquivos com extensão .dat no computador. O programa importa os arquivos e utiliza o teclado e monitor para a comunicação com o usuário.

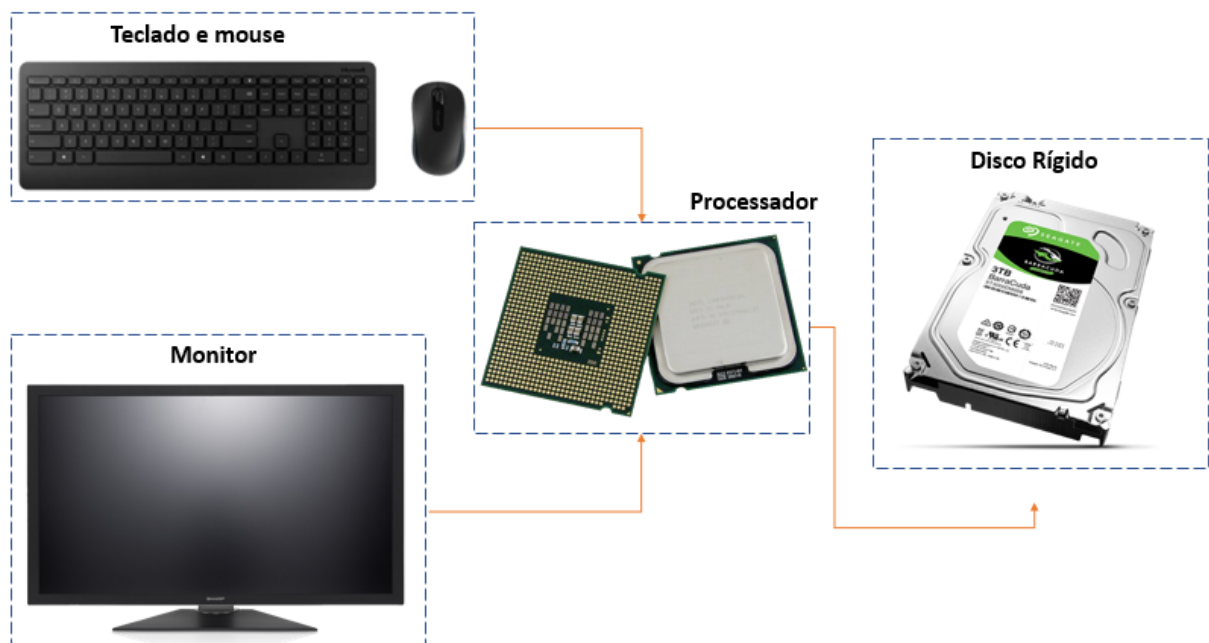


Figura 5.2: Diagrama de implantação.

Capítulo 6

Implementação

Neste capítulo do projeto de engenharia apresentamos os códigos fonte que foram desenvolvidos.

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

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

Nota: ao longo deste capítulo usamos inclusão direta de arquivos externos usando o pacote *listings* do L^AT_EX. Maiores detalhes de como a saída pode ser gerada estão disponíveis nos links abaixo.

- http://en.wikibooks.org/wiki/LaTeX/Source_Code_Listings.
- <http://mirrors.ctan.org/macros/latex/contrib/listings/listings.pdf>.

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 `CFluido`.

Listing 6.1: Arquivo de cabeçalho da classe `CFluido`.

```
1 /// Classe correspondente aos parametros de fluidos
2
3 #ifndef CFluido_h
4 #define CFluido_h
5
6 /// Classe que representa o conceito de fluido, assim como seus
   atributos e métodos
7 class CFluido{
```



```

8
9
10     protected:
11         /// fator volume-formacao
12         double fvf=0.0;
13         /// viscosidade
14         double mi=0.0;
15
16     public:
17         /// Contrutor default
18         CFluido(){};
19         /// Construtor sobrecarregado
20         CFluido(double _fvf, double _mi):fvf(_fvf),mi(_mi){};
21         /// Destrutor default
22         ~CFluido(){};
23
24         /// Fator Volume Formacao
25         void FVF (double _fvf){ fvf = _fvf;}
26         /// Fator Volume Formacao
27         double FVF () { return fvf;}
28
29         /// Viscosidade
30         void Mi (double _mi){ mi = _mi;}
31         /// Viscosidade
32         double Mi () { return mi;}
33
34 }; //< end class CFluido
35
36 #endif

```

Apresenta-se na listagem 6.2 o arquivo com código da classe CRocha.

Listing 6.2: Arquivo de cabeçalho da classe CRocha.

```

37 /// Classe correspondente aos parametros de rocha
38
39 #ifndef CRocha_h
40 #define CRocha_h
41
42 /// Classe que representa o conceito de rocha, assim como seus atributos
   e metodos
43 class CRocha{
44
45     protected:
46         /// permeabilidade vertical
47         double kv=0.0;
48         /// permeabilidade horizontal
49         double kh=0.0;
50         /// permeabilidade da zona alterada
51         double ks=0.0;

```

```

52      /// porosidade
53      double phi=0.0;
54      /// compressibilidade total
55      double ct=0.0;
56
57      public:
58      /// Contrutor default
59      CRocha(){};
60      /// Construtor sobrecarregado
61      CRocha(double _kv, double _kh, double _ks, double _phi, double _ct):
62          kv(_kv),kh(_kh),ks(_ks),phi(_phi),ct(_ct){};
63      /// Destrutor default
64      ~CRocha(){};
65
66      /// Permeabilidade vertical
67      void Kv (double _kv){ kv = _kv;}
68      /// Permeabilidade vertical
69      double Kv () { return kv;}
70      /// Permeabilidade horizontal
71      void Kh (double _kh){ kh = _kh;}
72      /// Permeabilidade horizontal
73      double Kh () { return kh;}
74      /// Permeabilidade da zona alterada
75      void Ks (double _ks){ ks = _ks;}
76      /// Permeabilidade da zona alterada
77      double Ks () { return ks;}
78      /// Porosidade
79      void Phi (double _phi){ phi = _phi;}
80      /// Porosidade
81      double Phi () { return phi;}
82      /// Compressibilidade total
83      void Ct (double _ct){ ct = _ct;}
84      /// Compressibilidade total
85      double Ct () { return ct;}
86  }; //< Fim class CRocha
87
88 #endif

```

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

Listing 6.3: Arquivo de cabeçalho da classe CPoco.

```

89 /// Classe correspondente aos parametros de poco
90
91 #ifndef CPoco_h
92 #define CPoco_h
93
94 #include <cmath>
95

```

```

96 /// Classe que representa o conceito de poço, assim como seus parâmetros
   e métodos
97 class CPoco{
98
99     protected:
100     /// pressão final do poço
101         double pwf=0.0;
102         /// variação de pressão
103     double deltaP=0.0;
104         /// raio do poço
105     double rw=0.0;
106         /// raio da zona alterada
107     double rs=0.0;
108         /// extensão do reservatório atingido pelo poço
109     double extensao=0.0;
110     /// ângulo de inclinação do poço. Por padrão o poço é perfeitamente
       vertical
111     double theta {0};
112
113     public:
114     /// Construtor default
115     CPoco(){};
116     /// Construtor sobrecarregado
117     CPoco(double _pwf, double _deltaP, double _rw, double _rs, double
       _extensao):pwf(_pwf),deltaP(_deltaP),rw(_rw),rs(_rs),extensao(
       _extensao){};
118     /// Destrutor default
119     ~CPoco(){};
120
121         /// Pressão final do poço
122     void Pwf (double _pwf){ pwf = _pwf;}
123     /// Pressão final do poço
124     double Pwf () { return pwf;}
125     /// Variação de pressão
126     void DeltaP (double _deltaP){ deltaP = _deltaP;}
127     /// Variação de pressão
128     double DeltaP () { return deltaP;}
129     /// Raio do poço
130     void Rw (double _rw){ rw = _rw;}
131     /// Raio do poço
132     double Rw () { return rw;}
133     /// Raio da zona alterada
134     void Rs (double _rs){ rs = _rs;}
135     /// Raio da zona alterada
136     double Rs () { return rs;}
137     /// Extensão do reservatório atingido pelo poço
138     inline double Extensao(double _h){extensao = _h/cos(theta); return
       extensao;}

```

```

139    /// Inclinação do poço
140    void Theta (double _theta){ theta = (M_PI/180.0)*_theta;}    ///<
        Funções trigonométricas no cmath possuem argumento em radianos
141    /// Inclinação do poço
142    double Theta () { return theta;}
143
144};    ///< Fim class CPoco
145
146#endif

```

Apresenta-se na listagem 6.4 o arquivo com código da classe CReservatorio.

Listing 6.4: Arquivo de cabeçalho da class CReservatorio.

```

147/// Classe correspondente aos parametros do reservatorio
148
149#ifndef CReservatorio_h
150#define CReservatorio_h
151
152/// Classe que representa o conceito de reservatorio, assim como seus
        parametros e metodos
153class CReservatorio{
154
155    protected:
156        /// raio do reservatorio
157        double re=0.0;
158        /// espessura do reservatorio
159        double h=0.0;
160        /// espessura da zona perfurada e nao-canhoneada
161        double h1=0.0;
162        /// espessura da zona canhoneada
163        double hp=0.0;
164        /// pressao inicial
165        double pi=0.0;
166
167    public:
168        /// Construtor default
169        CReservatorio(){};
170        /// Construtor sobrecarregado
171        CReservatorio(double _re, double _h, double _h1, double _hp, double
            _pi):re(_re),h(_h),h1(_h1),hp(_hp),pi(_pi){};
172        /// Destrutor default
173        ~CReservatorio(){};
174
175        /// Raio do reservatorio
176        void Re (double _re){ re = _re;}
177        /// Raio do reservatorio
178        double Re () { return re;}
179        /// Espessura do reservatorio
180        void H (double _h){ h = _h;}

```

```

181     /// Espessura do reservatorio
182     double H () { return h;}
183     /// Espessura da zona perfurada e nao-canhoneada
184     void H1 (double _h1){ h1 = _h1;}
185     /// Espessura da zona perfurada e nao-canhoneada
186     double H1 () { return h1;}
187     /// Espessura da zona canhoneada
188     void HP (double _hp){ hp = _hp;}
189     /// Espessura da zona canhoneada
190     double HP () { return hp;}
191     /// Pressao inicial
192     void Pi (double _pi){ pi = _pi;}
193     /// Pressao inicial
194     double Pi () { return pi;}
195
196 }; //< Fim class CReservatorio
197
198 #endif

```

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

Listing 6.5: Arquivo de cabeçalho da classe CSkin.

```

199 /// Classe correspondente aos parametros de dano a formacao (skin)
200
201 #ifndef CSkin_h
202 #define CSkin_h
203
204 /// Classe que representa o conceito generico de skin, assim como seus
    atributos e metodos
205 class CSkin{
206
207     public:
208     /// Construtor default
209     CSkin(){};
210     /// Destrutor default
211     ~CSkin(){};
212     /// Metodo generico para o calculo de skins
213     double Skin();
214
215 }; //< Fim CSkin
216
217 #endif

```

Apresenta-se na listagem 6.6 o arquivo com código da classe CSkinDano.

