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

PROJETO DE ENGENHARIA
DESENVOLVIMENTO DO SOFTWARE
Analise Incrustação Amostra de Salmoura

DISCIPLINA : Introdução ao Projeto de Engenharia

Setor de Modelagem Matemática Computacional

Versão 1:
AUTORES Allida Faial e João Vitor Pardo
Versão 2:
AUTORES

Prof. André Duarte Bueno

MACAÉ - RJ

Junho- 2025

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Concepção

O projeto foi desenvolvido com base nos princípios de Engenharia de Software e Modelagem Matemática Computacional. A metodologia adotada incluiu as etapas clássicas de concepção, elaboração, modelagem orientada a objetos e implementação modular. Utilizou-se C++ como linguagem de programação, aplicando-se princípios de POO (Programação Orientada a Objetos) para garantir expansibilidade, organização e manutenção do sistema. Diagramas UML foram empregados para representar os aspectos estruturais e dinâmicos do

software.

O projeto foi pensado na área de incrustação pois hoje a incrustação nos tubos de perfuração vem sendo um grande problema pois os sais em sistemas de produção de petróleo é um problema recorrente na indústria de óleo e gás. Essas precipitações podem ocorrer quando duas salmouras com diferentes composições iônicas entram em contato sob certas condições de pressão, temperatura e pH, formando sais insolúveis como sulfato de bário, carbonato de cálcio e outros. Esses sais podem obstruir tubulações, danificar equipamentos e comprometer a produção.

Motivados por esse desafio, desenvolvemos um simulador de precipitação de sais que permite prever a formação de sólidos a partir da mistura de salmouras, com base em dados químicos e condições termodinâmicas. O projeto é especialmente relevante em contextos de laboratório, onde o preparo manual das soluções pode ser substituído ou complementado por simulações digitais.

1.1 Metodologia

O desenvolvimento do sistema seguiu a metodologia de projeto orientado a objeto, conforme apresentado na Figura 1.1 do documento base. Iniciou-se com a concepção e levantamento de requisitos, seguido por análise de domínio, elaboração de diagramas UML (caso de uso, classes, sequência, comunicação, atividades e estado), e culminando com a implementação em C++.

1.2 Nome do Sistema/Produto

Nome	Sistema de Análise de Incrustação em Amostras de Salmoura
Componentes principais	Interface de entrada para criação de íons, sais, salmouras e condições termodinâmicas; módulo de análise da precipitação; e relatório de resultados.
Missão	Prever a possibilidade de formação de incrustação mineral em ambientes de produção de petróleo a partir da análise de dados de salmouras

1.3 Especificação

O principal objetivo deste projeto é construir um software capaz de:

- Cadastrar e gerenciar íons com suas propriedades (nome, carga),
- Definir sais como combinações específicas de íons, com seus respectivos produtos de solubilidade (Ksp),
- Criar salmouras contendo diferentes íons e sais dissolvidos em volumes definidos,
- Simular a mistura de salmouras e calcular as concentrações finais de íons na solução resultante,
- Verificar se ocorrerá precipitação, utilizando cálculos de produto iônico comparados ao Ksp dos sais,
- Exibir os resultados da simulação,plote de gráficos comparando diferentes cenários e indicando quais sais precipitam ou permanecem estáveis.

O sistema também permite alterar temperatura e pressão da simulação, o que afeta a solubilidade dos sais e a dinâmica da precipitação.

A justificativa para o desenvolvimento deste software se baseia na demanda por uma ferramenta prática, didática e confiável para simulações químicas em ambientes educacionais e industriais. Atualmente, a previsão de precipitação salina é feita com ferramentas pagas ou por meio de experimentos laboratoriais demoradosRequisitos

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

1.3.1 Requisitos funcionais

Apresenta-se a seguir os requisitos funcionais.

RF-01	O sistema deve permitir ao usuário cadastrar íons com nome, concentração e carga

RF-02	O sistema deve permitir ao usuário criar sais a partir de dois íons e um valor de Kps

RF-03	O sistema deve aceitar a entrada de condições termodinâmicas (temperatura e pressão)

RF-04	O sistema deve calcular o produto iônico com base nas informações inseridas

RF-05	O sistema deve informar se ocorre ou não precipitação para cada sal inserido

1.3.2 Requisitos não funcionais

RNF-01	O software deve ser multiplataforma (Windows, Linux, MacOS).

RNF-02	O sistema deve ser desenvolvido em linguagem C++ com estrutura orientada a objeto

RNF-03	O sistema deve ser implementado em linguagem C++ com interface gráfica amigável.

1.4 Casos de Uso do Software

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

Tabela 1.1: Caso de uso 1

Nome do caso de uso:	Analisar Precipitação de Sal em Salmoura
Resumo/descrição:	O usuário insere dados dos íons presentes na salmoura, as condições de temperatura e pressão, e o sistema calcula se ocorrerá ou não a precipitação de sais com base na comparação entre o produto iônico (Q) e o Kps.
Etapas:	1. Criar íons com nome, carga e concentração. 2. Criar sais com os íons e valor de Kps. 3. Inserir condições termodinâmicas (T e P). 4. Executar a análise de precipitação. 5. Obter diagnóstico (precipita / não precipita). |
Cenários alternativos:	O usuário pode inserir um sal que não atinge o Kps, e o sistema deverá corretamente indicar que não há precipitação. O sistema também deve lidar com inserções incompletas ou inconsistentes, emitindo mensagens de erro

1.4.1 Diagrama de caso de uso geral do Software

O diagrama de caso de uso geral descreve as funcionalidades acessíveis ao usuário:

- Definir íons e suas propriedades (nome e carga);
- Definir sais com íons participantes, coeficientes e Ksp;
- Criar amostras de salmouras e especificar concentrações molares;
- Informar condições termodinâmicas (temperatura e pressão);
- Executar a simulação;
- Visualizar os sais que precipitam.

Além disso, foram desenvolvidos casos de uso específicos para o preparo individual das salmouras aniônicas e catiônicas, refletindo a prática laboratorial de preparo em separado antes da mistura final.

Esses diagramas são compatíveis com os métodos implementados em CSimuladorPrecipitacao e refletem diretamente os fluxos interativos do terminal.

1.4.2 Diagrama de caso de uso especifico Aniônico

O diagrama de caso de uso específico representa as ações realizadas pelo usuário durante a preparação de uma salmoura aniônica no contexto laboratorial ou experimental. As interações descritas são:

- Separar Sais Aniônicos: O usuário seleciona os sais aniônicos adequados para a preparação da solução.
- Pesar Sais Aniônicos: Após a separação, o usuário realiza a pesagem precisa dos sais para garantir a proporção estequiométrica correta.
- Preparar Salmoura Aniônica: Com os sais pesados, o usuário prepara a solução de salmoura contendo os ânions desejados, dissolvendo-os em água deionizada ou outro meio apropriado.
- Verificar pH: Por fim, o usuário mede o pH da salmoura preparada para assegurar que a solução esteja dentro da faixa adequada às análises.

