

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

PROJETO DE ENGENHARIA
DESENVOLVIMENTO DO SOFTWARE
FLUXO MONOFÁSICO REATIVO EM MEIOS POROSOS
TRABALHO DA DISCIPLINA PROGRAMAÇÃO PRÁTICA

Versão 1:
IZIS ROSA PINHEIRO

MACAÉ - RJ
Novembro - 2023

Sumário

1	Introdução	1
1.1	Escopo do problema	1
1.2	Objetivos	2
2	Concepção	3
2.1	Nome do sistema/produto	3
2.2	Especificação	3
2.3	Requisitos	4
2.3.1	Requisitos funcionais	4
2.3.2	Requisitos não funcionais	5
2.4	Casos de uso	5
2.4.1	Diagrama de caso de uso geral	5
2.4.2	Diagrama de caso de uso específico	6
3	Elaboração	8
3.1	Análise de domínio	8
3.2	Formulação teórica	9
3.3	Identificação de pacotes – assuntos	12
3.4	Diagrama de pacotes – assuntos	13
4	AOO – Análise Orientada a Objeto	14
4.1	Diagramas de classes	14
4.1.1	Dicionário de classes	15
4.2	Diagrama de sequência – eventos e mensagens	16
4.2.1	Diagrama de sequência geral	16
4.3	Diagrama de comunicação – colaboração	16
4.4	Diagrama de estado	17
4.5	Diagrama de atividades	18
5	Projeto	20
5.1	Projeto do sistema	20
5.2	Projeto orientado a objeto – POO	21

5.3	Diagrama de componentes	22
5.4	Diagrama de implantação	23
5.4.1	Tabela classificação sistema	23

Lista de Figuras

2.1	Diagrama de caso de uso – Caso de uso geral	6
2.2	Diagrama de caso de uso específico: inserir dados de entrada	7
3.1	Fluxo dos fluidos em meio porosos (Aula USP)	10
3.2	Diagrama de Pacotes	13
4.1	Diagrama de classes	15
4.2	Diagrama de seqüência	16
4.3	Diagrama de comunicação	17
4.4	Diagrama de máquina de estado da classe X	18
4.5	Diagrama de atividades classe X::MetodoY	19
5.1	Diagrama de componentes	22
5.2	Diagrama de implantação	23

Lista de Tabelas

2.1 Caso de uso 1 5

Listagens

Capítulo 1

Introdução

No presente projeto de engenharia desenvolve-se o software fluxo monofásico reativo em meios porosos.

O fluxo monofásico reativo em meios porosos refere-se ao movimento de fluidos através de um meio poroso, onde reações químicas na interface sólido-fluido causam dissolução da matriz porosa circundante, criando *feedback* não linear. Este processo é frequentemente estudado no contexto de reações geoquímicas em reservatórios de petróleo. Dessa forma, a principal finalidade do programa é simular o transporte reativo em meios porosos, incluindo a dissolução e precipitação de espécies químicas, mudanças na permeabilidade e outros fatores que acoplam o transporte geoquímico ao fluxo de fluidos e transporte de energia.

1.1 Escopo do problema

O escoamento monofásico reativo em meios porosos abrange o desenvolvimento de esquemas numéricos e modelos matemáticos para simular as complexas interações entre escoamento de fluidos e transporte reativo em materiais porosos. Isso envolve a abordagem de vários desafios em métodos teóricos e numéricos, tais como o acoplamento do transporte geoquímico ao fluxo de fluidos e transporte de energia, e o desenvolvimento de esquemas de volumes finitos para equações parabólicas não lineares modelando escoamento monofásico com transporte reativo em meios porosos anisotrópicos. Portanto o problema abrange o desenvolvimento de esquemas numéricos avançados e modelos matemáticos para simular as complexas interações entre escoamento de fluidos e transporte reativo em materiais porosos, com foco na abordagem de desafios em métodos teóricos e numéricos, e na simulação de escoamento em meios porosos fraturados com dissolução mineral e precipitação.

1.2 Objetivos

Os objetivos deste projeto de engenharia são:

- Objetivo geral:
 - Desenvolver um projeto de engenharia de software baseado em simulação numérica implícita computacional para simular o escoamento monofásico reativo em meios porosos.
- Objetivos específicos:
 - Modelar física e matematicamente o problema.
 - Modelagem estática por meio de diagramas com interface amigável.
 - Calcular propriedades dos meios porosos.
 - Calcular propriedades dos fluidos.
 - Parâmetros de transporte reativo.
 - Resolver sistemas de equações.
 - Gerar gráficos externos a partir do software externo **gnuplot**.

Capítulo 2

Concepção

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

2.1 Nome do sistema/produto

Nome	FLUXO MONOFÁSICO REATIVO EM MEIOS POROSOS
Componentes principais	Modelos de fluxo, capacidades de transporte reativo, propriedades de meios porosos, propriedades de fluidos, parâmetros de transporte reativo, condições iniciais e de contorno, métodos numéricos, uma interface com o usuário
Missão	Gerar gráficos das soluções que a partir do sistema de equações

2.2 Especificação

O escoamento monofásico reativo em meios porosos refere-se ao movimento de um fluido através de um material poroso, onde o fluido sofre uma reação química com a matriz sólida do material. Esse processo é importante em várias aplicações naturais e industriais, como contaminação de águas subterrâneas, sequestro de carbono e recuperação aprimorada de petróleo. O comportamento do escoamento monofásico reativo em meios porosos pode ser descrito por meio de modelos matemáticos que levam em consideração o transporte do fluido, as reações químicas e as mudanças na estrutura porosa (Banshoya et al., 2023; Fumagalli & Scotti, 2021).

Deseja-se desenvolver um software que resolva as equações diferenciais parciais não lineares que governam este problema usando usando métodos numéricos para simular o comportamento do sistema ao longo do tempo e do espaço. O simulador será empregado

no ambiente acadêmico como um software de código aberto. Utilizando-se a programação orientada a objetos em C++ e o software gnuplot. A interface de interação com o usuário será por meio do teclado, mouse e monitor em sintonia com a interface do sistema desenvolvido. Essa abordagem visa garantir a acessibilidade generalizada e a utilização intuitiva do software.

- **Dados/Atributos:**

- O usuário terá que fornecer condições de contorno e condições iniciais de espaço e tempo;
- Saturação de água irreduzível;
- Viscosidade ;
- Porosidade;
- Permeabilidade;
- Pressão;
- Concentração;
- Parâmetros de transporte reativos.

2.3 Requisitos

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

2.3.1 Requisitos funcionais

Apresenta-se a seguir os requisitos funcionais.