Listing 6.6: Arquivo de cabeçalho da classe CSkinDano.

```

218 /// Classe correspondente aos parametros de skin por dano ou estimulo a
    formacao
219

```

```

220 #ifndef CSkinDano_h
221 #define CSkinDano_h
222
223 #include <cmath>
224 /// Classe-base
225 #include "CSkin.h"
226
227 /// Classe que representa o conceito de skin devido alteracao da
permeabilidade nas proximidades do poço, assim como seus parametros e
metodos
228
229 class CSkinDano: public CSkin{
230
231     protected:
232         /// skin devido alteracao da permeabilidade nas proximidades do
poco
233         double skinDano {0};
234
235     public:
236         /// Construtor default
237         CSkinDano(){}
238         /// Destrutor default
239         ~CSkinDano(){};
240         /// Skin devido alteracao da permeabilidade nas proximidades do poço
241         inline double Skin(double _kh, double _ks, double _rw, double _rs){
242             skinDano = ((_kh/_ks) - 1.0)*(log(_rs/_rw)); return skinDano;}
243 }; //< Fim CSkinDano
244 #endif

```

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

Listing 6.7: Arquivo de cabeçalho da classe CSkinCanhoneio.

```

245 /// Classe correspondente aos parametros de skin por canhoneio parcial
246
247 #ifndef CSkinCanhoneio_h
248 #define CSkinCanhoneio_h
249
250 /// Classe-base
251 #include "CSkin.h"
252
253 /// Classe que representa o conceito de skin causado por canhoneio
parcial, assim como seus atributos e metodos
254 class CSkinCanhoneio: public CSkin{
255
256     protected:
257
258         /// skin calculado pela formula proposta por Papatzacos (1987)
259         double skinCanhoneio {0};

```

```

260    /// espessura adimensional da zona canhoneada
261    double hpd=1.0;
262    /// espessura adimensional da zona perfurada e nao-canhoneada
263    double h1d=1.0;
264    /// raio adimensional proposto na formula de Papatzacos
265    double rd=1.0;
266    /// constante
267    double constanteA=1.0;
268    /// constante
269    double constanteB=1.0;
270
271    public:
272    /// Construtor default
273    CSkinCanhoneio(){};
274    /// Destrutor default
275    ~CSkinCanhoneio(){};
276
277    /// Skin calculado pela formula proposta por Papatzacos (1987)
278    double Skin(double _rw, double _h, double _hp, double _kv, double _kh
279    );
280    /// Espessura adimensional da zona canhoneada
281    double HPD(double _hp, double _h);
282    /// Espessura adimensional da zona perfurada e nao-canhoneada
283    double H1D(double _h1, double _h);
284    /// Raio adimensional proposto na formula de Papatzacos
285    double RD(double _rw, double _h, double _kv, double _kh);
286    /// Constante
287    double A();
288    /// Constante
289    double B();
290}; //< Fim CSkinCanhoneio
291
292#endif

```

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

Listing 6.8: Arquivo de implementação da classe CSkinCanhoneio.

```

293/// Implementacao da classe CSkinCanhoneio
294
295/// Inclusao de bibliotecas
296#include "CSkinCanhoneio.h"
297#include <cmath>
298
299double CSkinCanhoneio::HPD(double _hp, double _h){
300    hpd = _hp/_h;
301    return hpd;
302};
303

```

```

304 double CSkinCanhoneio::H1D(double _h1, double _h){
305     h1d = _h1/_h;
306     return h1d;
307 };
308
309 double CSkinCanhoneio::RD(double _rw, double _h, double _kv, double _kh)
310 {
311     rd = (_rw/_h)*sqrt(_kv/_kh);
312     return rd;
313 };
314
315 double CSkinCanhoneio::A(){
316     constanteA = 1.0/(h1d+(hpd/4.0));
317     return constanteA;
318 };
319
320 double CSkinCanhoneio::B(){
321     constanteB = 1.0/(h1d+(3.0*hpd/4.0));
322     return constanteB;
323 };
324
325 double CSkinCanhoneio::Skin(double _rw, double _h, double _hp, double _kv,
326     double _kh){
327     skinCanhoneio = ((1.0/HPD(_hp, _h))-1.0)*log(M_PI/(2.0*RD(_rw, _h, _kv,
328         _kh))) + (1.0/HPD(_hp, _h))*log((HPD(_hp, _h)/(2.0+HPD(_hp, _h)))*
329         sqrt((A()-1.0)/(B()-1.0)));
330     return skinCanhoneio;
331 };

```

Apresenta-se na listagem 6.9 o arquivo com código da classe CSkinDesvio.

Listing 6.9: Arquivo de cabeçalho da classe CSkinDesvio.

```

328 /// Classe correspondente aos parametros do skin devido inclinacao do
329 poco
330 #ifndef CSkinDesvio_h
331 #define CSkinDesvio_h
332
333 /// Classe-base
334 #include "CSkin.h"
335
336 /// Classe que representa o conceito de skin causado pela inclinacao do
337 poco, assim como seus parametros e metodos
338 class CSkinDesvio: public CSkin{
339
340     protected:
341         /// skin calculado pela formula proposta por Besson (1990)
342         double skinDesvio {0};
343
344         /// constante

```



```

343         double beta=1.0;
344         /// constante
345         double gamma=1.0;
346
347     public:
348         /// Construtor default
349         CSkinDesvio(){};
350         /// Destrutor default
351         ~CSkinDesvio(){};
352
353         /// Constante
354         double Beta(double _kv, double _kh);
355         /// Constante
356         double Gamma(double _h, double _extensao);
357         /// Skin calculado pela formula proposta por Besson (1990)
358         double Skin(double _rw, double _h, double _extensao);
359
360 }; //< Fim class CSkinDesvio
361 #endif

```

Apresenta-se na listagem 6.10 o arquivo de implementação da classe CSkinDesvio.

Listing 6.10: Arquivo de implementação da classe CSkinDesvio.

```

362 /// Implementacao da classe CSkinDesvio
363
364 /// Inclusao de bibliotecas
365 #include "CSkinDesvio.h"
366 #include <cmath>
367
368 double CSkinDesvio::Beta(double _kv, double _kh){
369     beta = sqrt(_kv/_kh);
370     return beta;
371 };
372
373 double CSkinDesvio::Gamma(double _h, double _extensao){
374     gamma = sqrt((1.0/pow(beta,2))+(pow(_h,2)/pow(_extensao,2))
375                 *(1.0-(1.0/pow(beta,2))));
376     return gamma;
377 };
378 double CSkinDesvio::Skin(double _rw, double _h, double _extensao){
379     skinDesvio = log((4.0*_rw)/(_extensao*beta*gamma))+((_h)/(gamma*_
380                 _extensao))*log((sqrt(_extensao*_h))/(4.0*_rw)*((2.0*beta*sqrt(
381                 gamma))/(1.0+(1.0/gamma))));
382     return skinDesvio;
383 };

```

Apresenta-se na listagem 6.11 o arquivo com código da classe CSimuladorCurvasIPR.

Listing 6.11: Arquivo de cabeçalho da classe CSimuladorCurvasIPR.

```

382
383 #ifndef CSimuladorCurvasIPR_h
384 #define CSimuladorCurvasIPR_h
385
386 #include <vector>
387 #include <string>
388 #include "CGnuplot.h"
389
390 #include "CRocha.h"
391 #include "CFluido.h"
392 #include "CPoco.h"
393 #include "CReservatorio.h"
394 #include "CSkin.h"
395 #include "CSkinDano.h"
396 #include "CSkinCanhoneio.h"
397 #include "CSkinDesvio.h"
398
399
400 using namespace std;
401
402 /// Classe que simula a analise do efeito de pelicula a partir de curvas
de IPR
403
404 class CSimuladorCurvasIPR{
405
406     protected:
407         /// vetor que armazena pressoes do poco
408         vector <double> pressaoPoco;
409         /// vetor que armazena vazao do poco considerando skin
410         vector <double> vazaoPocoSkin;
411         /// vetor que armazena vazao do poco considerando skin nulo
412         vector <double> vazaoPocoNullSkin;
413         /// pressao medida no poco
414         double pressao=0.0;
415         /// vazao do poco considerando skin
416         double vazaoSkin=0.0;
417         /// vazao do poco considerando skin nulo
418         double vazaoNullSkin=0.0;
419         /// tempo de producao do reservatorio em regime transiente
420         double tempo=0.0;
421         /// tempo final do reservatorio em regime transiente
422         double tpss=0.0;
423         /// skin total
424         double skinTotal=0.0;
425
426         /// Objeto da classe CPoco
427         CPoco poco;

```

```
428     /// Objeto da classe CReservatorio
429     CReservatorio reservatorio;
430     /// Objeto da classe CFluido
431     CFluido oleo;
432     /// Objeto da classe CRocha
433     CRocha rocha;
434     /// Objeto da classe CSkin
435     CSkin skin;
436     /// Objeto da classe CSkinDano
437     CSkinDano danoSkin;
438     /// Objeto da classe CSkinCanhoneio
439     CSkinCanhoneio canhoneioSkin;
440     /// Objeto da classe CSkinDesvio
441     CSkinDesvio desvioSkin;
442     /// Objeto da classe CGnuplot
443     CGnuplot grafico;
444
445     /// Para salvar dados ou abrir dados salvos anteriormente
446     string nomeArquivo;
447     /// Para salvar grafico gerado
448     string nomeGrafico;
449     /// Para entrar com parametros por arquivo de disco
450     string entradaDados;
451
452 public:
453     /// Construtor default
454     CSimuladorCurvasIPR(){};
455     /// Destrutor default
456     ~CSimuladorCurvasIPR(){};
457
458     /// Cabecalho do programa
459     void Header();
460
461     /// Maiores informacoes sobre o programa
462     void InformacaoPrograma();
463
464     /// Menu de execucao
465     int Menu();
466
467     /// Metodo que solicita a entrada de dados comuns a todos os regimes
468         pelo teclado
469     void EntradaDadosTeclado();
470
471     /// Metodo que solicita a entrada de parametros pelo arquivo de
472         disco
473     void EntradaDadosArquivoDisco();
474
475     /// Metodo que faz leitura de dados de pressao e vazao de um arquivo
```

```

        de disco
474     void LerDadosArquivoDisco();
475
476     /// Metodo que salva arquivos no disco
477     void SalvarDadosArquivoDisco();
478
479     /// Metodo que plota curvas de IPR
480     void Plotar();
481
482     /// Plotar dados de arquivo de disco
483     void PlotarDadosDisco();
484
485     /// Metodo que salva grafico
486     void SalvarGrafico();
487
488     /// Metodos relacionado ao calculo da vazao com reservatorio em
        regime transiente
489     void RegimeTransiente();
490     /// Metodos relacionado ao calculo da vazao com reservatorio em
        regime pseudopermanente
491     void RegimePseudoPermanente();
492     /// Metodos relacionado ao calculo da vazao com reservatorio em
        regime permanente
493     void RegimePermanente();
494     /// Tempo final do reservatorio em regime transiente
495     double TPSS(double _phi, double _mi, double _ct, double _re,
        double _kh);
496     /// Tempo de producao do reservatorio em regime transiente
497     void Tempo(double _tempo){tempo = _tempo;};
498     /// Metodo para calculo do skin total
499     double SkinTotal();
500
501}; //< encerra class CSimuladorCurvasIPR
502
503#endif

```

Apresenta-se na listagem 6.12 o arquivo de implementação da classe CSimuladorCurvasIPR.

Listing 6.12: Arquivo de implementação da classe CSimuladorCurvasIPR.

```

504/// Implementacao da classe CSimuladorCurvasIPR
505
506/// Inclusao de bibliotecas
507#include<iostream>        //< Biblioteca usada para iteracao tela e
        teclado
508#include<fstream>         //< Biblioteca para entrada e saida de disco
509#include<iomanip>          //< Biblioteca para a formatacao da saida de
        dados para tela e arquivo de disco
510#include<string>          //< Biblioteca para o uso de strings
511#include<vector>           //< Biblioteca para o uso de vetores

```

```

512#include<cmath>           //< Biblioteca para o uso de funções matemáticas
513#include<cstdlib>         //< Biblioteca usada para a saída do programa e
                             limpeza de dados da tela
514#include "CGnuplot.h"     //< Biblioteca usada para a utilização de
                             recursos gráficos usando o software externo GnuPlot
515
516/// Inclusão das demais classes
517#include "CSimuladorCurvasIPR.h"
518#include "CRocha.h"
519#include "CFluido.h"
520#include "CPoco.h"
521#include "CReservatorio.h"
522#include "CSkin.h"
523#include "CSkinDano.h"
524#include "CSkinCanhoneio.h"
525#include "CSkinDesvio.h"
526
527using namespace std;
528
529string linha = "
    -----
    ";
530
531/// Cabeçalho do software
532void CSimuladorCurvasIPR::Header(){
533
534    system("clear");
535    cout<<linha<<endl;
536    cout<< " | UNIVERSIDADE | ESTADUAL | DO | NORTE | FLUMINENSE | DARCY | RIBEIRO "<<
        endl;
537    cout<< " | CENTRO | DE | CIENCIAS | E | TECNOLOGIA | - | CTT "<<endl;
538    cout<< " | LABORATÓRIO | DE | ENGENHARIA | E | EXPLORAÇÃO | DE | PETRÓLEO | - | LENEP
        "<<endl;
539    cout<< " | DISCIPLINA | : | PROGRAMAÇÃO | PRÁTICA | - | PROJETO | C++ "<<endl;
540    cout<< " | PROFESSOR : | ANDRÉ | DUARTE | BUENO "<<endl;
541    cout<< " | ALUNO : | MATHEUS | CARVALHO | DUARTE "<<endl;
542    cout<< " | SOFTWARE | PARA | ANÁLISE | DO | EFEITO | DE | PELÍCULA | A | PARTIR | DE |
        CURVAS | DE | ÍNDICE | DE | PRODUTIVIDADE | (IPR) | EM | POÇOS | VERTICAIS | COM |
        FLUXO | MONOFÁSICO | DE | ÓLEO "<<endl;
543    cout<<linha<<endl;
544} //< encerra Header()
545
546/// Menu Inicial
547int CSimuladorCurvasIPR::Menu(){
548
549    cout << linha;
550    cout << "\nMenu de Inicialização" << endl;
551    cout<<"\nSelecione uma das opções abaixo: " << endl;