Este diagrama descreve o fluxo de atividades manuais que antecedem a simulação computacional, garantindo que os dados de entrada (concentração e tipo de íons) estejam corretamente preparados e representem com fidelidade as condições reais do experimento

Diagrama de caso de uso específico – Caso de Uso Simulação de Salmoura Aniônica

]

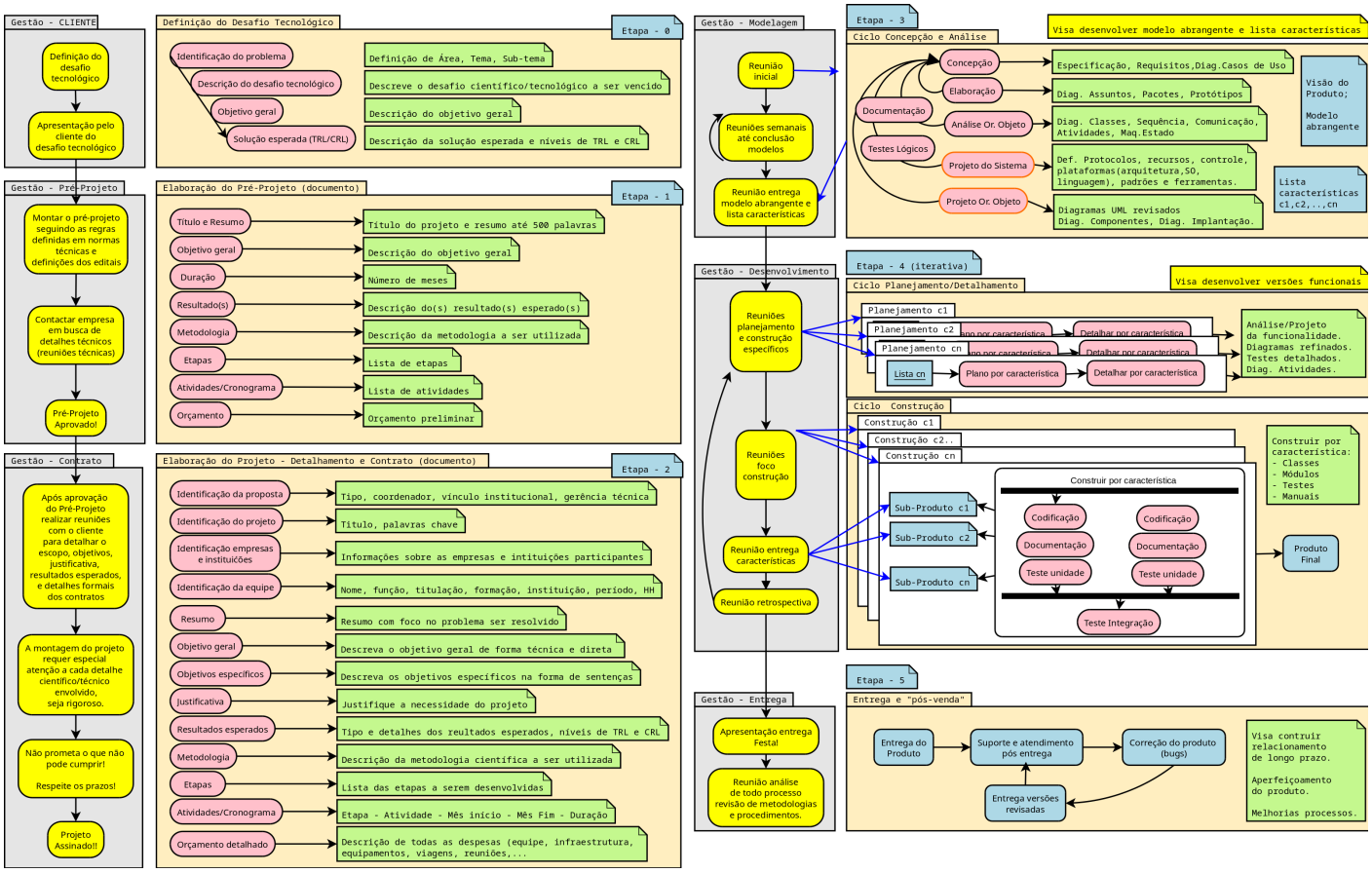

Figura 1.1: Metodologia utilizada no desenvolvimento do sistema

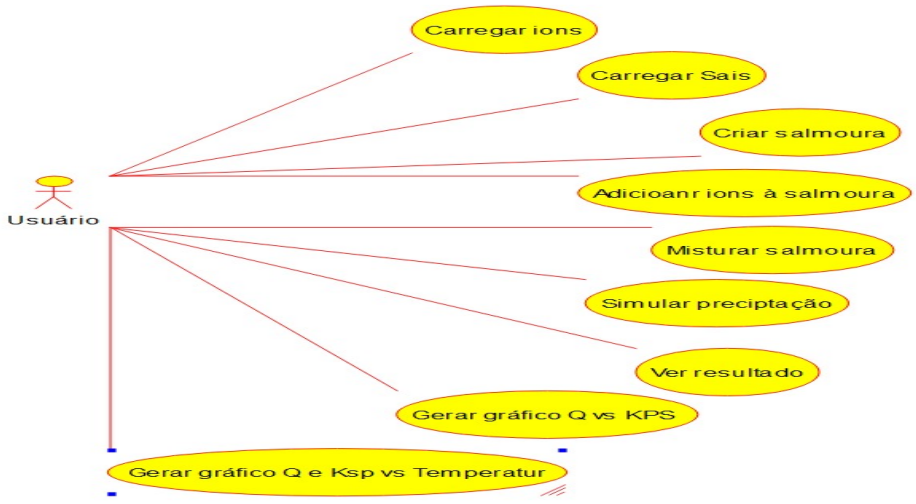

Figura 1.2: Diagrama de caso de uso – Caso de uso geral

1.4.3 Diagrama de caso de uso especifico Catiônico

O diagrama de caso de uso específico catiônico descreve as etapas realizadas pelo usuário durante a preparação da salmoura catiônica, no contexto laboratorial. Este processo faz parte da fase experimental do sistema de simulação e visa garantir que os dados inseridos no software representem adequadamente uma solução real.

As ações realizadas pelo usuário incluem:

- Separar Sais Catiônicos: O usuário identifica e seleciona os sais catiônicos que serão utilizados na formulação da salmoura.
- Pesar Sais Catiônicos: Os sais previamente separados são pesados com precisão para garantir as quantidades corretas na solução.
- Preparar Salmoura Catiônica: Os sais são dissolvidos em solvente apropriado (geralmente água deionizada), formando a salmoura com os cátions desejados.
- Verificar pH: Após a preparação, o pH da solução é medido para assegurar que ela está dentro da faixa adequada para os testes de simulação de precipitação.

Este diagrama ilustra claramente a sequência de interações que o usuário realiza com o sistema físico (antes da simulação computacional), servindo como base para a entrada de dados que serão analisados no simulador. Assim, reforça a integração entre a parte experimental e o funcionamento do software.