RF-01	O usuário tem a liberdade de escolher todos os dados de entrada.
RF-02	Deve permitir o carregamento de arquivos criados pelo software.
RF-03	O usuário pode modelar o processo de simulação escolhendo qual será a ordem da geração dos perfis.
RF-04	Deve mostrar os resultados na tela.
RF-05	O usuário poderá plotar seus resultados de simulação em gráficos. O gráfico poderá ser salvo como imagem ou ter seus dados exportados como texto.

2.3.2 Requisitos não funcionais

RNF-01	Os cálculos devem ser feitos utilizando-se o método numérico implícito para cada passo de tempo.
RNF-02	O programa deverá ser multi-plataforma, podendo ser executado em <i>Windows</i> , <i>GNU/Linux</i> ou <i>Mac</i> .

2.4 Casos de uso

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

Tabela 2.1: Caso de uso 1	
Nome do caso de uso:	Simulação de Fluxo Monofásico Reativo
Resumo/descrição:	O engenheiro interage com o software para configurar parâmetros, executar simulações e analisar os resultados.
Etapas:	<ol style="list-style-type: none"> 1. Inserir dados de entrada. 2. Executar os calculos. 3. Gerar gráficos. 4. Analisar resultados.
Cenários alternativos:	Um cenário alternativo é a não utilização de reação química no problema.

2.4.1 Diagrama de caso de uso geral

O diagrama de caso de uso geral da Figura 2.1 mostra o usuário acessando as principais funcionalidades do software para simular escoamento monofásico reativo em meios porosos. Este diagrama de caso de uso ilustra as etapas a serem executadas pelo usuário ou sistema, ou seja, a iteração do usuário com o sistema.

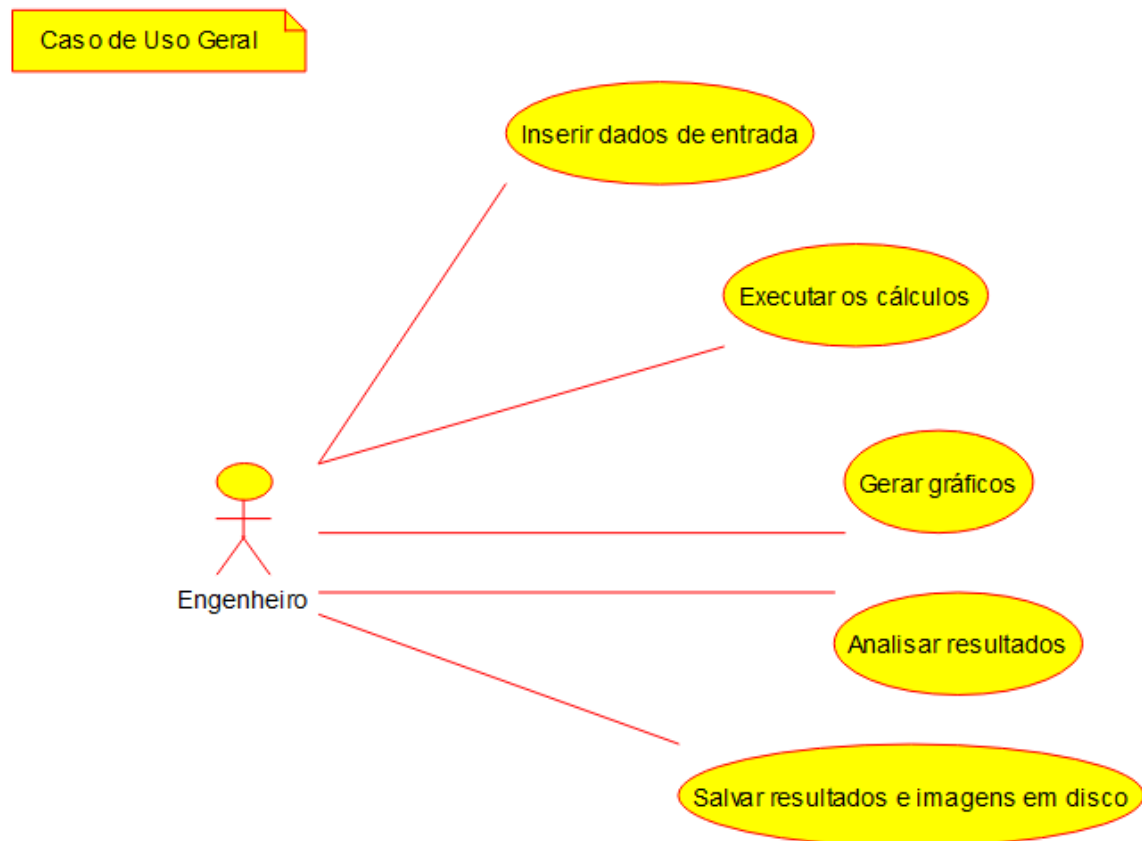


Figura 2.1: Diagrama de caso de uso – Caso de uso geral

2.4.2 Diagrama de caso de uso específico

O diagrama de caso de uso específico descrito na Figura 2.1 e na Tabela 2.1 é detalhado na Figura 2.2. Ele mostra a interação do usuário- software para simular escoamento monofásico reativo em meios porosos.

No primeiro caso de uso específico mostra-se as possibilidades de se simular o software com usuário tendo a possibilidade de escolher as condições iniciais e de contorno, e a taxa de reação química. Após a escolha desses dados, o usuário irá inserir os dados e, por fim, analisará os resultados (eventualmente gerará gráficos com os resultados obtidos utilizando um sistema externo, como o software *gnuplot*). Este diagrama de caso de uso ilustra as etapas a serem executadas pelo usuário ou sistema, a interação do usuário com o sistema.



Figura 2.2: Diagrama de caso de uso específico: inserir dados de entrada

Capítulo 3

Elaboração

Depois da definição dos objetivos, da especificação do software e da montagem dos primeiros diagramas de caso de uso, a equipe de desenvolvimento do projeto de engenharia passa por um processo de elaboração que envolve o estudo de conceitos relacionados ao sistema a ser desenvolvido, a análise de domínio e a identificação de pacotes.

Na elaboração fazemos uma análise dos requisitos, ajustando os requisitos iniciais de forma a desenvolver um sistema útil, que atenda às necessidades do usuário e, na medida do possível, permita seu reuso e futura extensão.