```

```

552
553     cout<<"\n-Entrada de dados pelo teclado
        .....1";
554     cout<<"\n-Entrada de dados por arquivo (.dat)
        .....2";
555     cout<<"\n-Visualizar dados de arquivo (.dat)
        .....3";
556     cout<<"\n-Salvar dados de pressao e vazao (.dat)
        .....4";
557     cout<<"\n-Plotar Curvas de IPR gerados pelo software
        .....5";
558     cout<<"\n-Plotar Curvas de IPR de um arquivo (.dat)
        .....6";
559     cout<<"\n-Salvar grafico (.png)
        .....7";
560     cout<<"\n-Sobre o programa
        .....8";
561     cout<<"\n-Sair
        .....9";
562     cout<< endl;
563
564     cout << "\nOpcao desejada: ";
565     short int opcao{1};
566     cin >> opcao;
567     cin.get();
568
569     if (opcao<=9){
570         switch (opcao){
571             case 1: EntradaDadosTeclado();          break;
572             case 2: EntradaDadosArquivoDisco();      break;
573             case 3: LerDadosArquivoDisco();          break;
574             case 4: SalvarDadosArquivoDisco();       break;
575             case 5: Plotar();                        break;
576             case 6: PlotarDadosDisco();              break;
577             case 7: SalvarGrafico();                 break;
578             case 8: InformacaoPrograma();            break;
579             case 9: exit(0);
580             default: EntradaDadosTeclado();          break;
581         } //< encerra switch
582     } //< encerra if
583     else
584     {
585         cerr << "\nOpcao invalida. Tente novamente." << endl << endl <<
            endl << endl << endl;
586         return Menu();
587     } //< encerra else
588
589     return 0;

```

```

590
591 } //< encerra Menu()
592
593 /// Entrada de dados pelo teclado
594 void CSimuladorCurvasIPR::EntradaDadosTeclado(){
595
596     cout << linha << endl;
597     cout<<"\n Selecione o regime do reservatorio: " << endl;
598     cout<<"\n - Regime Transiente
        .....1";
599     cout<<"\n - Regime Pseudopermanente
        .....2";
600     cout<<"\n - Regime Permanente
        .....3";
601     cout<<"\n - Voltar
        .....4";
602     cout<< endl;
603
604     cout << "\n Opcao desejada: ";
605     short int opcaoRegime;           //< retorna o regime selecionado
606     cin >> opcaoRegime;
607     cin.get();
608
609     if (opcaoRegime == 1){
610
611         /// Cabecalho para o regime transiente
612         system("clear");
613         cout << linha;
614         cout << "\n" << setw(90) << "Reservatorio em Regime Transiente\n";
615         cout << linha;
616
617         cout << "\n Informe a pressao inicial do reservatorio (psi): ";
618         double _pi;
619         cin >> _pi;
620         cin.get();
621         reservatorio.Pi(_pi);
622     };
623
624     if (opcaoRegime == 2){
625
626         /// Cabecalho para o regime pseudopermanente
627         system("clear");
628         cout << linha;
629         cout << "\n" << setw(90) << "Reservatorio em Regime
            Pseudopermanente\n";
630         cout << linha;
631
632         /// O nome da pressao muda de acordo com o regime (pressao inicial,

```

```

        media ou na fronteira), mas no programa um unico metodo recebe o
        valor
633
634     cout << "\nInforme a pressao media do reservatorio (psi): ";
635     double _pi;
636     cin >> _pi;
637     cin.get();
638     reservatorio.Pi(_pi);
639 };
640
641     if (opcaoRegime == 3){
642
643         /// Cabecalho para o regime permanente
644         system("clear");
645         cout << linha;
646         cout << "\n" << setw(90) << "Reservatorio em Regime Permanente\n";
647         cout << linha;
648
649         /// O nome da pressao muda de acordo com o regime ( pressao inicial,
        media ou na fronteira), mas no programa um unico metodo recebe o
        valor
650
651         cout << "\nInforme a pressao na fronteira do reservatorio (psi): ";
652         double _pi;
653         cin >> _pi;
654         cin.get();
655         reservatorio.Pi(_pi);
656     };
657
658     if (opcaoRegime == 4) {Menu();}
659
660     if (opcaoRegime > 4){
661         cerr << "\nOpcao invalida. Tente novamente." << endl << endl <<
            endl << endl << endl;
662         return EntradaDadosTeclado();
663     };
664
665     cout << "\nInforme a pressao final do poco (psi): ";
666     double _pwf;
667     cin >> _pwf;
668     cin.get();
669     poco.Pwf(_pwf);
670
671     /// A pressao final do poco deve ser menor que a pressao inicial
672     if ( poco.Pwf() > reservatorio.Pi() ) {cerr << "\nAtencao! A
        pressao do poco caiu ao longo do tempo de producao. Portanto, a
        pressao final deve ser menor que a pressao inicial. VocE sera
        redirecionado para o Menu anterior a fim de se redefinir os

```



```
        parmametros." << endl;
673     return EntradaDadosTeclado();
674 }
675
676
677     cout << "\n_Informe_a_variacao_de_pressao_em_que_ira_ser_medida_a_
        vazao_do_poco(psi):_";
678     double _deltaP;
679     cin >> _deltaP;
680     cin.get();
681     poco.DeltaP(_deltaP);
682
683     cout << "\n_Informe_o_raio_do_reservatorio(ft):_";
684     double _re;
685     cin >> _re;
686     cin.get();
687     reservatorio.Re(_re);
688
689     cout << "\n_Informe_o_raio_do_poco(ft):_";
690     double _rw;
691     cin >> _rw;
692     cin.get();
693     poco.Rw(_rw);
694
695     cout << "\n_Informe_o_raio_da_zona_alterada(ft):_";
696     double _rs;
697     cin >> _rs;
698     cin.get();
699     poco.Rs(_rs);
700
701     cout << "\n_Informe_a_inclinacao_do_poco(graus):_";
702     double _theta;
703     cin >> _theta;
704     cin.get();
705     poco.Theta(_theta);
706
707     cout << "\n_Informe_a_espessura_do_reservatorio(ft):_";
708     double _h;
709     cin >> _h;
710     cin.get();
711     reservatorio.H(_h);
712
713     cout << "\n_Informe_a_espessura_da_zona_canhoneada(ft):_";
714     double _hp;
715     cin >> _hp;
716     cin.get();
717     reservatorio.HP(_hp);
718
```

```

719     cout << "\nInforme a espessura da zona perfurada e nao-canhoneada (
        ft): ";
720     double _h1;
721     cin >> _h1;
722     cin.get();
723     reservatorio.H1(_h1);
724
725     cout << "\nInforme a permeabilidade horizontal (mD): ";
726     double _kh;
727     cin >> _kh;
728     cin.get();
729     rocha.Kh(_kh);
730
731     cout << "\nInforme a permeabilidade vertical (mD): ";
732     double _kv;
733     cin >> _kv;
734     cin.get();
735     rocha.Kv(_kv);
736
737     cout << "\nInforme a permeabilidade da zona alterada (mD): ";
738     double _ks;
739     cin >> _ks;
740     cin.get();
741     rocha.Ks(_ks);
742
743     cout << "\nInforme o fator volume-formacao do fluido (bbl/STB): ";
744     double _fvf;
745     cin >> _fvf;
746     cin.get();
747     oleo.FVF(_fvf);
748
749     cout << "\nInforme a viscosidade do fluido (cp): ";
750     double _mi;
751     cin >> _mi;
752     cin.get();
753     oleo.Mi(_mi);
754
755     /// Para o regime transiente sao necessarios mais parametros. Estes
        nao foram adicionados no primeiro if apenas para se manter a
        entrada de dados padronizada para todos os regimes.
756
757     if (opcaoRegime == 1){
758
759         cout << "\nInforme a porosidade do reservatorio: ";
760         double _phi;
761         cin >> _phi;
762         cin.get();
763         rocha.Phi(_phi);

```

```

764
765     cout << "\nInforme a compressibilidade total (1/psi): ";
766     double _ct;
767     cin >> _ct;
768     cin.get();
769     rocha.Ct(_ct);
770
771     /// O software informa ao usuario o tempo final do regime transiente
       . Este deve ser o tempo maximo que o usuario deve definir para
       que a solucao do regime transiente seja satisfeita.
772     cout << "\nTempo final do regime transiente (hr): " << TPSS(rocha.
        Phi(),oleo.Mi(),rocha.Ct(),reservatorio.Re(),rocha.Kh()) << ".\n"
        Informe o tempo de producao do poço (este deve ser menor ou igual
        ao tempo final): ";
773
774     double _tempo;
775     cin >> _tempo;
776     cin.get();
777
778     if (_tempo <= tpss){
779         Tempo(_tempo);
780     }
781     else {
782         cerr << "\nAtencao! Tempo maior que o de duracao do regime
            transiente. Insira tempo inferior ao limite ou selecione outro
            regime";
783         cout << endl << "\nPressione ENTER para retornar ao inicio." <<
            endl;
784         cin.get();
785         return EntradaDadosTeclado();
786     }; //< endif tpss
787
788     }; //< endif opcaoRegime
789
790     /// Calculo do skin total
791     SkinTotal();
792
793     /// Calculo da vazao em funcao da queda de pressao de acordo com
       regime definido pelo usuario.
794     switch (opcaoRegime){
795         case 1: RegimeTransiente();           break;
796         case 2: RegimePseudoPermanente();     break;
797         case 3: RegimePermanente();           break;
798     }; /// encerra switch
799
800 } //< encerra EntradaDadosTeclado()
801
802 /// Entrada de dados por aquivo de disco

```

```

803 void CSimuladorCurvasIPR::EntradaDadosArquivoDisco(){
804
805     cout << linha << endl;
806     cout<<"\n Selecione o regime do reservatorio: " << endl;
807     cout<<"\n - Regime Transiente
            .....1";
808     cout<<"\n - Regime Pseudopermanente
            .....2";
809     cout<<"\n - Regime Permanente
            .....3";
810     cout<<"\n - Voltar
            .....4";
811     cout<< endl;
812
813     cout << "\n Opcao desejada: ";
814     short int opcaoRegime;           //< retorna o regime selecionado
815     cin >> opcaoRegime;
816     cin.get();
817
818     cout << "\n O arquivo deve estar na pasta raiz do software. Informe
            o nome do arquivo com os parametros: ";
819     getline(cin, entradaDados);
820
821     ifstream fin (entradaDados + ".dat");
822     if (! fin){cerr << "\n Arquivo nao encontrado. Por favor, insira um
            nome de arquivo valido: ";
823                 getline(cin, entradaDados);
824                 fin.open(entradaDados + ".dat");}
825
826     string numero;
827     double data;
828     int contador = 0;
829
830     cout << linha;
831     cout << "\n Dados de entrada carregados pelo arquivo de disco: " <<
            endl;
832     cout << linha;
833
834     while (!fin.eof()){           //< leitura ate o final do arquivo de disco
835
836         contador++;
837         getline(fin, numero);           //< leitura do linha
838         data = atof(numero.c_str()); //< converte string para double
839
840         /// Switch armazena os dados nos devidos metodos e mostra na
            tela
841         /// Dados de entrada comuns a todos os regimes
842

```

```

843     switch(contador){
844     case 1:  cout << "\n_Pressao_inicial_(psi):_" << data;
              reservatorio.Pi(data);          break;
845     case 2:  cout << "\n_Pressao_final_do_poco_(psi):_" << data;
              poco.Pwf(data);                  break;
846     case 3:  cout << "\n_Variacao_de_pressao_(psi):_" << data; poco.
              DeltaP(data);                    break;
847     case 4:  cout << "\n_Raio_do_reservatorio_(ft):_" << data;
              reservatorio.Re(data);           break;
848     case 5:  cout << "\n_Raio_do_poco_(ft):_" << data; poco.Rw(data)
              ;                                break;
849     case 6:  cout << "\n_Raio_da_zona_alterada_(ft):_" << data; poco.
              Rs(data);                        break;
850     case 7:  cout << "\n_Inclinacao_do_poco_(graus):_" << data; poco
              .Theta(data);                    break;
851     case 8:  cout << "\n_Espessura_do_reservatorio_(ft):_" << data;
              reservatorio.H(data);            break;
852     case 9:  cout << "\n_Espessura_da_zona_canhoneada:_" << data;
              reservatorio.HP(data);           break;
853     case 10: cout << "\n_Espessura_da_zona_perfurada_e_nao-
              canhoneada_(ft):_" << data; reservatorio.H1(data);      break;
854     case 11: cout << "\n_Permabilidade_horizontal_(mD):_" << data;
              rocha.Kh(data);                  break;
855     case 12: cout << "\n_Permabilidade_vertical_(mD):_" << data;
              rocha.Kv(data);                  break;
856     case 13: cout << "\n_Permabilidade_da_zona_alterada_(mD):_" <<
              data; rocha.Ks(data);            break;
857     case 14: cout << "\n_Fator_volume-formacao_(bbl/STB):_" << data;
              oleo.FVF(data);                  break;
858     case 15: cout << "\n_Viscosidade_(cp):_" << data; oleo.Mi(data);
              break;
859
860     /// Para o regime transiente estes dados sao obrigatorios
861     case 16: cout << "\n_Porosidade_do_reservatorio:_" << data;
              rocha.Phi(data);                 break;
862     case 17: cout << "\n_Compressibilidade_total_(1/psi):_" << data;
              rocha.Ct(data);                  break;
863     case 18: cout << "\n_Tempo_de_producao_(hr):_" << data; Tempo(
              data);                            break;
864     } //< end switch
865 } //< end while
866
867 fin.close();    //< fecha arquivo de disco
868 cout << endl << "\n_Para_continuar_pressione_ENTER" << endl;
869 cin.get();
870
871 /// Calculo do skin total
872 SkinTotal();