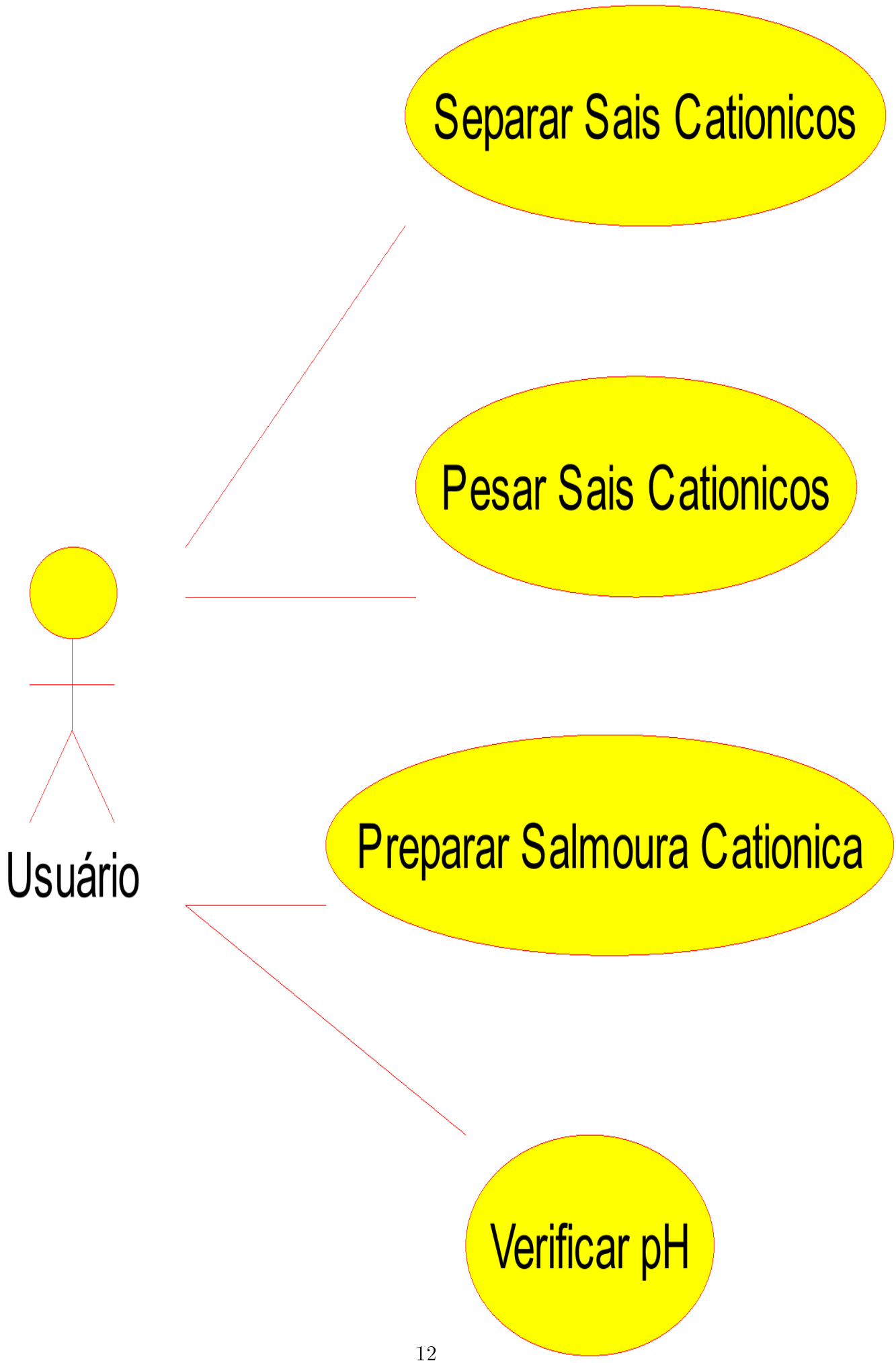

Figura 1.3: Diagrama de caso de uso específico – Caso de Uso Simulação de Salmoura Catiônica

Capítulo 2

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 processode 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.

Eliminam-se os requisitos "impossíveis" e ajusta-se a ideia do sistema de forma que este seja flexível, considerando-se aspectos como custos e prazos.

2.1 Análise de domínio

O sistema pertence à área de engenharia de petróleo e foca nos problemas relacionados à formação de incrustações minerais em tubulações e equipamentos. As incrustações são formadas a partir da precipitação de sais em águas de formação e representam um desafio significativo por reduzirem a eficiência da produção.

2.2 Formulação teórica

O software se baseia na equação do produto iônico (Q):
Q = [A]^a * [B]^b
Onde [A] e [B] são as concentrações dos íons, e "a" e "b" seus coeficientes estequiométricos. A comparação entre Q e o Kps (produto de solubilidade) define o estado do sistema:

- Se Q $<$ Kps: solução insaturada (não precipita);
- Se Q $=$ Kps: solução saturada;
- Se Q $>$ Kps: ocorre precipitação.

2.3 Identificação de pacotes – assuntos

Pacote CConcentracaoIons: Responsável por armazenar e recuperar íons cadastrados.

Pacote CSal: Modela os sais e seus dados (Kps, íons e coeficientes).

Pacote CCalcularPrecipitacao: Realiza os cálculos do produto iônico e compara com o Kps.

Pacote CIon: representa íons com nome, concentração e carga.

2.4 Diagrama de pacotes – assuntos

O diagrama de pacotes mostra como os módulos se organizam e se comunicam. Cada pacote representa

uma responsabilidade específica no sistema.

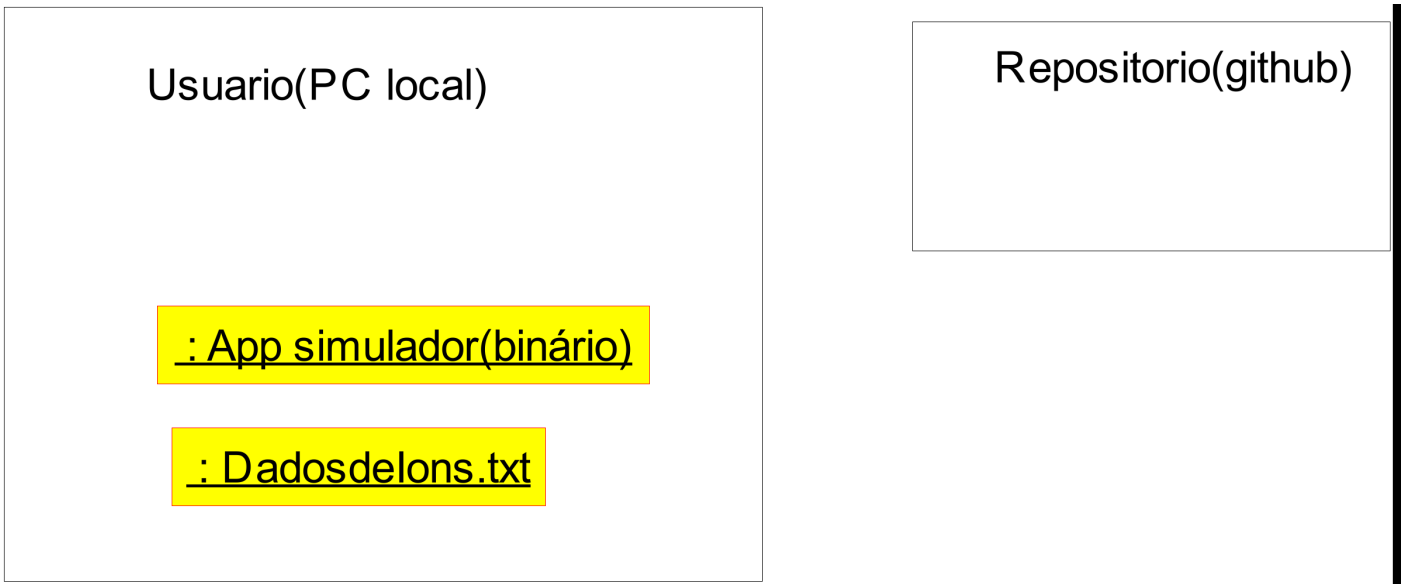

Figura 2.1: Diagrama de Pacotes

Capítulo 3

AOO – Análise Orientada a Objeto

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

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

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

3.1 Diagramas de classes

O diagrama de classes (Figura 3.1) mostra as classes do sistema, seus atributos e métodos, bem como suas relações. As principais classes incluem:

CIon: representa íons com nome, concentração e carga.

CConcentracaoIons: gerencia um mapa de íons.

CSal: modela sais com dois íons, seus coeficientes e o valor de Kps.

CCalcularPrecipitacao: módulo que executa a comparação entre Q e Kps.