3.1 Análise de domínio

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

- Engenharia de Reservatórios: parte fundamental na qual esse projeto se sustenta. O software desenvolvido, utiliza conceitos tais como de propriedades dos fluídos, propriedades de rochas e a Recuperação Avançada de Petróleo. Ele então aplicará todos esses conceitos na caracterização adicional do reservatório e do poço o que permite a predição do comportamento de ambos ao longo da produção
- Simulação de Reservatórios é um ramo da Engenharia de Reservatórios. Trata da utilização e do desenvolvimento de simuladores que buscam prever o comportamento de um reservatório de petróleo e de seus poços associados por meio de modelos matemáticos.
- Modelagem Numérica Computacional que desenvolve modelos matemáticos para a solução de um determinado problema físico e então parte para um o modelo computacional por meio de algoritmos a fim de encontrar a solução do problema.
- A Termodinâmica é uma área da física que estuda os efeitos de mudanças na temperatura, pressão, volume e outras propriedades termodinâmicas de um sistema. Ela

é extremamente importante no desenvolvimento de um simulador de reservatório pois os fluidos dele sofrem diversas alterações físico-químicas durante sua produção, sendo necessária uma boa modelagem termodinâmica para entender como eles reagem a estas alterações

- Álgebra linear e Cálculo Integral e Diferencial na resolução de sistemas de matrizes e em cálculos de derivadas parciais, por exemplo.
- Pacote Gráfico: usar-se-á um pacote gráfico para plotar o comportamento da pressão, por exemplo, ao longo do poço e do reservatório para que haja uma melhor compreensão e visualização.
- Software: serão utilizados métodos e funções já existentes para a resolução de sistemas de matrizes.

3.2 Formulação teórica

A formulação teórica do escoamento monofásico reativo em meios porosos envolve a integração de diversos modelos matemáticos e leis constitutivas. Esses modelos descrevem o fluxo de fluidos reativos, reações químicas e a interação entre o fluido e o meio poroso.

O termo "transporte reativo" refere-se à integração de reações químicas com o transporte de massa, comumente observado em soluções aquosas. Em sistemas nos quais a mistura é incompleta, o transporte de reagentes e produtos pode desempenhar um papel crucial na determinação das taxas globais de reação. A modelagem do transporte reativo é frequentemente aplicada à hidrologia subsuperficial, com o objetivo de prever a dispersão de contaminantes resultantes da extração de combustíveis fósseis, processamento químico e atividades de mineração. Além disso, é utilizada para otimizar a extração de combustíveis fósseis por meio da injeção de ácido para abrir espaços porosos na rocha e para desenvolver estratégias destinadas a remoção de dióxido de carbono (Ladd; Szymczak, 2021).

Para uma modelagem adequada do caminho do fluido e a preservação da integridade das leis físicas, é necessário entender o processo de escoamento dos fluidos nas estruturas geológicas. Quando falamos de extração de óleo de reservatórios, o transporte do meio é controlado por fenômenos de advecção, dispersão e difusão se não houver nenhuma reação química.

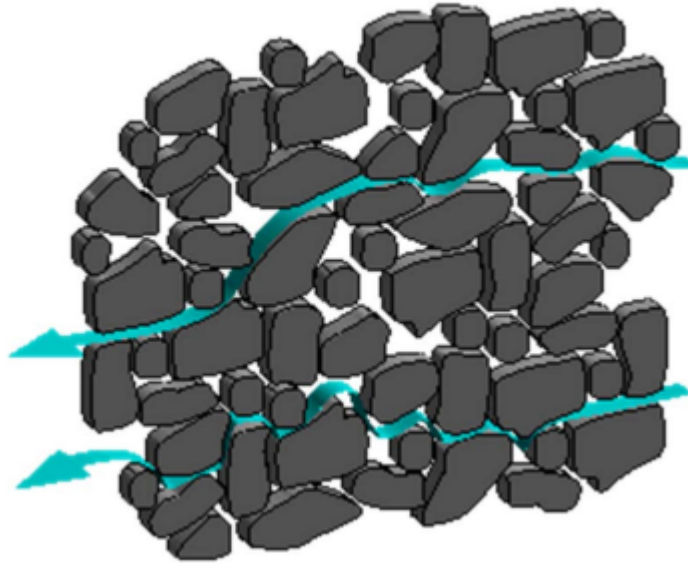


Figura 3.1: Fluxo dos fluidos em meio porosos (Aula USP)

A equação de continuidade genérica que descreve a concentração de um componente com o tempo para o escoamento de fluidos em meios porosos, considerando as forças advectivas, difusivas, dispersivas e reação química, é representada por:

$$\frac{\partial W_i}{\partial t} + \vec{\nabla} \cdot \vec{N}_i - R_i = 0 \quad i = 1, \dots, Nc \quad (3.1)$$

Agora iremos definir alguns termos que serão necessários para a construção da solução da Equação da Conservação de Massa.

O termo de acúmulo W_i , representa o somatório dos componentes i nas Np fases mais a fase estacionária, como:

$$\sum_{i=1}^{Nc} W_i = \phi \sum_{j=1}^{Np} \rho_j S_j + (1 - \phi) \rho_s \equiv \rho(\omega_i, p) \quad (3.2)$$

onde ϕ representa a porosidade do meio, ρ_j a massa específica da fase j e S_j a saturação da fase j . Pode-se considerar a densidade geral ρ como uma função da pressão local p e do conjunto das frações mássicas globais definida como:

$$\omega_i = \frac{W_i}{\sum_{i=1}^{Nc} W_i} \quad (3.3)$$

onde ω_i é a massa dos componentes i em todas as fases dividido pela massa total presente no meio poroso.

O fluxo \vec{N}_i representa o somatório dos fluxos de todas as fases do componente i , sendo composto por um termo de convecção (determinado pela velocidade de fase superficial \vec{u}_j) e um termo de dispersão (caracterizado pelo tensor de dispersão \vec{K}_{ij}).

$$\vec{N}_i = \sum_{j=1}^{N_p} \left(\rho_j \omega_{ij} \vec{u}_j - \phi \rho_j S_j \vec{k}_{ij} \vec{\nabla} \cdot \omega_{ij} \right) \quad (3.4)$$

O tensor de dispersão possui duas componentes, uma longitudinal e outra transversal para meios homogêneos e isotrópicos:

$$(K_{xx})_{ij} = \frac{D_{ij}}{\tau} + \frac{\alpha_{lj} u_{xj}^2 + \alpha_{tj} (u_{yj}^2 + u_{zj}^2)}{\phi S_j |u_j|} \quad (3.5)$$

$$(K_{xy})_{ij} = \frac{(\alpha_{lj} - \alpha_{tj}) u_{xj} u_{yj}}{u S |u_j|} \quad (3.6)$$

onde o subscrito l expressa a coordenada espacial na direção paralela, ou longitudinal ao fluxo total e t é qualquer direção perpendicular ou transversal a l. O termo D_{ij} é o coeficiente de difusão do componente i na fase j (Bird et al., 1960), α_{lj} e α_{tj} são as dispersividades longitudinal e transversal e τ é o fator de tortuosidade do meio.