```

```

873
874     /// Calculo da vazao em funcao da queda de pressao de acordo com
875         regime definido pelo usuario.
876     switch (opcaoRegime){
877     case 1: RegimeTransiente();           break;
878     case 2: RegimePseudoPermanente();     break;
879     case 3: RegimePermanente();           break;
880     };//< encerra switch
881 } //< encerra EntradaDadosArquivoDisco()
882
883 /// Visualizar dados de arquivo de disco
884 void CSimuladorCurvasIPR::LerDadosArquivoDisco(){
885
886     cout << "\n_o_arquivo_deve_estar_na_pasta_raiz_do_software._Informe_o_nome_do_arquivo: ";
887     getline(cin, entradaDados);
888
889     ifstream fin (entradaDados + ".dat");
890     if (! fin){cerr << "\n_Arquivo_nao_encontrado._Por_favor,_insira_um_nome_de_arquivo_valido: ";
891             getline(cin, entradaDados);
892             fin.open(entradaDados + ".dat");}
893
894     string data;
895
896     cout << endl;
897     while (!fin.eof()){
898         getline(fin, data);           ///< Leitura de cada linha do arquivo
899         cout << data << endl;       ///< Mostra na tela
900     }//< end while
901
902     cin.get();
903
904 } //< encerra LerDadosArquivoDisco()
905
906 /// Salvar dados em arquivo de disco
907 void CSimuladorCurvasIPR::SalvarDadosArquivoDisco(){
908
909     cout << "\n_Informe_o_nome_que_deseja_salvar_o_arquivo: ";
910     getline(cin, nomeArquivo);
911     ofstream fout (nomeArquivo + ".dat");
912
913     /// Cabecalho do arquivo
914     fout << setw(40) << right << "Pressao(psi)" << setw(40) << right <<
        "Vazao_c/_skin_(STB/day)" << setw(40) << right << "Vazao_s/_skin_(STB/day)" << endl;
915

```

```

916     /// Saída de Dados
917     cout.precision(4); //< Numero de algarismos significativos
918     for (int i=0; i<pressaoPoco.size();i++){
919         fout << setw(40) << right << pressaoPoco[i] << setw(40) << right
920             << vazaoPocoSkin[i] << setw(40) << right <<
921             vazaoPocoNullSkin[i]<<endl;
922     }
923     fout.close(); /// fecha arquivo de disco
924     cout << "\nDados salvos em arquivo com sucesso.\n" << linha;
925
926     cin.get();
927
928 } //< encerra SalvarDadosArquivoDisco()
929
930 /// Metodo para plot das curvas de IPR
931 void CSimuladorCurvasIPR::Plotar(){
932
933     grafico.ResetPlot(); //< reseta ultimo plot aberto
934     grafico.Style("lines"); //< grafico em linha
935     grafico.PlotVector(vazaoPocoNullSkin,pressaoPoco,"Skin_nulo"); //<
936         plot de vetores
937     grafico.PlotVector(vazaoPocoSkin,pressaoPoco,"Skin_ nao-nulo"); //<
938         plot de vetores
939     grafico.Title("Analise do Efeito de Pelicula a partir de curvas de
940         IPR"); //< titulo do grafico
941     grafico.XLabel("Vazao (STB/d)"); //< titulo eixo x
942     grafico.YLabel("Pressao (psi)"); //< titulo eixo y
943     grafico.Grid(1); //< ativa grid
944     cin.get();
945
946 } //< encerra Plotar()
947
948 /// Plotar grafico de arquivo de disco
949 void CSimuladorCurvasIPR::PlotarDadosDisco(){
950
951     cout << "\nO arquivo deve estar na pasta raiz do software. Informe
952         o nome do arquivo: ";
953     getline(cin, nomeArquivo);
954
955     ifstream fin (nomeArquivo + ".dat");
956     if (! fin){cerr << "\nArquivo de disco nao encontrado. Por favor,
957         insira um nome de arquivo valido: ";
958         getline(cin, nomeArquivo);
959         fin.open(nomeArquivo + ".dat");}
960
961     grafico.ResetPlot(); //< reseta ultimo plot aberto
962     grafico.Style("lines"); //< grafico em linha
963     grafico.PlotFile(nomeArquivo + ".dat",3,1,"skin_nulo"); //< plot de

```

```

        y(coluna 1) versus x (coluna 3)
957   grafico.PlotFile(nomeArquivo + ".dat", 2, 1, "skin_nao-nulo"); //< plot
        de y (coluna 1) versus x (coluna 2)
958   grafico.Title("Analise do Efeito de Pelicula a partir de curvas de
        IPR"); //< titulo do grafico
959   grafico.XLabel("Vazao (STB/d)"); //< titulo eixo x
960   grafico.YLabel("Pressao (psi)"); //< titulo eixo y
961   grafico.Grid(1); //< ativa grid
962
963   fin.close(); //< fecha o arquivo de disco
964
965   cin.get();
966
967} //< encerra PlotarDadosDisco()
968
969/// Salvar grafico
970void CSimuladorCurvasIPR::SalvarGrafico(){
971
972   cout << "\n Informe o nome do arquivo que deseja salvar o grafico: ";
973   ;
974   getline(cin, nomeGrafico);
975   grafico.savetopng(nomeGrafico);
976   cout << "\n Grafico salvo com sucesso.";
977   cin.get();
978} //< encerra SalvarGrafico()
979
980/// Informacoes sobre o programa para usuario iniciante
981void CSimuladorCurvasIPR::InformacaoPrograma(){
982
983   cout << linha;
984   cout << "\n Este programa analisa o efeito de pelicula a partir de
        curvas de indice de Produtividade (IPR) em pocos verticais com
        fluxo monofasico de oleo." << endl;
985   cout << "\n A partir dos parametros de entrada realiza-se uma
        estimativa dos efeitos de pelicula." << endl;
986   cout << "\n Os efeitos de pelicula calculados sao:" << endl;
987   cout << "\n - Efeito de pelicula devido alteracao da permeabilidade
        nas proximidades do poco calculado pela Formula de Hawkins (1956)
        ." << endl;
988   cout << "\n - Efeito de pelicula causado pelo canhoneio parcial do
        reservatorio a partir de formulas propostas por Papatzacos (1987)
        ." << endl;
989   cout << "\n - Efeito de pelicula causado pela inclinacao do poco a
        partir de formulas propostas por Besson (1990)." << endl;
990   cout << "\n - Efeito de pelicula total (soma dos anteriores)." <<
        endl;
991   cout << "\n A partir destas estimativas o programa calcula o vazao

```



```

        no_fundo_do_poco_para_pocos_verticais_de_oleo_em_regime_
        transiente, pseudopermanente ou permanente (definido pelo usuario
        ) considerando efeito de película nulo e nao-nulo para avaliar se
        ha um dano ou estimulo a formacao." << endl;
992 cout << "\n0_usuario podera ler, salvar e visualizar os dados em '.
        dat', plotar graficos e salva-los em '.png' para melhor analise
        do efeito de película total." << endl << linha;
993 cin.get();
994
995 } //< encerra InformacaoPrograma()
996
997 /// Implementacao dos Regimes do Reservatorio
998
999 void CSimuladorCurvasIPR::RegimeTransiente(){
1000
1001     int num_medidas = (reservatorio.Pi() - poco.Pwf()) / poco.DeltaP() ;
        //< numero de medidas de vazao a serem feitas
1002
1003     /// Limpeza dos vetores
1004     pressaoPoco.clear();
1005     vazaoPocoSkin.clear();
1006     vazaoPocoNullSkin.clear();
1007
1008     for (int i = 0; i <= num_medidas; i++){
1009
1010         pressao = reservatorio.Pi() - i*poco.DeltaP(); //<
        Queda de pressao medida no poco
1011
1012         pressaoPoco.push_back(pressao); //<
        Armazena queda de pressao
1013
1014         /// Vazao considerando skin
1015         vazaoSkin = (rocha.Kh()*reservatorio.H()*(reservatorio.Pi() -
        pressaoPoco[i]))/(162.6*oleo.FVF()*oleo.Mi()*(log10(tempo) +
        log10(rocha.Kh()/(rocha.Phi()*oleo.Mi()*rocha.Ct()*pow(poco.
        Rw(),2))) - 3.23 + 0.87*skinTotal));
1016
1017         vazaoPocoSkin.push_back(vazaoSkin);
1018
1019         /// Vazao considerando skin nulo
1020         vazaoSkin = (rocha.Kh()*reservatorio.H()*(reservatorio.Pi() -
        pressaoPoco[i]))/( 162.6*oleo.FVF()*oleo.Mi()*(log10(tempo) +
        log10(rocha.Kh()/(rocha.Phi()*oleo.Mi()*rocha.Ct()*pow(poco.
        Rw(),2))) -3.23));
1021
1022         vazaoPocoNullSkin.push_back(vazaoSkin);
1023     };
1024

```

```

1025     /// Retorna medidas na tela
1026     cout << endl;
1027     cout << linha << endl;
1028     cout << setw(40) << right << "Pressao_(psi)" << setw(40) << right <<
        "Vazao_com_Skin_(STB/day)" << setw(40) << right << "Vazao_sem_
        Skin_(STB/day)" << endl;
1029     cout << linha << endl;
1030
1031     cout.precision(4); //< Numero de algarismos significativos
1032     for (int i = 0; i <= num_medidas; i++){
1033         cout << setw(40) << right << pressaoPoco[i] << setw(40) << right
            << vazaoPocoSkin[i] << setw(40) << right <<
            vazaoPocoNullSkin[i] << endl;
1034     };
1035     cout << linha;
1036
1037 } /// encerra RegimeTransiente()
1038
1039 void CSimuladorCurvasIPR::RegimePseudoPermanente(){
1040
1041     int num_medidas = (reservatorio.Pi() - poco.Pwf()) / poco.DeltaP() ;
        // numero de medidas de vazao a serem feitas
1042
1043     /// Limpeza dos vetores
1044     pressaoPoco.clear();
1045     vazaoPocoSkin.clear();
1046     vazaoPocoNullSkin.clear();
1047
1048     for (int i = 0; i <= num_medidas; i++){
1049
1050         pressao = reservatorio.Pi() - i*poco.DeltaP(); //<
            Queda de pressao medida no poco
1051
1052         pressaoPoco.push_back(pressao); //<
            Armazena queda de pressao
1053
1054         /// Vazao considerando skin
1055         vazaoSkin = (rocha.Kh()*reservatorio.H()*(reservatorio.Pi() -
            pressaoPoco[i]))/(141.2*oleo.FVF()*oleo.Mi()*(log(
            reservatorio.Re()/poco.Rw()) - (3./4.) + skinTotal));
1056
1057         vazaoPocoSkin.push_back(vazaoSkin);
1058
1059         /// Vazao considerando skin nulo
1060         vazaoSkin = (rocha.Kh()*reservatorio.H()*(reservatorio.Pi() -
            pressaoPoco[i]))/(141.2*oleo.FVF()*oleo.Mi()*(log(
            reservatorio.Re()/poco.Rw()) - (3./4.)));;
1061

```

```

1062     vazaoPocoNullSkin.push_back(vazaoSkin);
1063 };
1064
1065     /// Retorna medidas na tela
1066     cout << endl;
1067     cout << linha << endl;
1068     cout << setw(40) << right << "Pressao_(psi)" << setw(40) << right <<
        "Vazao_com_Skin_(STB/day)" << setw(40) << right << "Vazao_sem_
        Skin_(STB/day)" << endl;
1069     cout << linha << endl;
1070
1071     cout.precision(4); //< Numero de algarismos significativos
1072     for (int i = 0; i <= num_medidas; i++){
1073         cout << setw(40) << right << pressaoPoco[i] << setw(40) << right
            << vazaoPocoSkin[i] << setw(40) << right <<
            vazaoPocoNullSkin[i] << endl;
1074     };
1075     cout << linha;
1076
1077 } /// encerra RegimePseudoPermanente()
1078
1079 void CSimuladorCurvasIPR::RegimePermanente(){
1080
1081     int num_medidas = (reservatorio.Pi() - poco.Pwf()) / poco.DeltaP() ;
        //< numero de medidas de vazao a serem feitas
1082
1083     /// Limpeza dos vetores
1084     pressaoPoco.clear();
1085     vazaoPocoSkin.clear();
1086     vazaoPocoNullSkin.clear();
1087
1088     for (int i = 0; i <= num_medidas; i++){
1089
1090         pressao = reservatorio.Pi() - i*poco.DeltaP(); //<
            Queda de pressao medida no poco
1091
1092         pressaoPoco.push_back(pressao); //<
            Armazena queda de pressao
1093
1094         /// Vazao considerando skin
1095         vazaoSkin = (rocha.Kh()*reservatorio.H()*(reservatorio.Pi() -
            pressaoPoco[i]))/ (141.2*oleo.FVF()*oleo.Mi()*(log(
            reservatorio.Re()/poco.Rw()) + skinTotal));
1096
1097         vazaoPocoSkin.push_back(vazaoSkin);
1098
1099         /// Vazao considerando skin nulo
1100         vazaoSkin = (rocha.Kh()*reservatorio.H()*(reservatorio.Pi() -