3.1.1 Dicionário de classes

- CIon: Contém informações básicas dos íons, como nome e carga elétrica. Essa classe é utilizada na construção dos sais (CSal) e no controle das concentrações nas salmouras.
- CTabelaPropriedadesIons: Armazena todos os íons disponíveis no sistema em um unordered_map. Permite adicionar novos íons, salvar e carregar arquivos de dados, além de recuperar íons por nome.
- CSalmoura: Representa uma solução aquosa contendo sais e íons dissolvidos. Permite adicionar sais e íons, calcular o mapa de mols de cada espécie e controlar o volume da salmoura.
- CMisturaSalmouras: Responsável por armazenar um conjunto de salmouras e combinar seus dados. Fornece métodos para obter os sais de todas as salmouras e calcular as concentrações finais dos íons.

- CSalt: Modela um sal com base em seu nome, produto de solubilidade (Ksp), lista de íons participantes e seus coeficientes estequiométricos. Implementa os cálculos do produto iônico e da condição de precipitação.
- CSimuladorPrecipitacao: Gerencia toda a simulação, criando salmouras, misturando-as, executando os cálculos e verificando se ocorre ou não a precipitação de sais com base nas condições e concentrações finais.

3.2 Diagrama de seqüência – eventos e mensagens

Foram elaborados dois diagramas de sequência específicos:

- Criação de Íons e Sais: descreve o processo de inserção de dados pelo usuário e posterior construção de objetos CIon e CSal usando a fábrica CriarIons.
- Execução da Simulação: mostra a ordem de chamada de métodos para definição das condições termodinâmicas, adição de íons à salmoura, cálculo das concentrações e análise de precipitação utilizando CalcularPrecipitação::AnalisarPrecipitacao().

Esses diagramas refletem a dinâmica real de execução do software e ajudam a visualizar dependências entre os módulos.

3.2.1 Diagrama de sequência geral

Veja o diagrama de seqüência na Figura 3.2 3.2.

O diagrama de sequência apresentado descreve de forma detalhada a execução do método executar() da classe CSimuladorPrecipitacao, que orquestra a simulação da precipitação de sais em solução. O processo se inicia na função principal main, que chama o método executar() do simulador. Em seguida, são criadas duas instâncias da classe CSalmoura por meio das funções criarSalmouraTeste1() e criarSalmouraTeste2(), representando soluções com diferentes íons. Essas salmouras são adicionadas a um objeto CMisturaSalmouras, responsável por armazenar e tratar múltiplas soluções simultaneamente.

Após a mistura ser definida, um íon é criado utilizando a classe CIon, com nome e carga específicos, e é adicionado à salmoura com a respectiva concentração. Posteriormente, o método calcularConcentracoesFinais() da classe CMisturaSalmouras é invocado para obter as concentrações finais dos íons na mistura. Com esses dados, é instanciado um sal (CSal) utilizando seu nome, valor de constante de solubilidade (Ksp), íons envolvidos e seus coeficientes estequiométricos.

Esse diagrama mostra de forma clara e cronológica a interação entre os principais objetos do sistema durante uma simulação. Ele reforça a boa organização do código, evidenciando a separação de responsabilidades entre as classes, e auxilia na documentação e compreensão da lógica de simulação implementada.

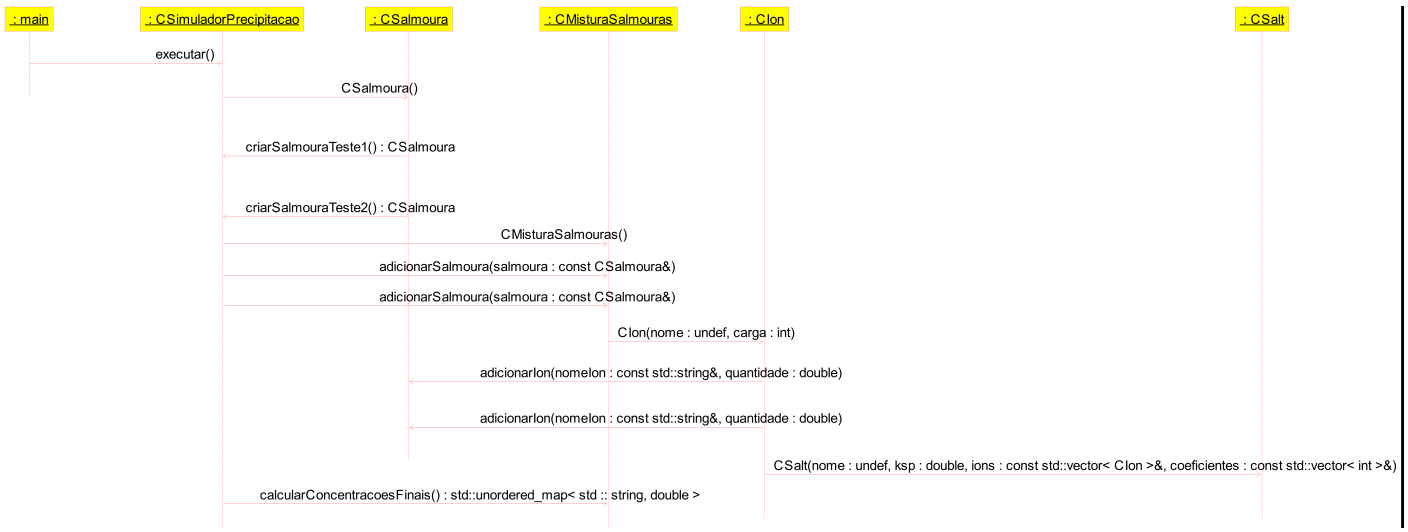

Figura 3.2: Diagrama de seqüência

3.2.2 Diagrama de sequência específico

Na Figura 3.3. observamos o diagrama de sequência específico. O diagrama ilustra a execução pontual do método executar() da classe CSimuladorPrecipitacao, destacando o fluxo lógico de verificação da possibilidade de precipitação de sais. A sequência tem início com a chamada ao método calcularConcentracoesFinais() no objeto CMisturaSalmouras, que retorna um mapa contendo as concentrações molares dos íons resultantes da mistura.

Em seguida, essas concentrações são utilizadas para calcular o produto iônico do sal, por meio do método CalcularProdutoIon(), pertencente à classe CSal. Durante esse cálculo, há uma chamada ao método getNome() da classe CIon, que fornece os nomes dos íons envolvidos na reação.

Por fim, o método VaiPrecipitar() é invocado para comparar o produto iônico obtido com o valor da constante de solubilidade (Ksp). Se o produto for superior ao Ksp, o sistema determina que haverá precipitação, encerrando a simulação com a exibição do resultado ao usuário.

3.3 Diagrama de comunicação – colaboração

O diagrama de estado (Figura 3.4). O diagrama de comunicação ilustra a troca de mensagens entre os principais objetos do sistema durante a execução da simulação de precipitação. O fluxo inicia-se a partir do objeto CSimuladorPrecipitacao, que coordena a execução chamando métodos da classe CMisturaSalmouras para adicionar salmouras (adicionarSalmoura()) e calcular as concentrações finais dos íons (calcularConcentracoesFinais()).

Essas salmouras são compostas por objetos da classe CSalmoura, que armazenam os sais e íons dissolvidos. A inserção de íons nas salmouras é feita por meio do método adicionarIon(), que busca o íon desejado na CTabelaPropriedadesIons através da função obterIon().

A classe CIon, retirados da tabela, possuem métodos como getNome() que são utilizados na construção de sais (CSal). Estes, por sua vez, realizam o cálculo do produto iônico usando calculateIonicProduct(), com base nas concentrações finais fornecidas pela mistura. Ao final, o sistema determina se

ocorre ou não precipitação.