• Relações Auxiliares

A Equação abaixo é uma versão da lei de Darcy para fluxos multifásicos em meio poroso (Collins, 1976) e neste formato, assume-se que o fluxo não tem escorregamento entre as fases. A função potencial da velocidade superficial da fase \vec{u}_j é representada pela soma vetorial $\vec{\nabla} p_j + \rho_j \vec{g}$ onde p_j é a pressão na fase j e \vec{g} é o vetor gravitacional que pode ser escrito como:

$$\vec{g} = g \vec{\nabla} D_z \quad (3.7)$$

onde g é vetor de magnitude gravitacional, e D_z é uma distância positiva abaixo de algum plano de referência horizontal.

A forma tensorial da permeabilidade \vec{k} implica em um meio poroso anisotrópico e permeável tendo o eixo das coordenadas não alinhados em relação aos eixos principais de \vec{k} . λ_{rj} é a mobilidade relativa da fase j, definida como o quociente entre a permeabilidade relativa k_{rj} e a viscosidade μ_j .

$$\lambda_{rj} = \frac{k_{rj}(S, \omega, \vec{x})}{\mu_j(\omega, \vec{u}_j)} \quad (3.8)$$

As permeabilidades relativas e as viscosidades k_{rj} e μ_j geralmente são determinadas experimentalmente para o cálculo de λ_{rj} . Em linhas gerais, pode-se escrever o termo $\lambda_{rj} \vec{k}$ como:

$$\lambda_{rj} \vec{k} = \frac{\vec{k}_j}{\mu_j} \quad (3.9)$$

onde \vec{k}_j é o tensor de permeabilidade da fase.

Fluxo Fracionário

Definindo o fluxo fracionário:

$$f_i = \frac{u_j}{u} \quad (3.10)$$

$$f_j = \frac{u_j}{u_t} \therefore u_t f_j = u_j \quad (3.11)$$

Isoterma de Adsorção

As isotermas de adsorção fornecem a quantidade de massa de certo contaminante adsorvido por unidade de massa de sólido (Ruthven, 1984). A isoterma de adsorção é uma curva que expressa a quantidade de substância adsorvida em um processo pela quantidade de adsorvente utilizada (q_e em $mg.g^{-1}$) em função da concentração de adsorvato (C_e em $mg.L^{-1}$) na solução no equilíbrio. A partir dessas isotermas é possível inferir sobre a capacidade de adsorção dos materiais adsorventes.

A isoterma de Henry, regida pela lei de Henry aplica-se à adsorção em uma superfície uniforme a baixas concentrações, tais que todas as moléculas são isoladas de seus vizinhos mais próximos. A relação entre a fase fluida e concentrações de equilíbrio de fase adsorvida é **linear**, com uma constante de proporcionalidade, que é igual a constante do equilíbrio de adsorção, conhecida como a constante de Henry (k_H) (Suzuki, 1990).

$$q_e = k_H C_e \quad (3.12)$$

onde:

q_e é a capacidade de adsorção no equilíbrio na fase sólida ($mg.g^{-1}$);

C_e é a concentração de equilíbrio na fase líquida ($mg.L^{-1}$);

k_H é a constante de Henry.

3.3 Identificação de pacotes – assuntos

A partir da análise dos modelos apresentados, pode-se identificar os seguintes assuntos/pacotes:

- Engenharia de Reservatórios: este pacote recebe arquivos digitados pelo usuário ou os lê de um arquivo de extensão .txt. Nele, os dados se separam, de acordo com suas característica: rocha, fluido, os parâmetros de transporte reativos, as condições de contornos e as condições iniciais. Quando juntos, fornecem uma caracterização do reservatório como um todo e servem de base para os cálculos da simulação.

- Simulador: relaciona os pacotes, sendo responsável pela criação e destruição de objetos, assim como interagir com o usuário através de um interface via texto para definir todas ações a serem tomadas.
- Modelagem Numérica Computacional: contém os algoritmos matemáticos necessários para a solução do modelo do simulador.
- Termodinâmica: pacote que envolve todos os conceitos físicos (efeitos de mudanças na temperatura, volume e outras propriedades termodinâmicas de um sistema) sendo necessário no desenvolvimento de um simulador de reservatório devido ao dinamismo do comportamento dos fluidos.
- Álgebra linear e Cálculo Integral e Diferencial: pacote com deduções matemáticas, teoremas. Base de todo o processo.
- Pacote Gráfico: é um pacote que utiliza o gnuplot para plotar as soluções numéricas obtidas, isto é, as distribuições de pressão. Em outras palavras, é o software gnuplot que implementa a saída gráfica dos dados calculados.
- Biblioteca: serão utilizadas métodos e funções já existentes para a resolução de sistemas de matrizes, bibliotecas padrão de C++ tais como (STL) e bibliotecas como a iostream, iomanip, etc.

3.4 Diagrama de pacotes – assuntos

O diagrama de pacotes da Figura 3.2 mostra as relações existentes entre os pacotes deste software.

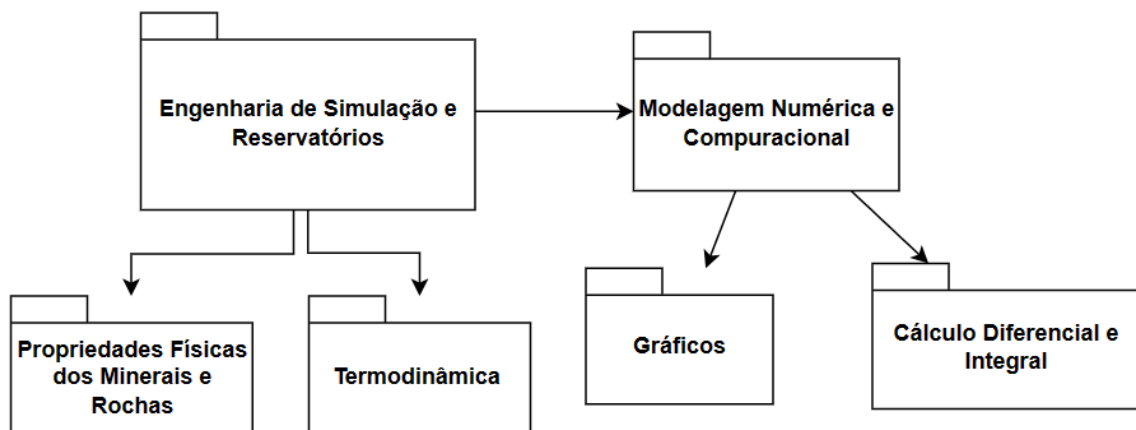


Figura 3.2: Diagrama de Pacotes

Capítulo 4

AOO – Análise Orientada a Objeto

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

Nas próximas seções, serão apresentados um conjunto de diagramas (de classes, de sequência, de comunicação, de máquina de estado e de atividades) com o objetivo de identificar os objetos e seus relacionamentos e assim visualizar o software de várias formas.