```

```

        pressaoPoco[i]))/          (141.2*oleo.FVF()*oleo.Mi()*(log(
        reservatorio.Re()/poco.Rw())));
1101
1102     vazaoPocoNullSkin.push_back(vazaoSkin);
1103 };
1104
1105     /// Retorna medidas na tela
1106     cout << endl;
1107     cout << linha << endl;
1108     cout << setw(40) << right << "Pressao_(psi)" << setw(40) << right <<
        "Vazao_com_Skin_(STB/day)" << setw(40) << right << "Vazao_sem_
        Skin_(STB/day)" << endl;
1109     cout << linha << endl;
1110
1111     cout.precision(4); //< Numero de algarismos significativos
1112     for (int i = 0; i <= num_medidas; i++){
1113         cout << setw(40) << right << pressaoPoco[i] << setw(40) << right
            << vazaoPocoSkin[i] << setw(40) << right <<
            vazaoPocoNullSkin[i] << endl;
1114     };
1115     cout << linha;
1116
1117 } //< encerra RegimePermanente()
1118
1119 /// Metodo que retorna o tempo limite para o regime transiente
1120 double CSimuladorCurvasIPR::TPSS(double _phi, double _mi, double _ct,
        double _re, double _kh){
1121     tpss = (1200*_phi*_mi*_ct*pow(_re,2))/_kh;
1122     return tpss;
1123 } //< encerra TPSS
1124
1125 /// Metodo que retorna o calculo do skin total
1126 double CSimuladorCurvasIPR::SkinTotal(){
1127
1128     /// Skin de dano
1129     danoSkin.Skin(rocha.Kh(),rocha.Ks(),poco.Rw(), poco.Rs());
1130
1131
1132     /// Skin por canhoneio parcial
1133     canhoneioSkin.HPD(reservatorio.HP(),reservatorio.H());
1134     canhoneioSkin.H1D(reservatorio.H1(),reservatorio.H());
1135     canhoneioSkin.RD(poco.Rw(),reservatorio.H(),rocha.Kv(),rocha.Kh());
1136     canhoneioSkin.A();
1137     canhoneioSkin.B();
1138     canhoneioSkin.Skin(poco.Rw(),reservatorio.H(),reservatorio.HP(),
        rocha.Kv(),rocha.Kh());
1139
1140     /// Skin por desvio do poco

```

```

1141     desvioSkin.Beta(rocha.Kv(),rocha.Kh());
1142     poco.Extensao(reservatorio.H());
1143     desvioSkin.Gamma(reservatorio.H(),poco.Extensao(reservatorio.H()));
1144     desvioSkin.Skin(poco.Rw(),reservatorio.H(),poco.Extensao(
        reservatorio.H()));
1145
1146     /// Skin total
1147     skinTotal = danoSkin.Skin(rocha.Kh(),rocha.Ks(),poco.Rw(), poco.Rs())
        + canhoneioSkin.Skin(poco.Rw(),reservatorio.H(),reservatorio.HP
        (),rocha.Kv(),rocha.Kh()) + desvioSkin.Skin(poco.Rw(),
        reservatorio.H(),poco.Extensao(reservatorio.H()));
1148
1149     cout << endl << linha << "\n_Calculo_dos_efeitos_de_película" <<
        endl << linha;
1150     cout << "\n_Efeito_de_película_por_dano_a_permeabilidade_nas_
        proximidades_do_poco:" << danoSkin.Skin(rocha.Kh(),rocha.Ks(),
        poco.Rw(), poco.Rs());
1151     cout << "\n_Efeito_de_película_causado_por_completacao_parcial_do_
        reservatorio:" << canhoneioSkin.Skin(poco.Rw(),reservatorio.H(),
        reservatorio.HP(),rocha.Kv(),rocha.Kh());
1152     cout << "\n_Efeito_de_película_causado_pela_inclinacao_do_poco:" <<
        desvioSkin.Skin(poco.Rw(),reservatorio.H(),poco.Extensao(
        reservatorio.H()));
1153     cout << "\n_Efeito_de_película_total:" << skinTotal;
1154
1155     if (skinTotal > 0){
1156         cout << "\n_Efeito_de_película_resultante_causa_dano_a_formacao(
            skin_total_positivo)." << endl;
1157     }
1158     else{
1159         cout << "\n_Efeito_de_película_resultante_causa_estimulo_a_
            formacao(skin_total_negativo)." << endl;
1160     }
1161
1162     cout << "\n_Pressione_ENTER_para_continuar";
1163     cin.get();
1164
1165     return skinTotal;
1166 } //< encerra SkinTotal

```

Apresenta-se na listagem 6.13 a implementação da função main().

Listing 6.13: Arquivo de implementação da função main().

```

1168 /// Implementação da função main()
1169
1170 /// Inclusão de bibliotecas
1171 #include<string>
1172 #include<vector>
1173 #include<cmath>

```

```

1174
1175 /// Inclusão das demais classes
1176 #include "CSimuladorCurvasIPR.h"
1177
1178 int main (){
1179
1180     CSimuladorCurvasIPR simulador; //< Objeto da classe
        CSimuladorCurvasIPR
1181
1182     /// Primeiro acesso ao rodar o software
1183     simulador.Header();
1184     simulador.Menu();
1185
1186     short int opcaoMenu{1};
1187
1188     while (opcaoMenu == 1){
1189         cout << "\n_Deseja_retornar_ao_Menu_Inicial?";
1190         cout << "\n_Digite_1_para_SIM";
1191         cout << "\n_Digite_2_para_NÃO";
1192         cout << "\n_Opção_desejada:_";
1193         cin >> opcaoMenu;
1194         cin.get();
1195         if (opcaoMenu == 1){simulador.Menu();}
1196         cout << endl;
1197     }
1198     return 0;
1199 } //< Fim da função main()

```

Apresenta-se na listagem 6.14 a implementação do arquivo Makefile.

Listing 6.14: Arquivo de implementação do arquivo Makefile.

```

1200 # Arquivo makefile
1201
1202 all: simulador teste
1203
1204 simulador:
1205     g++ -std=c++11 -c CSimuladorCurvasIPR.cpp
1206     g++ -std=c++11 -c CGnuplot.cpp
1207     g++ -std=c++11 -c CSkinCanhoneio.cpp
1208     g++ -std=c++11 -c CSkinDesvio.cpp
1209     ar -cr libSimuladorCurvasIPR.a CSimuladorCurvasIPR.o CGnuplot.o
        CSkinCanhoneio.o CSkinDesvio.o
1210
1211 teste: simulador
1212     g++ -std=c++11 Simulador.teste.cpp libSimuladorCurvasIPR.a -o
        Simulador.teste
1213
1214 clean:
1215     rm *.o *.a Simulador.teste

```

```
1216
1217 help :
1218 #===== Sistema Simulador de Curvas IPR =====
1219 # A classe CSimuladorCurvasIPR representa a simulação de curvas IPR
1220 # A classe CSkinCanhoneio representa o conceito de skin causado por
      canhoneio parcial
1221 # A classe CSkinDesvio representa o conceito de skin causado pela
      inclinação do poço
1222 #
1223 # Uso do make :
1224 # all : compila todos os arquivos
1225 # simulador : compila arquivos da hierarquia CSimuladorCurvasIPR e gera
      libSimuladorCurvasIPR.a
1226 # teste : compila arquivo Simulador.teste.cpp anexa as libs e gera
      executável Simulador.teste
1227 # clean : apaga arquivos ( rm *.o *.a Simulador.teste )
1228 # help : mostra esta ajuda
```

Capítulo 7

Teste

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

7.1 Teste 1: Entrada de dados pelo teclado

Neste primeiro teste, foi escolhido a entrada de dados pelo teclado (Figura 7.1), onde o usuário deverá digitar os dados solicitados. O teste foi realizado para os regimes transiente (Figura 7.2), pseudopermanente (Figura 7.3) e permanente (Figura 7.4). Vale ressaltar que a quantidade de parâmetros exigida para o regime transiente é maior, sendo a entrada de dados mais extensa comparada aos outros dois regimes. Destaca-se ainda que os testes foram realizados usando-se valores iguais para um mesmo parâmetro com intuito de demonstrar que a estimativa do efeito de película não depende do regime selecionado (Figura 7.5, 7.6 e 7.7). Entretanto, observa-se mudança na vazão de fundo uma vez que a solução da E.D.H. é diferente para cada regime.

```
UNIVERSIDADE ESTADUAL DO NORTE FLUMINENSE DARCY RIBEIRO
CENTRO DE CIÊNCIAS E TECNOLOGIA - CTT
LABORATÓRIO DE ENGENHARIA E EXPLORAÇÃO DE PETRÓLEO - LENEP
DISCIPLINA: PROGRAMAÇÃO PRÁTICA - PROJETO C++
PROFESSOR: ANDRÉ DUARTE BUENO
ALUNO: MATHEUS CARVALHO DUARTE
SOFTWARE PARA ANÁLISE DO EFEITO DE PELÍCULA A PARTIR DE CURVAS DE ÍNDICE DE PRODUTIVIDADE (IPR) EM POCOS VERTICAIS COM FLUXO MONOFÁSICO DE ÓLEO

Menu de Inicialização

Selecione uma das opcoes abaixo:

- Entrada de dados pelo teclado.....1
- Entrada de dados por arquivo (.dat).....2
- Visualizar dados de arquivo (.dat).....3
- Salvar dados de pressão e vazão (.dat).....4
- Plotar Curvas de IPR gerados pelo software.....5
- Plotar Curvas de IPR de um arquivo (.dat).....6
- Salvar gráfico (.png).....7
- Sobre o programa.....8
- Sair.....9

Opção desejada: 1
```

Figura 7.1: Tela do programa mostrando o menu de inicialização e a entrada de dados pelo teclado


```
Reservatório em Regime Transiente
Informe a pressão inicial do reservatório (psi): 3000
Informe a pressão final do poço (psi): 2500
Informe a variação de pressão em que irá ser medida a vazão do poço (psi): 50
Informe o raio do reservatório (ft): 1000
Informe o raio do poço (ft): 0.4
Informe o raio da zona alterada (ft): 1.4
Informe a inclinação do poço (graus): 25
Informe a espessura do reservatório (ft): 80
Informe a espessura da zona canhoneada (ft): 40
Informe a espessura da zona perfurada e não-canhoneada (ft): 0
Informe a permeabilidade horizontal (mD): 20
Informe a permeabilidade vertical (mD): 20
Informe a permeabilidade da zona alterada (mD): 2
Informe o fator volume-formação do fluido (bbl/STB): 1.25
Informe a viscosidade do fluido (cp): 2.5
Informe a porosidade do reservatório: 0.2
Informe a compressibilidade total (1/psi): 0.000013
Tempo final do regime transiente (hr): 390. Informe o tempo de produção do poço (este deve ser menor ou igual ao tempo final): 200
```

Figura 7.2: Tela do programa mostrando a entrada de dados pelo teclado para o regime transiente

```
Reservatório em Regime Pseudopermanente
Informe a pressão média do reservatório (psi): 3000
Informe a pressão final do poço (psi): 2500
Informe a variação de pressão em que irá ser medida a vazão do poço (psi): 50
Informe o raio do reservatório (ft): 1000
Informe o raio do poço (ft): 0.4
Informe o raio da zona alterada (ft): 1.4
Informe a inclinação do poço (graus): 25
Informe a espessura do reservatório (ft): 80
Informe a espessura da zona canhoneada (ft): 40
Informe a espessura da zona perfurada e não-canhoneada (ft): 0
Informe a permeabilidade horizontal (mD): 20
Informe a permeabilidade vertical (mD): 20
Informe a permeabilidade da zona alterada (mD): 2
Informe o fator volume-formação do fluido (bbl/STB): 1.25
Informe a viscosidade do fluido (cp): 2.5
```