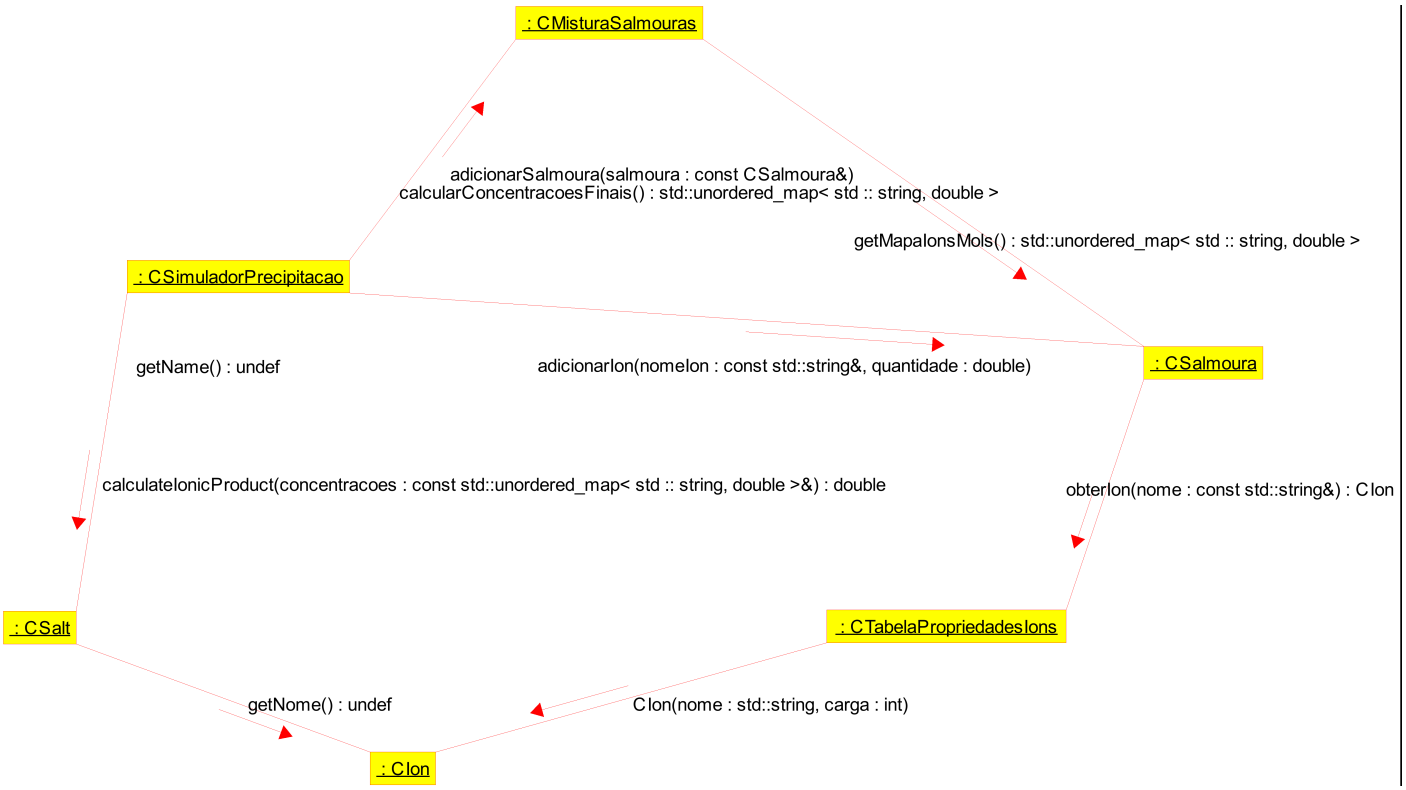

Figura 3.4: Diagrama de comunicação

3.4 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.