4.1 Diagramas de classes

O diagrama de classes é apresentado na Figura 4.1.

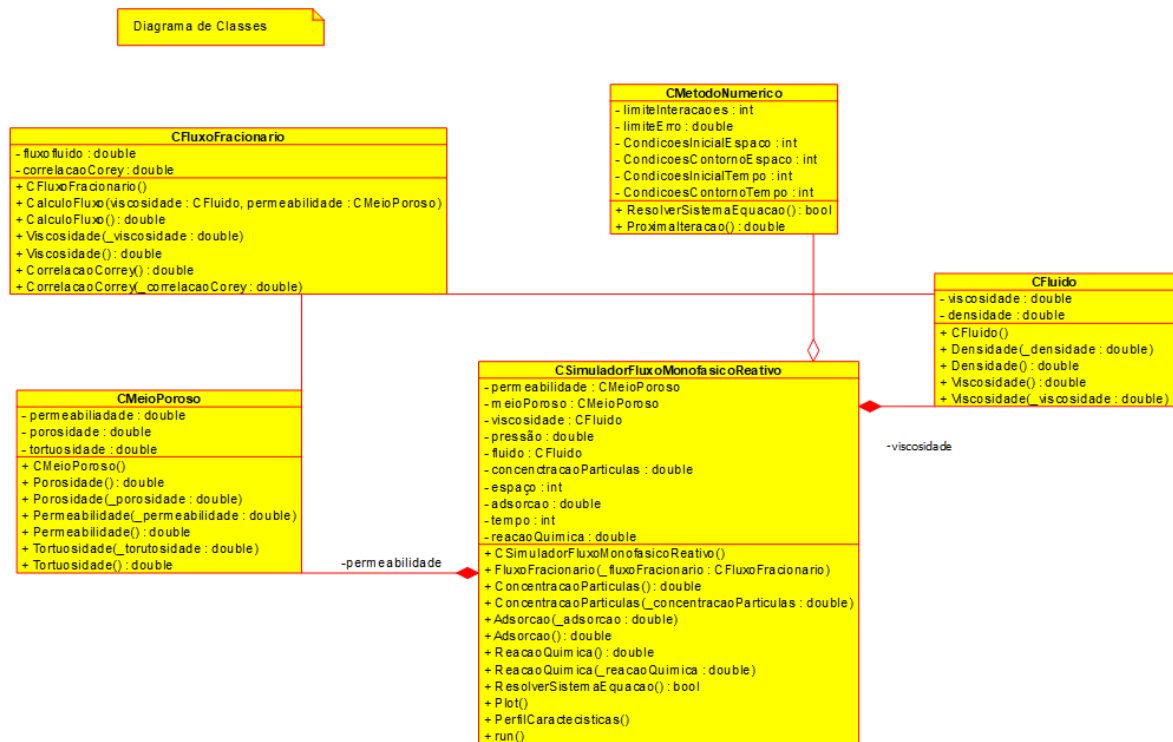


Figura 4.1: Diagrama de classes

4.1.1 Dicionário de classes

- Classe **CMeioPoroso**: representa as propriedades do meio poroso, como, por exemplo, porosidade, permeabilidade e tortuosidade.
- Classe **CFluido**: representa as propriedades do fluido, como viscosidade, densidade e composição química.
- Classe **CMetodoNumerico**: representa o método numérico para realização dos cálculos do problema, possui as condições iniciais e de contorno, número de iteração e o limite de erro.
- Classe **CFluxoFracionario**: e representa o método para cálculo do fluxo fracionário.
- Classe **CSimuladorFluxoMonofasicoReativo**: representa o processo de simulação, incluindo a configuração da simulação, a resolução do modelo de fluxo e o cálculo do perfil de concentração.
- Classe **CGnuplot**: classe que possibilita a geração de gráficos usando o software externo gnuplot.

4.2 Diagrama de sequência – eventos e mensagens

O diagrama de sequência enfatiza a troca de eventos e mensagens e sua ordem temporal. Contém informações sobre o fluxo de controle do software. Costuma ser montado a partir de um diagrama de caso de uso e estabelece o relacionamento dos atores (usuários e sistemas externos) com alguns objetos do sistema. O diagrama de sequência pode ser geral, englobando todas as operações do sistema ou específico, que enfatiza uma determinada operação.

4.2.1 Diagrama de sequência geral

O diagrama de sequência geral do software é mostrado na Figura 4.2.

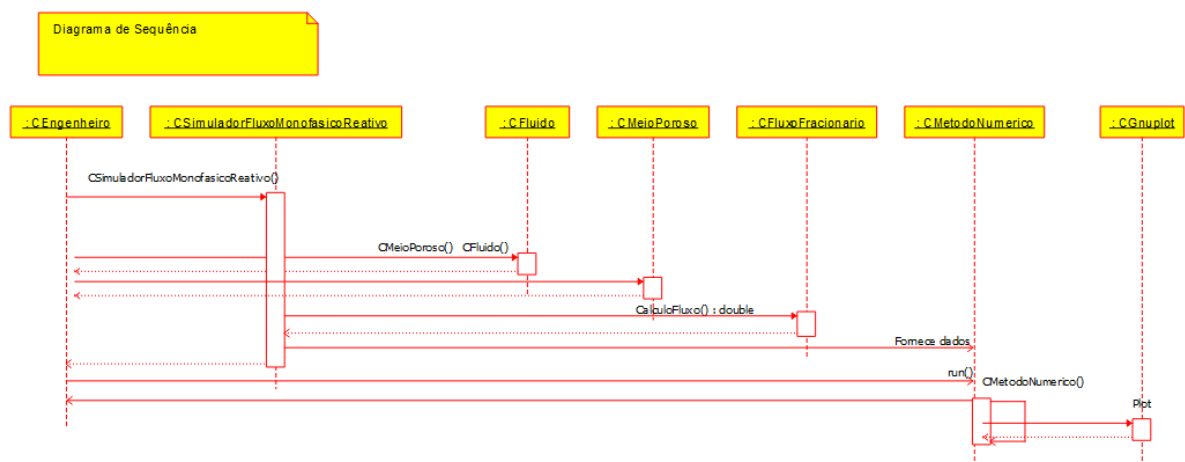


Figura 4.2: Diagrama de sequência

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. O usuário está sempre informando ao computador dados que são necessários para o processamento da simulação. Aqui a ênfase é o entendimento das mensagens que chegam e saem de cada objeto.

Veja na Figura 4.3 o diagrama de comunicação com foco no simulador.