Figura 7.3: Tela do programa mostrando a entrada de dados pelo teclado para o regime pseudopermanente

```

Reservatório em Regime Permanente
Informe a pressão na fronteira do reservatório (psi): 3000
Informe a pressão final do poço (psi): 2500
Informe a variação de pressão em que irá ser medida a vazão do poço (psi): 50
Informe o raio do reservatório (ft): 1000
Informe o raio do poço (ft): 0.4
Informe o raio da zona alterada (ft): 1.4
Informe a inclinação do poço (graus): 25
Informe a espessura do reservatório (ft): 80
Informe a espessura da zona canhoneada (ft): 40
Informe a espessura da zona perfurada e não-canhoneada (ft): 0
Informe a permeabilidade horizontal (mD): 20
Informe a permeabilidade vertical (mD): 20
Informe a permeabilidade da zona alterada (mD): 2
Informe o fator volume-formação do fluido (bbl/STB): 1.25
Informe a viscosidade do fluido (cp): 2.5

```

Figura 7.4: Tela do programa mostrando a entrada de dados pelo teclado para o regime permanente

```

Cálculo dos efeitos de película
Efeito de película por dano à permeabilidade nas proximidades do poço: 11.2749
Efeito de película causado por completção parcial do reservatório: 3.96608
Efeito de película causado pela inclinação do poço: -0.420298
Efeito de película total: 14.8206
Efeito de película resultante causa dano à formação (skin total positivo).
Pressione ENTER para continuar

```

Pressao (psi)	Vazao com Skin (STB/day)	Vazao sem Skin (STB/day)
3000	0	0
2950	8.179	24.77
2900	16.36	49.55
2850	24.54	74.32
2800	32.72	99.1
2750	40.9	123.9
2700	49.08	148.6
2650	57.25	173.4
2600	65.43	198.2
2550	73.61	223
2500	81.79	247.7

Figura 7.5: Tela do programa mostrando o resultado do efeito de película e da vazão de fundo para o regime transiente

```

Cálculo dos efeitos de película
Efeito de película por dano à permeabilidade nas proximidades do poço: 11.27
Efeito de película causado por completção parcial do reservatório: 3.966
Efeito de película causado pela inclinação do poço: -0.4203
Efeito de película total: 14.82
Efeito de película resultante causa dano à formação (skin total positivo).
Pressione ENTER para continuar

```

Pressao (psi)	Vazao com Skin (STB/day)	Vazao sem Skin (STB/day)
3000	0	0
2950	8.281	25.63
2900	16.56	51.26
2850	24.84	76.89
2800	33.12	102.5
2750	41.4	128.1
2700	49.68	153.8
2650	57.96	179.4
2600	66.25	205
2550	74.53	230.7
2500	82.81	256.3

Figura 7.6: Tela do programa mostrando o resultado do efeito de película e da vazão de fundo para o regime pseudopermanente

Cálculo dos efeitos de película

Efeito de película por dano à permeabilidade nas proximidades do poço: 11.27
 Efeito de película causado por completação parcial do reservatório: 3.966
 Efeito de película causado pela inclinação do poço: -0.4203
 Efeito de película total: 14.82
 Efeito de película resultante causa dano à formação (skin total positivo).

Pressione ENTER para continuar

Pressao (psi)	Vazao com Skin (STB/day)	Vazao sem Skin (STB/day)
3000	0	0
2950	8.006	23.17
2900	16.01	46.35
2850	24.02	69.52
2800	32.03	92.69
2750	40.03	115.9
2700	48.04	139
2650	56.04	162.2
2600	64.05	185.4
2550	72.06	208.6
2500	80.06	231.7

Figura 7.7: Tela do programa mostrando o resultado do efeito de película e da vazão de fundo para o regime permanente

7.2 Teste 2: Entrada de dados por arquivo de disco

No segundo teste, foi escolhido a entrada de dados por arquivo de disco, onde o usuário deverá informar o nome do arquivo em que os dados se encontram. No exemplo mostrando abaixo, foi criado o arquivo *input.dat* (Figura 7.8). O programa inicialmente lê o arquivo e mostra os valores que estão sendo atribuídos a cada parâmetro para que o usuário possa verificar e confirmar se os dados de entrada estão corretos (Figura 7.9). A saída de dados é mostrada na Figura 7.10.

```

3000 // Pi(psi) - pressao inicial
2500 // Pwf(psi) - pressao final
50 // DeltaP(psi) - variacao de pressao
1000 // re(ft) - raio do reservatorio
0.4 // rw(ft) - raio do poco
1.4 // rs(ft) - raio da zona atingida
25 // theta(graus) - inclinacao do poco
80 // h(ft) - espessura do reservatorio
40 // hp(ft) - espessura da zona canhoneada
0 // h1(ft) - espessura da zona perfurada e nao-
canhoneada
20 // kh(mD) - permeabilidade horizontal
20 // kv(mD) - permeabilidade vertical
2 // ks(mD) - permeabilidade da zona alterada
1.25 // B(bbl/STB) - fator volume-formacao
2.5 // mi(cp) - viscosidade
0.2 // phi - porosidade
0.000013 // ct(1/psi) - compressibilidade total
200 // t(hr) - tempo de producao
  
```

Plain Text ▾ Tab Width: 8 ▾ Ln 18, Col 41 ▾ INS

Figura 7.8: Arquivo *input.dat* usado para a entrada de dados pelo arquivo de disco

```

Dados de entrada carregados pelo arquivo de disco:
-----
Pressao inicial (psi): 3000
Pressao final do poço (psi): 2500
Variação de pressão (psi): 50
Raio do reservatório (ft): 1000
Raio do poço (ft): 0.4
Raio da zona alterada (ft): 1.4
Inclinação do poço (graus): 25
Espessura do reservatório (ft): 80
Espessura da zona canhoneada: 40
Espessura da zona perfurada e não-canhoneada (ft): 0
Permeabilidade horizontal (mD): 20
Permeabilidade vertical (mD): 20
Permeabilidade da zona alterada (mD): 2
Fator volume-formação (bbl/STB): 1.25
Viscosidade (cp): 2.5
Porosidade do reservatório: 0.2
Compressibilidade total (1/psi): 1.3e-05
Tempo de produção (hr): 200

```

Figura 7.9: Dados carregados do arquivo de disco *input.dat*

```

Cálculo dos efeitos de peli cula
-----
Efeito de peli cula por dano à permeabilidade nas proximidades do poço: 11.27
Efeito de película causado por completação parcial do reservatório: 3.966
Efeito de película causado pela inclinação do poço: -0.4203
Efeito de película total: 14.82
Efeito de peli cula resultante causa dano à formação (skin total positivo).

```

Pressione ENTER para continuar

Pressao (psi)	Vazao com Skin (STB/day)	Vazao sem Skin (STB/day)
3000	0	0
2950	8.179	24.77
2900	16.36	49.55
2850	24.54	74.32
2800	32.72	99.1
2750	40.9	123.9
2700	49.08	148.6
2650	57.25	173.4
2600	65.43	198.2
2550	73.61	223
2500	81.79	247.7

Figura 7.10: Tela do programa mostrando o resultado do efeito de película e da vazão de fundo usando dados carregados pelo arquivo de disco *input.dat*

7.3 Teste 3: Visualizar dados de arquivo de disco

No teste 3, foi escolhido visualizar dados de arquivo de disco, onde o usuário poderá verificar o conteúdo do arquivo de dentro do programa, antes de utilizá-lo. Neste caso, o usuário poderá visualizar arquivo com dados de entrada, como o arquivo *input.dat*, ou um arquivo com dados de pressão e vazão como o mostrado a seguir (Figura 7.11). No exemplo abaixo, foi utilizado o arquivo *dadosIPR.dat* (Figura 7.12) contendo dados de pressão e vazão considerando skin nulo e não-nulo. A Figura 7.13 mostra a tela do programa com o conteúdo do arquivo *dadosIPR.dat*.

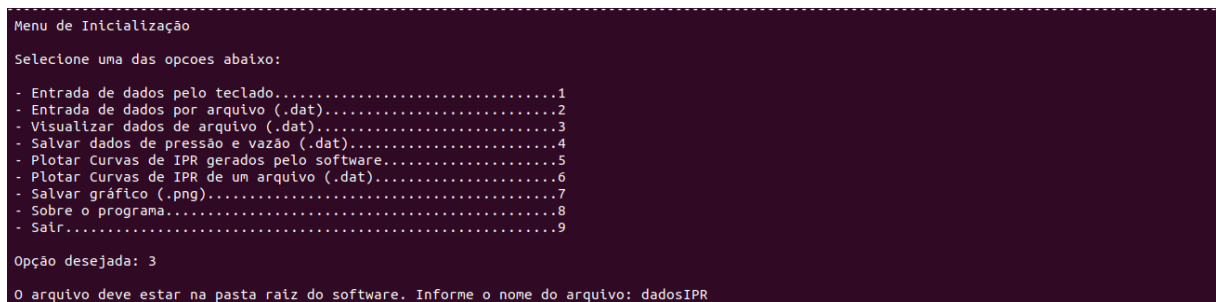


Figura 7.11: Tela do programa mostrando o usuário informando o arquivo *dadosIPR.dat* para visualização

Pressao(psi)	Vazao c/ skin (STB/day)	Vazao s/ skin (STB/day)
3000	0	0
2950	8.00643	23.1726
2900	16.0129	46.3451
2850	24.0193	69.5177
2800	32.0257	92.6902
2750	40.0321	115.863
2700	48.0386	139.035
2650	56.045	162.208
2600	64.0514	185.38
2550	72.0579	208.553
2500	80.0643	231.726

Figura 7.12: Arquivo de disco *dadosIPR.dat* para visualização

```

O arquivo deve estar na pasta raiz do software. Informe o nome do arquivo: dadosIPR

```

Pressao(psi)	Vazao c/ skin (STB/day)	Vazao s/ skin (STB/day)
3000	0	0
2950	8.00643	23.1726
2900	16.0129	46.3451
2850	24.0193	69.5177
2800	32.0257	92.6902
2750	40.0321	115.863
2700	48.0386	139.035
2650	56.045	162.208
2600	64.0514	185.38
2550	72.0579	208.553
2500	80.0643	231.726

Figura 7.13: Tela do programa mostrando o conteúdo do arquivo de disco *dadosIPR.dat*

7.4 Teste 4: Salvar dados em arquivo de disco

No teste 4, foi escolhido a opção de salvar os dados gerados em arquivo de disco (Figura 7.14). Neste exemplo, o arquivo salvo *dadosIPR2* recebe os dados de pressão e vazão considerando skin nulo e não-nulo (Figura 7.15).

```

Menu de Inicialização
Selecione uma das opcoes abaixo:
- Entrada de dados pelo teclado.....1
- Entrada de dados por arquivo (.dat).....2
- Visualizar dados de arquivo (.dat).....3
- Salvar dados de pressão e vazão (.dat).....4
- Plotar Curvas de IPR gerados pelo software.....5
- Plotar Curvas de IPR de um arquivo (.dat).....6
- Salvar gráfico (.png).....7
- Sobre o programa.....8
- Sair.....9

Opção desejada: 4
Informe o nome que deseja salvar o arquivo: dadosIPR2
Dados salvos em arquivo com sucesso.