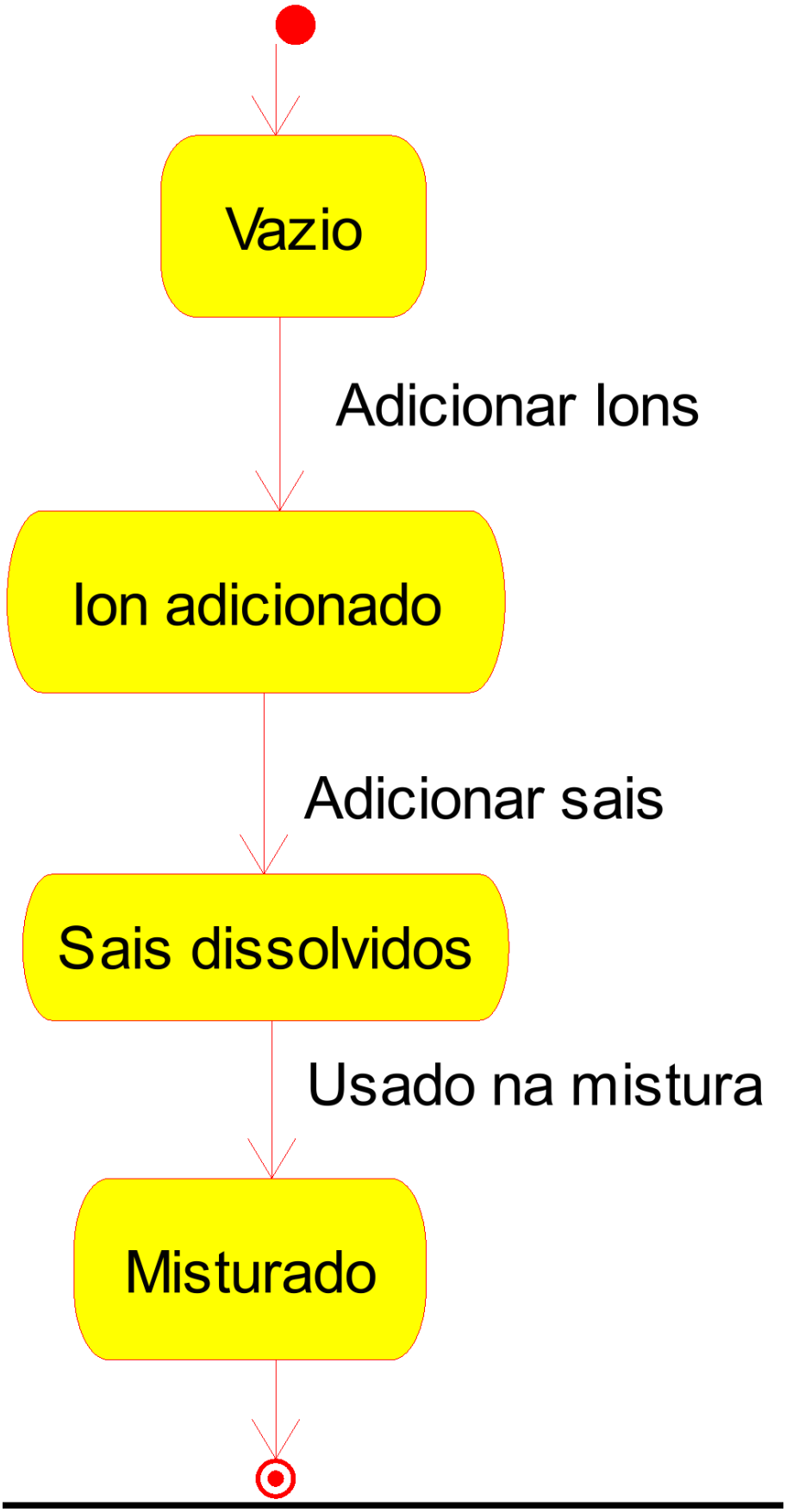

Figura 3.5: Diagrama de máquina de estado

3.5 Diagrama de atividades

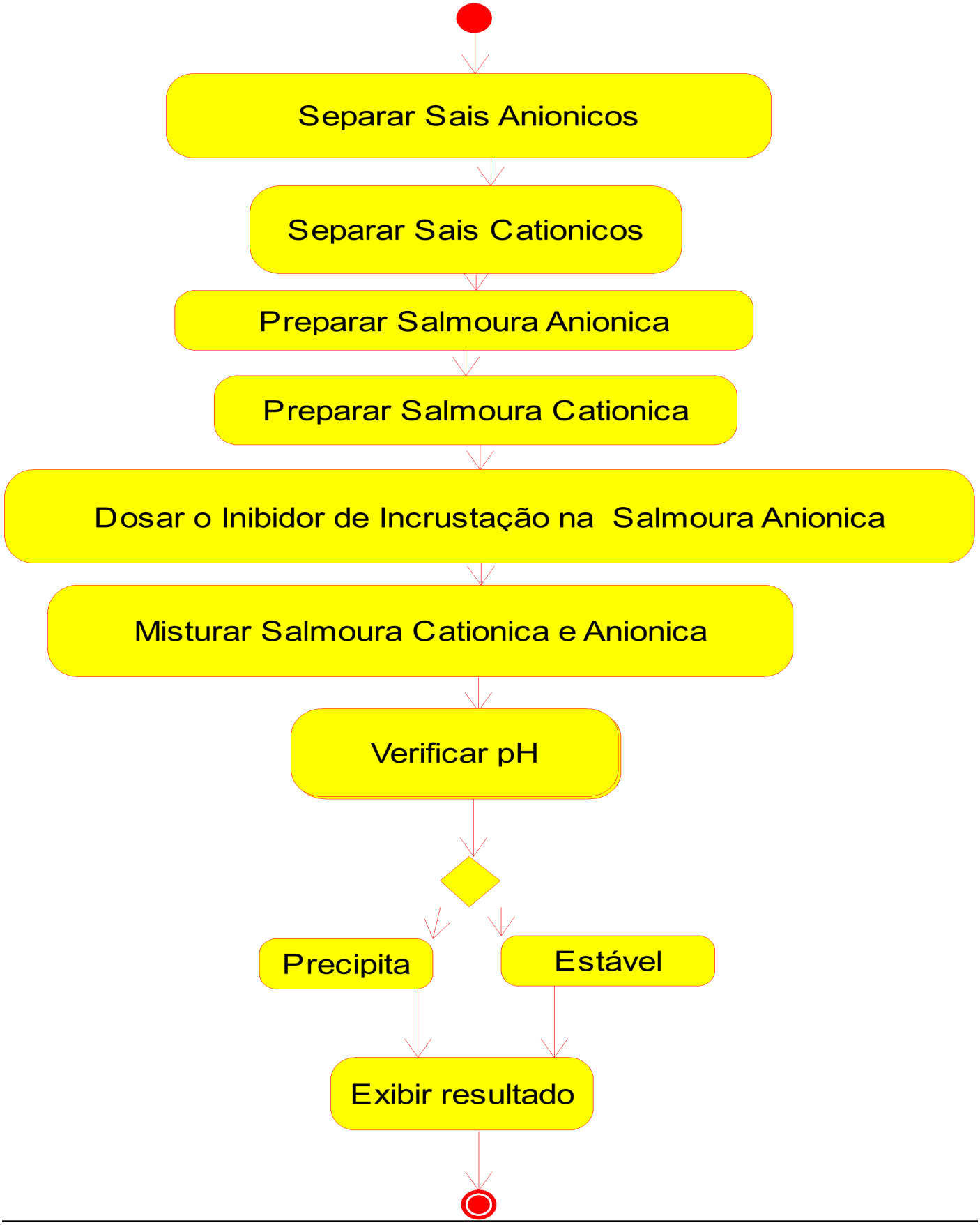

Figura 3.6: Diagrama de atividades

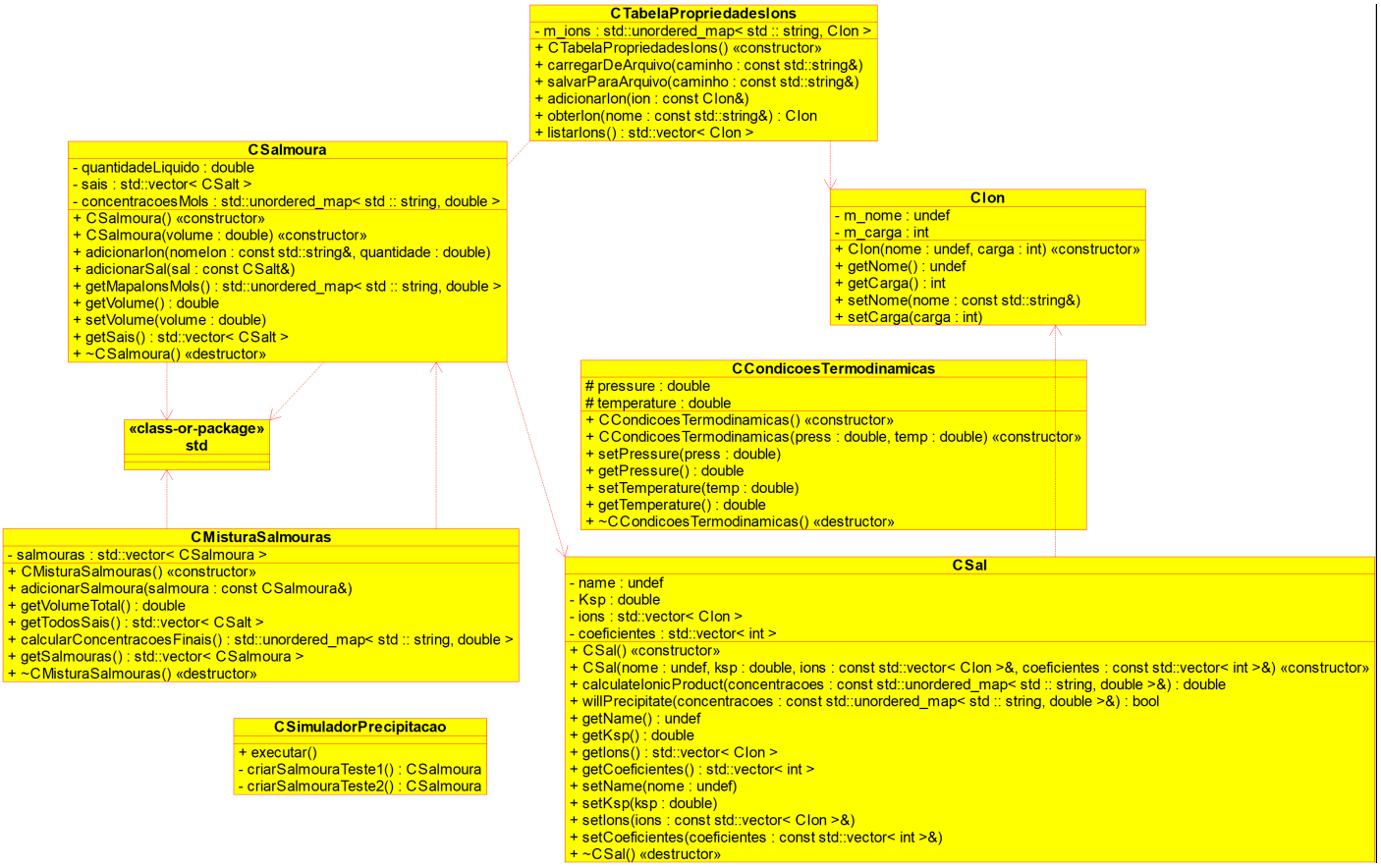

Figura 3.1: Diagrama de classes

Capítulo 4

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.

4.1 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.