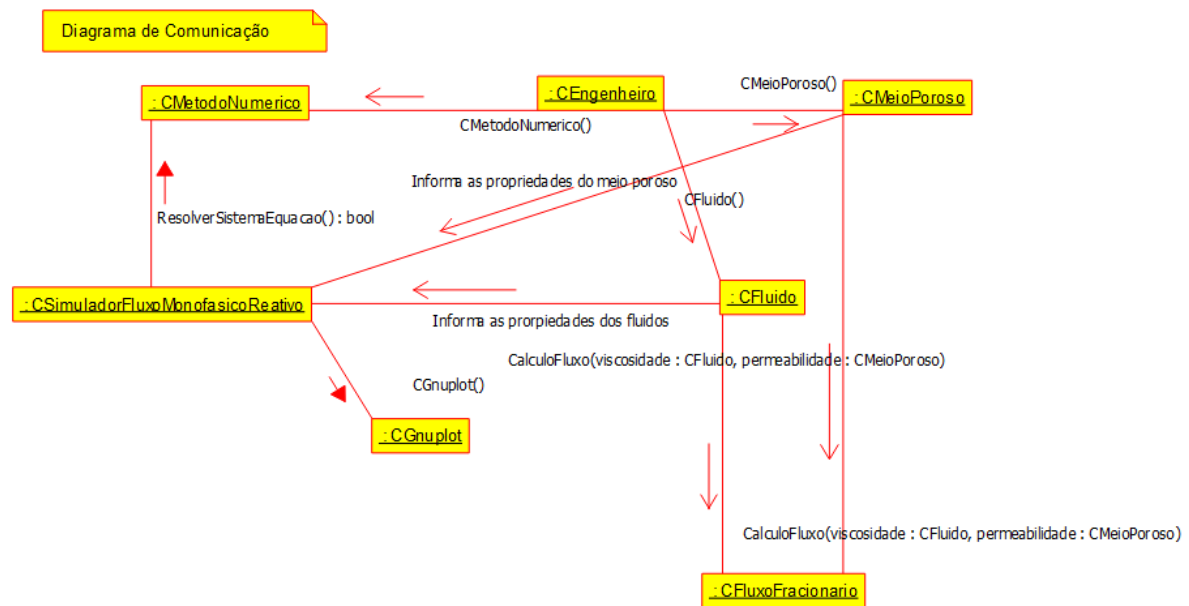


Figura 4.3: Diagrama de comunicação

4.4 Diagrama 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 4.4 o diagrama de máquina de estado para o objeto `CSimuladorFluxoMonofasicoReativo`.

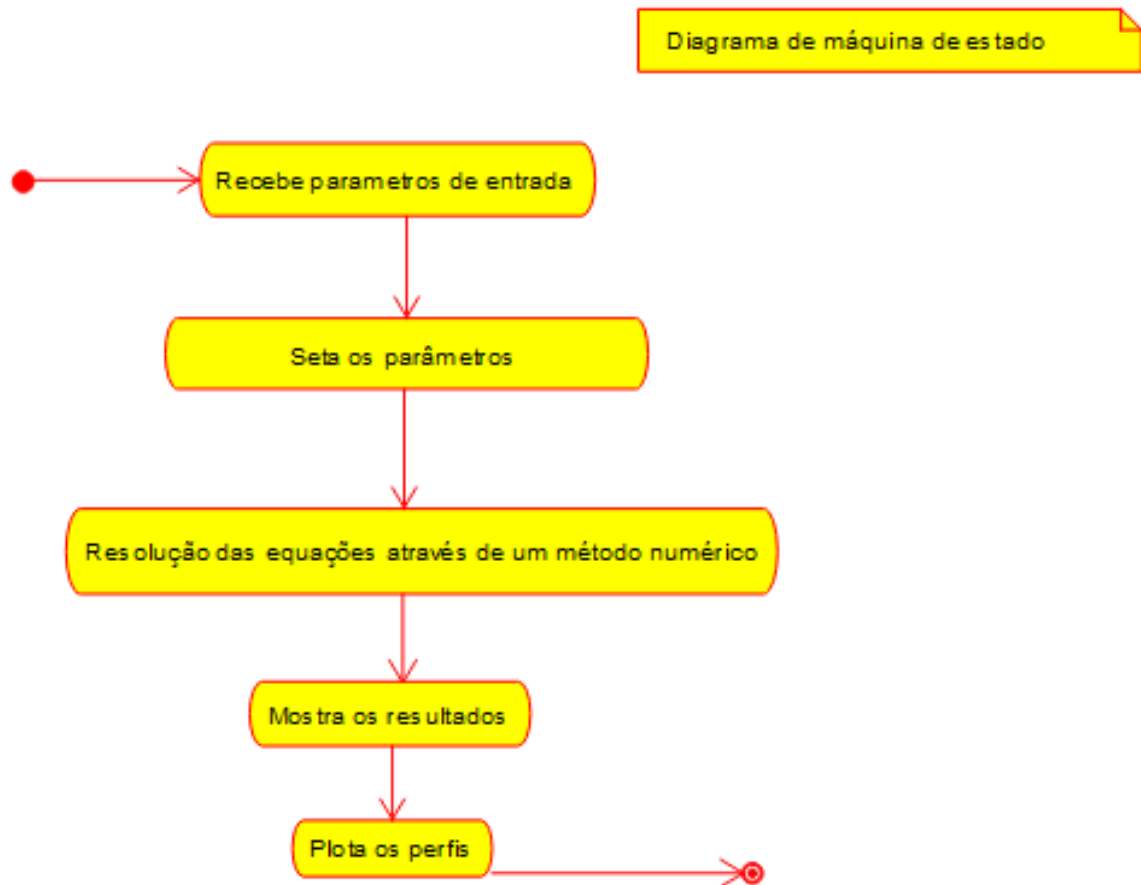


Figura 4.4: Diagrama de máquina de estado da classe X

4.5 Diagrama de atividades

Veja na Figura 4.5 o diagrama de atividades correspondente a uma atividade específica do diagrama de máquina de estado. Neste caso, para gerar um gráfico de perfil de concentração química.