```

Figura 7.14: Tela do programa mostrando a opção de salvar dados em arquivo de disco

dadosIPR2.dat ~/Documents/Software 2.6			
Pressao(psi)	Vazao c/ skin (STB/day)	Vazao s/ skin (STB/day)	
3000	0	0	
2950	8.17921	24.7743	
2900	16.3584	49.5487	
2850	24.5376	74.323	
2800	32.7168	99.0973	
2750	40.8961	123.872	
2700	49.0753	148.646	
2650	57.2545	173.42	
2600	65.4337	198.195	
2550	73.6129	222.969	
2500	81.7921	247.743	

Figura 7.15: Arquivo de disco *dadosIPR2* salvo na pasta raiz do programa

7.5 Teste 5: Plotar curvas de IPR

No teste 5, foi escolhido a opção de plotar curvas de IPR a partir dos dados de pressão e vazão calculados pelo programa (Figura 7.16). A Figura 7.17 mostra um gráfico gerado a partir do software externo Gnuplot plotando as curvas de IPR para um reservatório em regime permanente.

```

Menu de Inicialização
Selecione uma das opcoes abaixo:
- Entrada de dados pelo teclado.....1
- Entrada de dados por arquivo (.dat).....2
- Visualizar dados de arquivo (.dat).....3
- Salvar dados de pressão e vazão (.dat).....4
- Plotar Curvas de IPR gerados pelo software.....5
- Plotar Curvas de IPR de um arquivo (.dat).....6
- Salvar gráfico (.png).....7
- Sobre o programa.....8
- Sair.....9

Opção desejada: 5

```

Figura 7.16: Tela do programa mostrando a opção de plotar curvas de IPR a partir de dados gerados pelo software

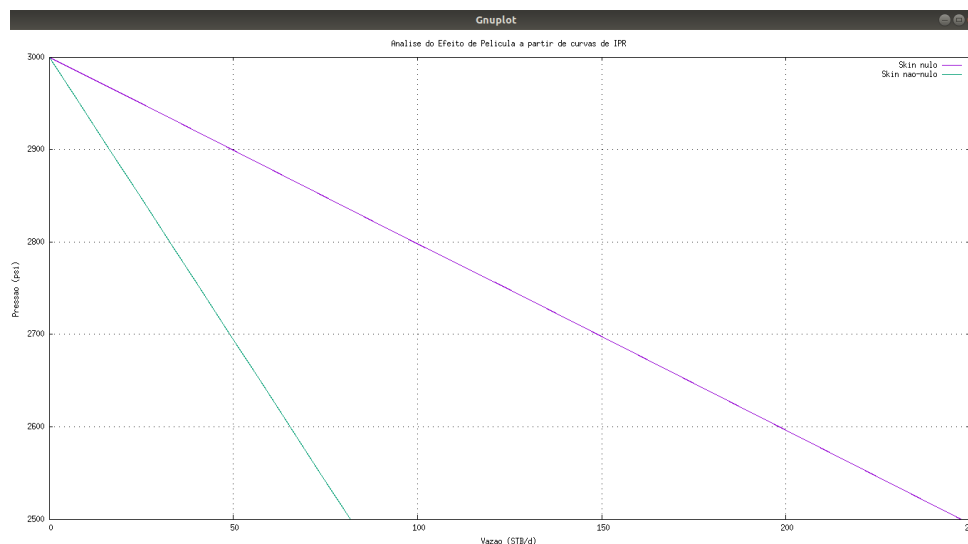


Figura 7.17: Software externo Gnuplot gerando gráfico de curvas de IPR de um reservatório em regime permanente

7.6 Teste 6: Plotar curvas de IPR de um arquivo de disco

No teste 6, foi escolhido a opção de plotar curvas de IPR a partir de um arquivo de disco contendo dados calculados pelo software. Esta opção é fornecida ao usuário para que o mesmo não precise informar ao programa todos os parâmetros novamente, uma vez que há um arquivo de disco com os dados salvos. O arquivo de disco informado pelo usuário é o arquivo *dadosIPR2.dat* (Figura 7.18). A Figura 7.19 mostra o software externo Gnuplot plotando as curvas de IPR a partir deste arquivo.

```
Menu de Inicialização
Selecione uma das opcoes abaixo:
- Entrada de dados pelo teclado.....1
- Entrada de dados por arquivo (.dat).....2
- Visualizar dados de arquivo (.dat).....3
- Salvar dados de pressão e vazão (.dat).....4
- Plotar Curvas de IPR gerados pelo software.....5
- Plotar Curvas de IPR de um arquivo (.dat).....6
- Salvar gráfico (.png).....7
- Sobre o programa.....8
- Sair.....9

Opção desejada: 6
O arquivo deve estar na pasta raíz do software. Informe o nome do arquivo: dadosIPR2
```

Figura 7.18: Tela do programa mostrando a opção de plotar curvas de IPR a partir do arquivo de disco *dadosIPR2.dat*

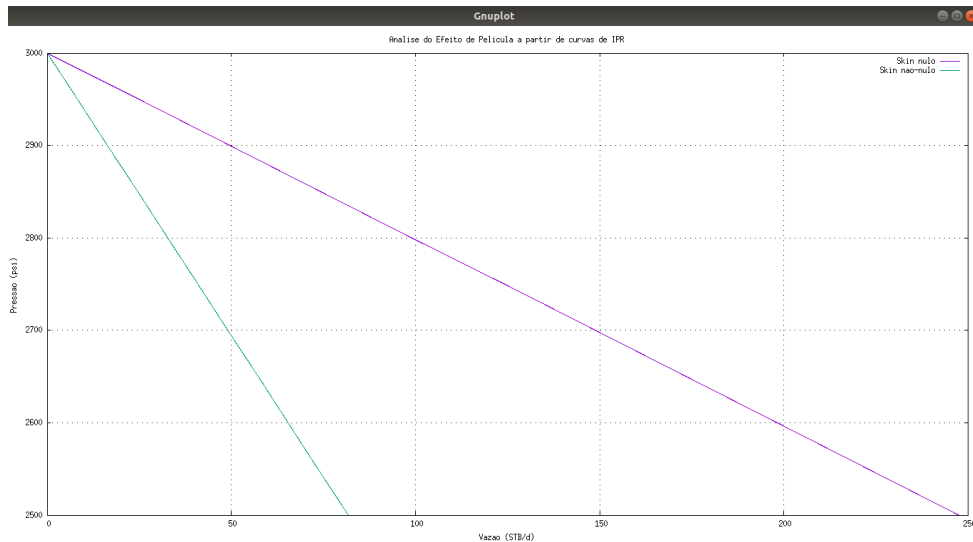


Figura 7.19: Software externo Gnuplot gerando gráfico de curvas de IPR a partir do arquivo de disco *dadosIPR2.dat*

7.7 Teste 7: Salvar gráfico

No teste 7, foi escolhido a opção de salvar o gráfico gerado pelo programa. O usuário deverá informar o nome do arquivo em que deseja salvá-lo, em formato *.png*. No exemplo abaixo, o nome *curvaIPR* é dado ao gráfico gerado pelo Gnuplot (Figura 7.20). A Figura 7.21 mostra a pasta raiz do programa com o arquivo *curvaIPR.png*.

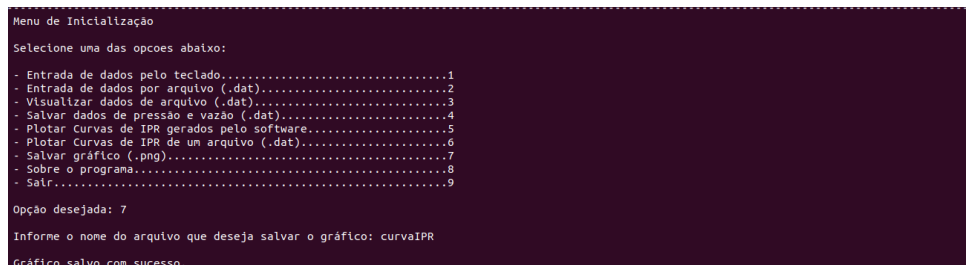


Figura 7.20: Tela do programa mostrando a opção de salvar gráfico e nomeando o arquivo como *curvaIPR.png*

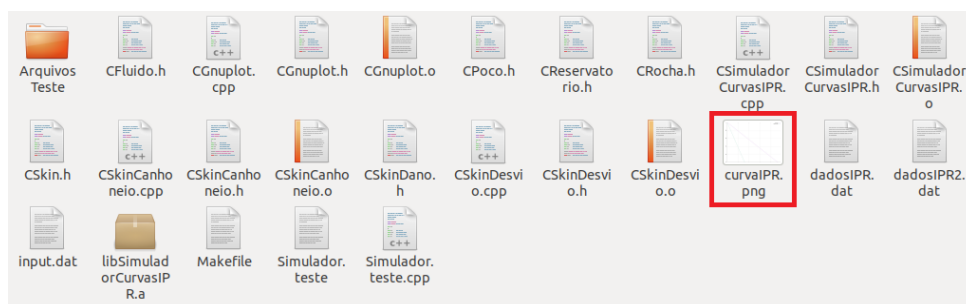


Figura 7.21: Pasta raiz do software mostrando o arquivo *curvaIPR.png* (em destaque em vermelho) salvo em disco

7.8 Teste 8: Informações sobre o programa

No teste 8, foi escolhido a opção de obter maiores informações sobre o programa. O intuito desta opção é informar a um usuário leigo no programa os recursos fornecidos pelo software, assim como esclarecer como os parâmetros são calculados (Figura 7.22).

```
Menu de Inicialização
Selecione uma das opcoes abaixo:
- Entrada de dados pelo teclado.....1
- Entrada de dados por arquivo (.dat).....2
- Visualizar dados de arquivo (.dat).....3
- Salvar dados de pressão e vazão (.dat).....4
- Plotar Curvas de IPR gerados pelo software.....5
- Plotar Curvas de IPR de um arquivo (.dat).....6
- Salvar gráfico (.png).....7
- Sobre o programa.....8
- Sair.....9

Opção desejada: 8
-----
Este programa analisa o efeito de película a partir de curvas de Índice de Produtividade (IPR) em poços verticais com fluxo monofásico de óleo.
A partir dos parâmetros de entrada realiza-se uma estimativa dos efeitos de película.
Os efeitos de película calculados são:
- Efeito de película devido alteração da permeabilidade nas proximidades do poço calculado pela Fórmula de Hawkins (1956).
- Efeito de película causado pelo canhoneio parcial do reservatório a partir de fórmulas propostas por Papatzacos (1987).
- Efeito de película causado pela inclinação do poço a partir de fórmulas propostas por Besson (1990).
- Efeito de película total (soma dos anteriores).
A partir destas estimativas o programa calcula o vazão no fundo do poço para poços verticais de óleo em regime transiente, pseudopermanente ou permanente (definido pelo usuário) considerando efeito de película nulo e não-nulo para avaliar se há um dano ou estímulo à formação.
O usuário poderá ler, salvar e visualizar os dados em '.dat', plotar gráficos e salvá-los em '.png' para melhor análise do efeito de película total.
```

Figura 7.22: Tela do programa mostrando maiores informações ao usuário sobre os recursos oferecidos

Capítulo 8

Documentação

Apresenta-se neste capítulo a documentação de uso do software Análise do Efeito de Película a partir de Curvas de Índice de Produtividade (IPR) para Poços Verticais em Escoamento Monofásico de Óleo.

8.1 Documentação do usuário

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

8.1.1 Como rodar o software

Compile o programa e, depois, execute-o. Logo após esta etapa, siga os seguintes passos:

1. No menu inicial escolha uma das opções abaixo:
 - (a) Opção 1 - Entrada de dados pelo teclado
 - (b) Opção 2 - Entrada de dados por um arquivo de disco
 - (c) Opção 3 - Visualizar um arquivo de disco
 - (d) Opção 4 - Salvar dados em arquivo de disco
 - (e) Opção 5 - Plotar curvas de IPR a partir de dados gerados pelo programa
 - (f) Opção 6 - Plotar curvas de IPR a partir de um arquivo de disco
 - (g) Opção 7 - Salvar gráfico
 - (h) Opção 8 - Informações sobre o programa
 - (i) Opção 9 - Sair do programa
2. Caso o usuário opte pela entrada de dados pelo teclado ou por um arquivo de disco (opção 1 ou 2), o programa mostrará um segundo menu onde o usuário deverá

escolher o regime do reservatório ou a opção de voltar para o menu inicial listados da seguinte forma:

- (a) Opção 1 - Regime Transiente
- (b) Opção 2 - Regime Pseudopermanente
- (c) Opção 3 - Regime Permanente
- (d) Opção 4 - Voltar

Obs.: Após a escolha do regime do reservatório, se o usuário optou pela entrada de dados pelo teclado, o programa executará o menu de preenchimento pelo teclado e as instruções para a ordem de entrada serão dadas. Caso escolha a entrada de dados por arquivo de disco, o programa irá perguntar o nome do arquivo onde os dados estão salvos. O nome do arquivo de disco deve possuir extensão *.dat* e ser digitado sem o formato de extensão (Exemplo: se o nome do arquivo for *dadosEntrada.dat*, o usuário deve digitar apenas *dadosEntrada*).

3. Caso o usuário opte por visualizar um arquivo de disco (opção 3), o programa irá perguntar o nome do arquivo onde os dados estão salvos. O nome do arquivo de disco deve possuir extensão *.dat* e ser digitado sem o formato de extensão (Exemplo: se o nome do arquivo for *dadosIPR.dat*, o usuário deve digitar apenas *dadosIPR*).
4. Caso o usuário opte por salvar um arquivo de disco (opção 4), o programa irá perguntar o nome do arquivo a ser salvo. O nome do arquivo de disco possuirá extensão *.dat* e deverá ser digitado sem o formato de extensão (Exemplo: se o nome do arquivo for *dadosIPR.dat*, o usuário deve digitar apenas *dadosIPR*).
5. Caso o usuário opte por plotar curvas de IPR a partir de dados gerados pelo software (opção 5), o programa irá automaticamente plotar o último conjunto de dados calculados: pressão, vazão com skin e vazão sem skin.
6. Caso o usuário opte por plotar curvas de IPR a partir de um arquivo de disco (opção 6), o programa irá perguntar o nome do arquivo onde os dados estão salvos. O nome do arquivo de disco deve possuir extensão *.dat* e ser digitado sem o formato de extensão (Exemplo: se o nome do arquivo for *dadosIPR.dat*, o usuário deve digitar apenas *dadosIPR*).
7. Caso o usuário opte por salvar um gráfico (opção 7), o programa irá perguntar o nome do arquivo a ser salvo. O nome do arquivo possuirá extensão *.png* e deverá ser digitado sem o formato de extensão (Exemplo: se o nome do arquivo for *grafico.png*, o usuário deve digitar apenas *grafico*).
8. Caso o usuário opte por obter maiores informações sobre o programa (opção 8), o programa irá descrever os principais recursos fornecidos, assim como informar de maneira breve como os parâmetros são calculados.