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

1. Protocolos

    - Definição dos protocolos de comunicação entre os diversos elementos externos
      Neste projeto não há comunicação com dispositivos externos, mas o sistema pode ser adaptado para ler sensores de pH ou condutividade via serial no futuro.
    - A comunicação interna entre objetos ocorre por chamadas diretas a métodos públicos.
    - A interface API é implementada implicitamente nas classes que encapsulam a lógica de precipitação e manipulação de dados (ex: CalcularPrecipitacao, Sal, ion)
    - Os arquivos gerados são .txt ou .csv, formatos abertos e compatíveis com editores comuns e softwares de análise.
    - Os arquivos devem usar codificação de caracteres UTF-8.

2. Recursos

- Os recursos são gerenciados pelo próprio sistema, que armazena os dados em memória RAM. Não há alocação externa.
- Não há necessidade de banco de dados;
- O sistema não requer armazenamento de massa dedicado, apenas pequenos arquivos locais armazenados em disco.

3. Controle

- O controle do sistema é sequencial, baseado em eventos gerados pela interação do usuário com a interface (via terminal ou GUI futura).
- O sistema prevê condições inválidas (ex: sais sem coeficientes definidos) com mensagens de erro.
- Não foi feito otimização nesta versão do código
- O sistema não possui concorrência ou paralelismo nesta versão.

4. Plataformas

- O sistema segue arquitetura tradicional cliente-desktop.
- Os subsistemas identificados são: Pacote CConcentracaoIons,Pacote CSal,Pacote CCalcular-Precipitacao e Pacote CIon.
- O sistema suporta as plataformas Windows, Linux e Mac via compilação com CMake.
- As bibliotecas externas utilizadas são padrão do C++
- Utilizdos os dados da literatura e livros de qumica para temos acesso aos Kps e cargas dos Ios.

5. Padrões de projeto

- O sistema emprega encapsulamento, modularização e composição de objetos.
- Os nomes das classes seguem padrão com prefixo C (ex: CSal, CIon.CTabelaPropriedadesIons)

4.2 Projeto Orientado a Objeto – POO

O projeto orientado a objeto considera as decisões tomadas no projeto do sistema e implementa uma solução concreta. Foram feitas otimizações de métodos e atributos, modularização do código, uso de encapsulamento, clareza nos nomes e divisão lógica das responsabilidades.

Efeitos do projeto no modelo estrutural

- Os diagramas foram atualizados com inclusão de bibliotecas e classes utilitárias.
- Foram criadas classes adicionais como CalcularPrecipitacao e CriarIon para modularização.
- Dependências entre Sal, Ion e CriarIons foram estabelecidas com composição.

Efeitos do projeto no modelo dinâmico

- O diagrama de sequência foi ajustado para representar interações com o novo fluxo de entrada de dados e execução.
- Não há necessidade de novos diagramas de máquina de estado nesta versão.

Efeitos do projeto nos atributos

- Foram incluídos atributos auxiliares como coef1, coef2, Ksp, concentração, carga, e nome com validadores nas classes.

Efeitos do projeto nos métodos

- Métodos para análise de precipitação foram adicionados (CalcularProdutoIonico, VaiPrecipitar).
- Métodos foram organizados em funções públicas de controle e funções internas de cálculo.

Efeitos do projeto nas heranças

- Não há uso de herança nesta versão para manter o sistema simples e modular. Futuras versões poderão introduzir polimorfismo para tipos de sais.

Efeitos do projeto nas associações

- As associações são diretas (ex: Sal contém dois ions).
- Quando necessário, utiliza-se unordered_map como dicionário de íons.

Efeitos do projeto nas otimizações

- Loops otimizados com acesso direto às estruturas.
- Arquitetura simples evita acessos encadeados profundos.
- Atributos temporários foram criados para facilitar operações.
- Execução ajustada para resposta em tempo real.
- As associações foram revisadas para garantir performance sem comprometer legibilidade.

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

4.3 Diagrama de Componentes

O diagrama de componentes representa a estrutura lógica do sistema, mostrando como os principais módulos do software interagem entre si. No projeto Simulador de Precipitação de Sais, cada classe foi implementada com responsabilidade própria, sem a divisão tradicional em camadas (como na arquitetura MVC), refletindo uma abordagem orientada à funcionalidade. Na Figura 4.1 observa-se que:

- A entrada de dados é realizada diretamente pelas classes responsáveis, sem um módulo exclusivo de entrada. As classes CIons e CSal cuidam da criação, armazenamento e manipulação dos íons e sais, respectivamente.
- A classe CSimuladorPrecipitaçao executa os cálculos principais do programa, verificando a possibilidade de precipitação de sais com base nas concentrações iônicas e valores de Ksp. Ela representa o núcleo da análise do sistema.O sistema tambem será possivel criar diferentes graficos para que seja possivel comparar quando irá incrustar no cenário
- O arquivo principal Sandbox.cpp atua como ponto de execução, agregando as classes descritas, orquestrando a criação dos dados e invocando os métodos de análise. Não há uma camada específica de visualização neste projeto, embora seja possível futuramente integrar uma interface gráfica .
- Todos esses componentes são compilados em um único executável final:SimuladorPrecipitacao.exe.
- Nota:
  Não perca de vista a visão do todo; do projeto de engenharia como um todo. Cada capítulo, cada seção, cada parágrafo deve se encaixar. Este é um diferencial fundamental do engenheiro em relação ao técnico, a capacidade de desenvolver projetos, de ver o todo e suas diferentes partes, de modelar processos/sistemas/produtos de engenharia.

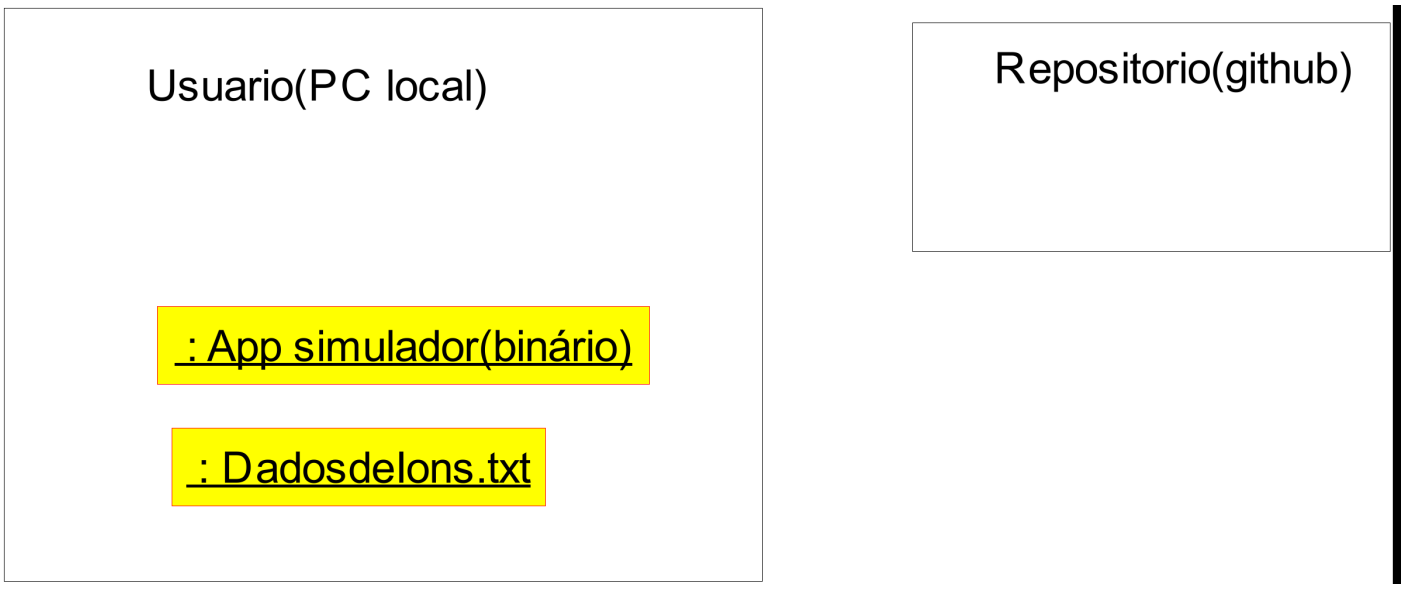

Figura 4.1: Diagrama de componentes

4.4 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.