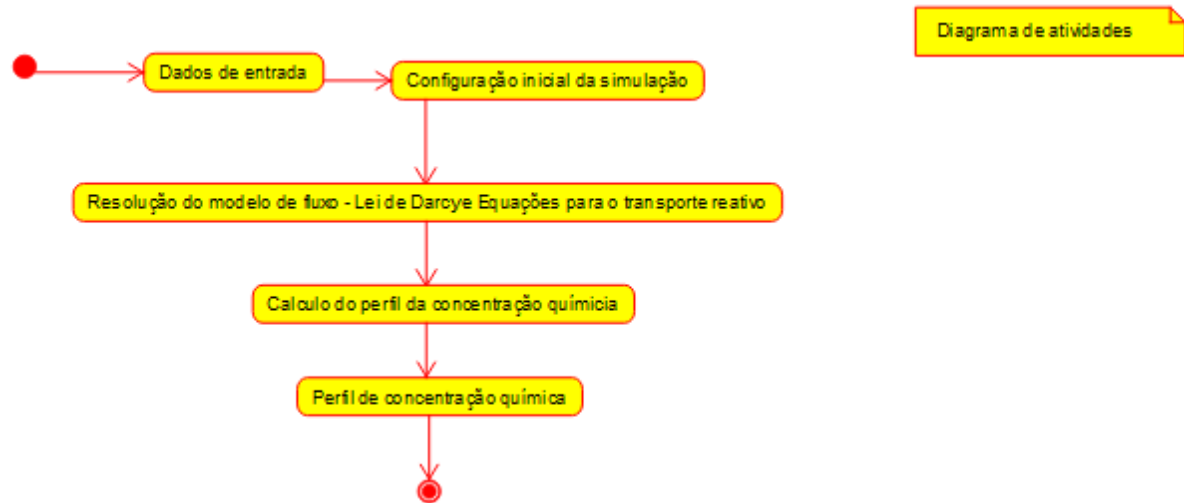


Figura 4.5: Diagrama de atividades classe X::MetodoY

Capítulo 5

Projeto

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

5.1 Projeto do sistema

Depois da análise orientada a objeto desenvolve-se o projeto do sistema, o qual envolve etapas como definição dos protocolos, interface API, uso de recursos, subdivisão do sistema em subsistemas, alocação dos subsistemas ao hardware, seleção das estruturas de controle, 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. A importância de se definir padrões específicos de documentação, nome das classes, padrões de retorno, interface do usuário e características de desempenho constitui-se como uma ferramenta estratégica para resolver o problema, elaborar uma solução e garantir repetibilidade e expansão para problemas similares. Nessa etapa serão avaliadas algumas características do software tais como:

1. Protocolos

- O programa permite salvar dados em disco no formato aberto, como .txt.
- Será efetuada a entrada de dados via arquivo de texto .txt.
- Neste projeto o software irá se comunicar com o componente externo Gnuplot que gera gráficos.

2. Recursos

- O programa utilizará uma máquina computacional com HD, CPU, RAM, periféricos, processador, teclado para a entrada de dados e o monitor para a saída de dados.

- O simulador utiliza o programa externo Gnuplot que plota figuras e gráficos.
3. Controle
 - Este software requer um controle sequencial.
 4. Plataformas
 - O programa é multiplataforma, o que permite executá-lo em Windows e Mac OS X , mas será desenvolvido na plataforma Windows. A linguagem de software utilizada é a C++ orientada a objeto.
 - Ambiente de desenvolvimento Microsoft Visual C ++ (MSVC).

5.2 Projeto orientado a objeto – POO

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

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

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

Efeitos do projeto no modelo estrutural

- O programa utiliza o HD, o processador e o teclado do computador.
- O Software pode ser executado nas plataformas GNU/Linux ou Windows.
- Existe a necessidade de instalação do software Gnuplot para o funcionamento do programa.

5.3 Diagrama de componentes

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

Veja na Figura 5.1 um exemplo de diagrama de componentes. Observe que este inclui muitas dependências, ilustrando as relações entre os arquivos.

Algumas observações úteis para o diagrama de componentes:

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

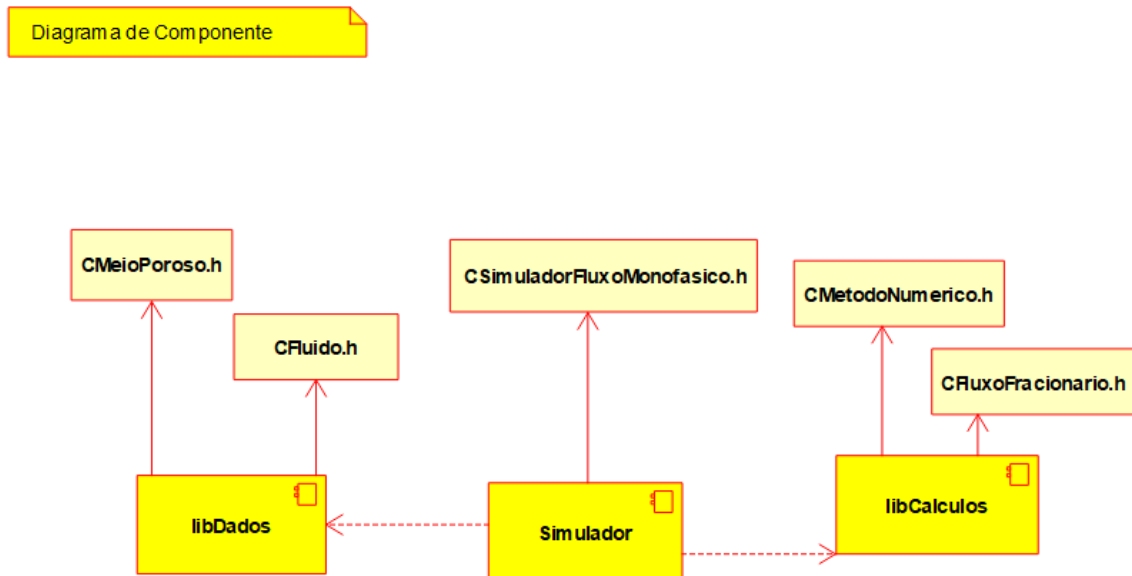


Figura 5.1: Diagrama de componentes

5.4 Diagrama de implantação

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

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

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

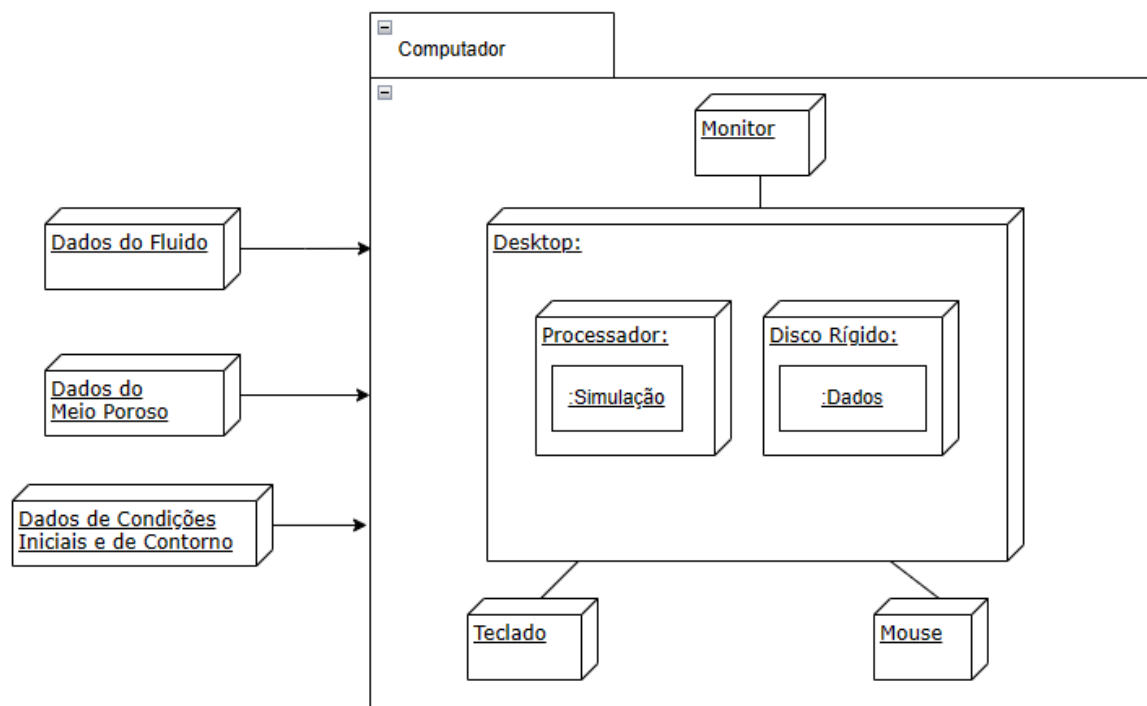


Figura 5.2: Diagrama de implantação

5.4.1 Tabela classificação sistema

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

Licença:	<input checked="" type="checkbox"/> livre GPL-v3 <input type="checkbox"/> proprietária
Engenharia de software:	<input type="checkbox"/> tradicional <input checked="" type="checkbox"/> ágil <input type="checkbox"/> outras

Paradigma de programação:	<input type="checkbox"/> estruturada <input checked="" type="checkbox"/> orientado a objeto - POO <input type="checkbox"/> funcional
Modelagem UML:	<input checked="" type="checkbox"/> básica <input checked="" type="checkbox"/> intermediária <input type="checkbox"/> avançada
Algoritmos:	<input checked="" type="checkbox"/> alto nível <input checked="" type="checkbox"/> baixo nível
	implementação: <input type="checkbox"/> recursivo ou <input checked="" type="checkbox"/> iterativo; <input checked="" type="checkbox"/> determinístico ou <input type="checkbox"/> não-determinístico; <input type="checkbox"/> exato ou <input checked="" type="checkbox"/> aproximado
	concorrências: <input checked="" type="checkbox"/> serial <input checked="" type="checkbox"/> concorrente <input checked="" type="checkbox"/> paralelo
	paradigma: <input checked="" type="checkbox"/> dividir para conquistar <input type="checkbox"/> programação linear <input type="checkbox"/> transformação/ redução <input type="checkbox"/> busca e enumeração <input type="checkbox"/> heurístico e probabilístico <input type="checkbox"/> baseados em pilhas
Software:	<input type="checkbox"/> de base <input checked="" type="checkbox"/> aplicativos <input type="checkbox"/> de cunho geral <input checked="" type="checkbox"/> específicos para determinada área <input checked="" type="checkbox"/> educativo <input checked="" type="checkbox"/> científico
	instruções: <input checked="" type="checkbox"/> alto nível <input type="checkbox"/> baixo nível
	otimização: <input checked="" type="checkbox"/> serial não otimizado <input checked="" type="checkbox"/> serial otimizado <input checked="" type="checkbox"/> concorrente <input checked="" type="checkbox"/> paralelo <input type="checkbox"/> vetorial
	interface do usuário: <input type="checkbox"/> kernel numérico <input type="checkbox"/> linha de comando <input type="checkbox"/> modo texto <input checked="" type="checkbox"/> híbrida (texto e saídas gráficas) <input type="checkbox"/> modo gráfico (ex: Qt) <input type="checkbox"/> navegador
Recursos de C++:	<input checked="" type="checkbox"/> C++ básico (FCC): variáveis padrões da linguagem, estruturas de controle e repetição, estruturas de dados, struct, classes(objetos, atributos, métodos), funções; entrada e saída de dados (streams), funções de cmath
	<input checked="" type="checkbox"/> C++ intermediário: funções lambda. Ponteiros, referências, herança, herança múltipla, polimorfismo, sobrecarga de funções e de operadores, tipos genéricos (templates), smart pointers. Diretrizes de pré-processador, classes de armazenamento e modificadores de acesso. Estruturas de dados: enum, uniões. Bibliotecas: entrada e saída acesso com arquivos de disco, redirecionamento. Bibliotecas: filesystem
	<input checked="" type="checkbox"/> C++ intermediário 2: A biblioteca de gabaritos de C++ (a STL), containers, iteradores, objetos funções e funções genéricas. Noções de processamento paralelo (múltiplas threads, uso de thread, join e mutex). Bibliotecas: random, threads
	<input type="checkbox"/> C++ avançado: Conversão de tipos do usuário, especializações de templates, excessões. Cluster de computadores, processamento paralelo e concorrente, múltiplos processos (pipes, memória compartilhada, sinais). Bibliotecas: expressões regulares, múltiplos processos
Bibliotecas de C++:	<input checked="" type="checkbox"/> Entrada e saída de dados (streams) <input checked="" type="checkbox"/> cmath <input checked="" type="checkbox"/> filesystem <input type="checkbox"/> random <input checked="" type="checkbox"/> threads <input type="checkbox"/> expressões regulares <input type="checkbox"/> múltiplos processos

Bibliotecas externas:	<input checked="" type="checkbox"/> CGnuplot <input checked="" type="checkbox"/> QCustomPlot <input type="checkbox"/> Qt diálogos <input type="checkbox"/> QT Janelas/menus/BT_____
Ferramentas auxiliares:	Montador: <input checked="" type="checkbox"/> make <input type="checkbox"/> cmake <input checked="" type="checkbox"/> qmake
IDE:	<input checked="" type="checkbox"/> Editor simples: kate/gedit/emacs <input type="checkbox"/> kdevelop <input type="checkbox"/> QT-Creator <input type="checkbox"/> _____
SCV:	<input type="checkbox"/> cvs <input type="checkbox"/> svn <input checked="" type="checkbox"/> git
Disciplinas correlacionadas	<input type="checkbox"/> estatística <input type="checkbox"/> cálculo numérico <input checked="" type="checkbox"/> modelamento numérico <input checked="" type="checkbox"/> análise e processamento de imagens