9. Caso o usuário opte por sair do programa (opção 9), o software será encerrado.
10. Caso o usuário digite uma opção que não corresponda as sugeridas em quaisquer que sejam os menus, uma mensagem de erro aparecerá e o usuário terá que fazer uma nova escolha.

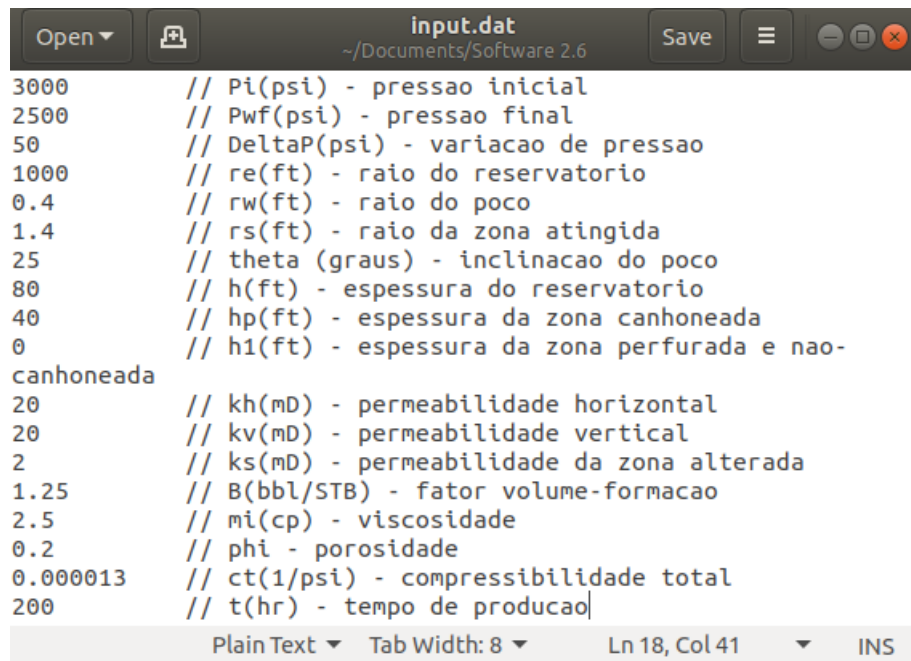
8.2 Documentação para desenvolvedor

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 GNU/Linux:
 - Instalar o compilador g++ da GNU disponível em <http://www.gnuplot.info/>.
 - Para instalar no GNU/Linux use o comando `yum install gcc`.
- No sistema operacional Windows
 - Instalar um compilador apropriado.
 - Por ter sido testado, recomenda-se o Dev C++ disponível em <http://dev-c.softonic.com.br/>.
- No sistema operacional Mac
 - Instalar um compilador apropriado.
 - Recomenda-se o Xcode disponível em <https://developer.apple.com/xcode/>.
- O software Gnuplot deve ser instalado.
 - Biblioteca CGnuplot: Os arquivos para acesso a biblioteca CGnuplot encontram-se no diretório com os códigos do software;
 - O software **gnuplot**, disponível no endereço <http://www.gnuplot.info/>, deve estar instalado.
- O programa depende da existência de arquivo de dados no formato *.dat* para preencher os parâmetros de entrada, caso o usuário opte pela entrada de dados por um arquivo de disco. A Figura 8.1 exemplifica a ordem em que os parâmetros devem estar para uma leitura correta dos dados.



```
3000 // Pi(psi) - pressao inicial
2500 // Pwf(psi) - pressao final
50 // DeltaP(psi) - variacao de pressao
1000 // re(ft) - raio do reservatorio
0.4 // rw(ft) - raio do poço
1.4 // rs(ft) - raio da zona atingida
25 // theta (graus) - inclinacao do poço
80 // h(ft) - espessura do reservatorio
40 // hp(ft) - espessura da zona canhoneada
0 // h1(ft) - espessura da zona perfurada e nao-
canhoneada
20 // kh(mD) - permeabilidade horizontal
20 // kv(mD) - permeabilidade vertical
2 // ks(mD) - permeabilidade da zona alterada
1.25 // B(bbl/STB) - fator volume-formacao
2.5 // mi(cp) - viscosidade
0.2 // phi - porosidade
0.000013 // ct(1/psi) - compressibilidade total
200 // t(hr) - tempo de producao
```

Figura 8.1: Arquivo *input.dat* usado para a entrada de dados pelo arquivo de disco mostrando a ordem dos parâmetros

8.2.2 Como gerar a documentação usando doxygen

A documentação do código do software foi feita usando o padrão JAVADOC. Depois de documentação do código, software *doxygen* foi usado para gerar a documentação do desenvolvedor no formato html. O software *doxygen* lê os arquivos com os códigos (*.h e *.cpp) e gera uma documentação muito útil e de fácil navegação no formato html.

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

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

Documentação Software Análise do Efeito de Película 1.0













Página Principal	Classes ▾	Arquivos ▾	Busca
Lista de Classes			
Aqui estão as classes, estruturas, uniões e interfaces e suas respectivas descrições:			
 CFluido	Classe correspondente aos parametros de fluidos		
 CPoco	Classe correspondente aos parametros de poco		
 CReservatorio	Classe correspondente aos parametros do reservatorio		
 CRocha	Classe correspondente aos parametros de rocha		
 CSimuladorCurvasIPR	Classe que simula a analise do efeito de pelcula a partir de curvas de IPR		
 CSkin	Classe correspondente aos parametros de dano a formacao (skin)		
 CSkinCanhoneio	Classe correspondente aos parametros de skin por canhoneio parcial		
 CSkinDano	Classe correspondente aos parametros de skin por dano ou estimulo a formacao		
 CSkinDesvio	Classe correspondente aos parametros do skin devido inclinacao do poco		
 Gnuplot	Classe de interface para acesso ao programa gnuplot		
 GnuplotException	Erros em tempo de execucao		
Gerado por  1.8.14			

Figura 8.2: Documentação do código usando o software *doxygen* mostrando a lista de classes do programa

Documentação Software Análise do Efeito de Película 1.0


















Página Principal	Classes ▾	Arquivos ▾	Busca
Lista de Arquivos			
Esta é a lista de todos os arquivos e suas respectivas descrições:			
[nível de detalhes 1 2 3]			
<div> <div>ModeloDocumento-ProjetoEngenharia-ProgramacaoPratica-master</div> <div>listagens</div> <div>  CFluido.h  CGnuplot.cpp  CGnuplot.h  CPoco.h  CReservatorio.h  CRocha.h  CSimuladorCurvasIPR.cpp  CSimuladorCurvasIPR.h  CSkin.h  CSkinCanhoneio.cpp  CSkinCanhoneio.h  CSkinDano.h  CSkinDesvio.cpp  CSkinDesvio.h  Programa.cpp  Simulador.teste.cpp </div> </div>			
Gerado por  1.8.14			

Figura 8.3: Documentação do código usando o software *doxygen* mostrando a lista de arquivos do programa

Documentação Software Análise do Efeito de Película 1.0

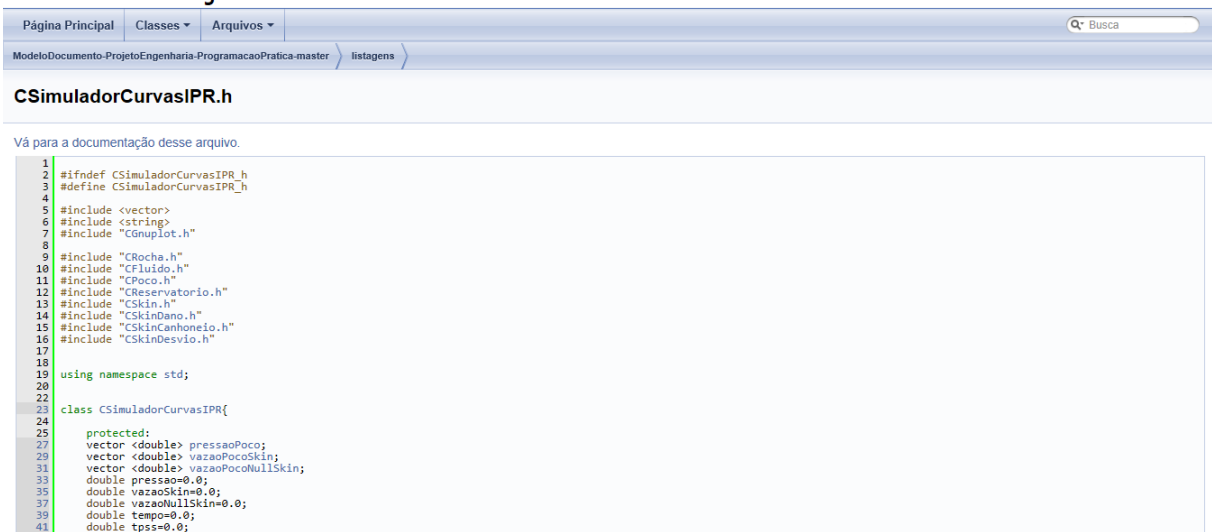


Figura 8.4: Documentação do código usando o software *doxygen* mostrando a declaração da classe `CSimuladorCurvasIPR.h`

Capítulo 9

Sugestões para Trabalhos Futuros

Para um melhoramento do projeto, sugere-se que sejam incorporadas as seguintes expansões:

- Na formulação teórica:
 - Adicionar novos cálculos de *skins*, como o causado pela penetração do canho-neado e pela turbulência do fluxo.
 - Implementar outras soluções analíticas como:
 - * Solução para reservatórios de óleo acima da pressão de bolha em poços horizontais.
 - * Solução para reservatórios de óleo abaixo da pressão de bolha, tanto para poços verticais como horizontais.
 - * Solução para reservatórios de gás, tanto para poços verticais como horizontais.
 - * Solução para reservatórios com escoamento multifásico de óleo, gás e água, tanto para poços verticais como horizontais.
 - * Soluções numéricas (FTCS, BTCS, entre outras) para as proposições acima.
- Na formulação do código:
 - Implementar sobrecarga de métodos com utilização de ponteiros.
 - Implementar o conceito de *friend* para classes e/ou métodos.
 - Desenvolver uma interface que facilite a interação entre o programa e o usuário.

Referências Bibliográficas

- [Besson 1990] BESSON, J. e. a. Performance of slanted and horizontal wells on an anisotropic medium. *European Petroleum Conference*, 1990.
- [Economides et al. 2013] ECONOMIDES, M. J. et al. *Petroleum production systems*. [S.l.]: Pearson Education, 2013. 2
- [Everdingen, Hurst et al. 1949] EVERDINGEN, A. V.; HURST, W. et al. The application of the laplace transformation to flow problems in reservoirs. *Journal of Petroleum Technology*, Society of Petroleum Engineers, v. 1, n. 12, p. 305–324, 1949.
- [Jr et al. 1956] JR, M. F. H. et al. A note on the skin effect. *Journal of Petroleum Technology*, Society of Petroleum Engineers, v. 8, n. 12, p. 65–66, 1956.
- [Karakas, Tariq et al. 1991] KARAKAS, M.; TARIQ, S. M. et al. Semianalytical productivity models for perforated completions. *SPE Production Engineering*, Society of Petroleum Engineers, v. 6, n. 01, p. 73–82, 1991.
- [Papatzacos et al. 1987] PAPATZACOS, P. et al. Approximate partial-penetration pseudoskin for infinite-conductivity wells. *SPE Reservoir Engineering*, Society of Petroleum Engineers, v. 2, n. 02, p. 227–234, 1987.
- [Rosa, Carvalho e Xavier 2006] ROSA, A. J.; CARVALHO, R. de S.; XAVIER, J. A. D. *Engenharia de reservatórios de petróleo*. [S.l.]: Interciência, 2006. 9

Índice Remissivo

A

Análise de domínio, 8
Análise orientada a objeto, 16
AOO, 16
Associações, 25
atributos, 24

C

Cenário, 5
colaboração, 20
comunicação, 20
Concepção, 4

D

Diagrama de colaboração, 20
Diagrama de componentes, 26
Diagrama de execução, 27
Diagrama de máquina de estado, 20
Diagrama de sequência, 18

E

Efeitos do projeto nas associações, 25
Efeitos do projeto nas heranças, 25
Efeitos do projeto nos métodos, 24
Elaboração, 8
especificação, 4
estado, 20
Eventos, 18

H

Heranças, 25
heranças, 25

I

Implementação, 28

M

Mensagens, 18
métodos, 24
modelo, 23, 24

O

otimizações, 25

P

Plataformas, 23
POO, 23
Projeto do sistema, 22
Projeto orientado a objeto, 23
Protocolos, 22

R

Recursos, 22