No caso do Software de Análise de Incrustação de Amostras de Salmouras, o sistema é executado localmente em um computador pessoal (desktop ou notebook), sem necessidade de rede. Todos os módulos estão compilados em um único executável e são armazenados localmente, sendo a execução feita por linha de comando ou futura interface gráfica.

Veja na Figura 4.2 4.2 o diagrama de implantação adaptado ao projeto. Ele mostra:

- Um nó denominado Computador Corporativo, com anotações do tipo {localização: laboratório de simulação}.
- Conectado a este nó está o executável SimuladorPrecipitacao.exe, que contém internamente os módulos Analise, EntradaDados e Interface.
- Todos os arquivos .cpp e .h estão acessíveis localmente, não sendo necessário banco de dados nem rede para o funcionamento do sistema.

Essa arquitetura simples e portátil facilita a execução do sistema em ambientes acadêmicos e industriais, com poucos requisitos de hardware. O uso de arquivos .txt para entrada e saída também favorece integração com outros softwares externos.

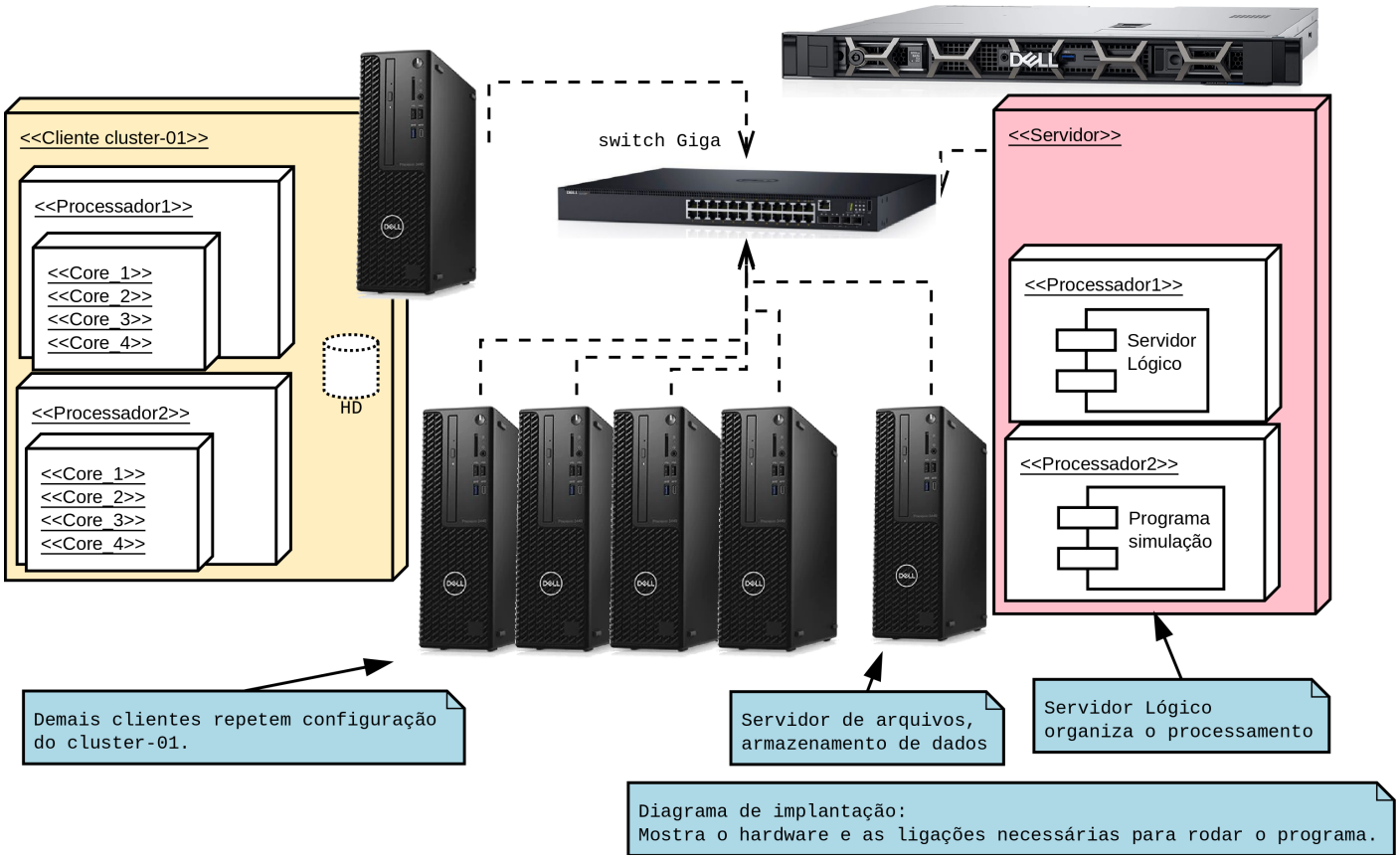

Figura 4.2: Diagrama de implantação

Capítulo 5

Lista das Características/*Features*

Neste capítulo lista-se as características do sistema a ser desenvolvido (seção 5.1).

5.1 Lista de características <<features>>

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

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

- v0.1
  - O sistema deve permitir o cadastro de íons com nome, concentração e carga elétrica.
  - O sistema deve permitir recuperar dados de íons previamente cadastrados (Ex: "Ca2+", "CO3_2-").
  - O sistema deve permitir definir um sal com nome, produto de solubilidade (Ksp), dois íons e seus coeficientes estequiométricos.
  - O sistema deve calcular o produto iônico e compará-lo com o Ksp para verificar se há precipitação.
  - O sistema deve plotar a saída no terminal, indicando o nome do sal, o valor de Q (produto iônico), o valor do Ksp e se ocorre ou não precipitação.
  - O sistema deve permitir inserir condições termodinâmicas (pressão e temperatura), mesmo que ainda não influenciem os cálculos diretamente.
- v0.3
  - O sistema deve armazenar múltiplos sais e avaliar todos com um único comando.
  - O sistema deve permitir expansão para novos tipos de sais, com diferentes coeficientes estequiométricos.
  - O sistema deve incluir mensagens mais claras para o usuário final (ex: usar nomes comerciais dos sais).

  - Testes unitários devem ser desenvolvidos para as seguintes classes:
  - CreateIons
  - Salt
  - PrecipitationCalculator
  - Lista de classes a serem testadas e aprimoradas:
  - Salt – incluir variação da constante Ksp com temperatura (futuramente).
  - PrecipitationCalculator – incluir uso de condições termodinâmicas reais.

- v0.7
  - O sistema deve apresentar uma interface simples (GUI) ou permitir leitura de dados a partir de arquivos externos (.txt, .dat ou .csv).
  - O sistema deve plotar gráficos de pressão vs. temperatura, simulando cenários reais de precipitação de sais.
  - – Testes de integração entre os módulos devem ser realizados

Referências Bibliográficas

[Blaha and Rumbaugh, 2006] Blaha, M. and Rumbaugh, J. (2006). *Modelagem e Projetos Baseados em Objetos com UML 2*. Campus, Rio de Janeiro. 23

[Rumbaugh et al., 1994] Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., and Lorensen, W. (1994). *Modelagem e Projetos Baseados em Objetos*. Edit. Campus, Rio de Janeiro. 23