



Denis Alexandre Carneiro Cunha

Licenciado em Ciências da Engenharia Química e Bioquímica

Migração de Software em Engenharia Química e suas vantagens

Dissertação para obtenção do Grau de Mestre em
Engenharia Química e Bioquímica

Orientador: Eng.º João Gouveia, SoftMorphis, Lda.

Coorientador: Professor Mário Fernando José Eusébio, FCT-UNL.

Júri:

Presidente: Prof. Dra. Isabel Maria Figueiredo Ligeiro da Fonseca

Arguente (s): Prof. Dr. José Paulo Mota

Vogal (ais): Eng.º João Carlos Antunes Abrantes Gouveia



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Março, 2016

Denis Alexandre Carneiro Cunha

Licenciado em Engenharia Química e Bioquímica

Migração de Software em Engenharia Química e suas vantagens

Dissertação para a obtenção do Grau de Mestre em Engenharia Química e Bioquímica

Orientador: Eng.º João Gouveia, SoftMorphis, Lda.

Coorientador: Professor Mário Fernando José Eusébio, FCT-UNL.

Júri:

Presidente: Prof. Dra. Isabel Maria Figueiredo Ligeiro da Fonseca

Arguente: Prof. Dr. José Paulo Mota

Vogais: Eng.º João Carlos Antunes Abrantes Gouveia

Março 2016

Migração de Software em Engenharia Química e suas vantagens

© Denis Alexandre Carneiro Cunha

Faculdade de Ciências e Tecnologia

Universidade Nova de Lisboa

A Faculdade de Ciências e Tecnologia e a Universidade Nova de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objetivos educacionais ou de investigação, não comerciais desde que seja dado crédito ao autor e editor.

Agradecimentos

Agradeço especialmente aos meus avós que sempre me apoiaram e me deram força necessária para alcançar os meus objetivos, ao meu pai, à minha mãe e em especial à minha namorada, que sempre se mostrou disponível em ajudar-me com os seus conhecimentos e ideias.

Agradeço ao Professor Mário Eusébio que sempre me acompanhou, ajudou em pontos cruciais do projeto, pelas dicas na estruturação da tese, pelo apoio incondicional e perseverança.

Um Agradecimento especial para o Fernando Ramalho por me ter permitido fazer esta tese na Morphis.

Agradeço a todos os elementos da Morphis pela disponibilidade e auxílio que me deram de forma a conseguir concluir com sucesso esta etapa, quero destacar o Nuno Guerreiro e o João Gouveia por todo o conhecimento que me transmitiram, sem a sua ajuda e paciência esta tese não poderia ter sido concluída

Agradeço a todos os meus colegas pela disponibilidade, simpatia, conhecimentos transmitidos e pela paciência e auxílio que me deram. Dos meus colegas gostaria de destacar o Júlio Machado, o Daniel Albano, o Sérgio Rebelo, o Rui Meneses e o João Costa por tudo o que me transmitiram e ajudaram.

Agradeço ao meu colega e amigo, Daniel Palos, por todo o apoio, conselhos e disponibilidade.

A todos os amigos de longa data Cláudia França, Carla França, Patrícia Correia, Marco Metelo, João Plácido, Luís Nunes, João Rafael, Bruno Ramos e a todos os amigos que fiz na faculdade, pela amizade ao longo destes anos.

Agradeço também a todos os professores pelos conhecimentos transmitidos e a Faculdade pelo percurso da minha vida em que me acompanharam.

Para a minha Namorada e Família por todo o apoio que me deram

Resumo

Numa sociedade cada vez mais tecnológica em que cada dia novas tecnologias são desenvolvidas, muitas empresas deparam-se com um problema de como atualizar os seus sistemas informáticos para englobar estas novas alterações sem perder as funcionalidades que tinham implementado ou, o elevado custo que teriam para voltar a projetar e instalar um sistema de raiz.

Como resposta a este problema surgiu a possibilidade de migração de tecnologia que tem como garantia manter as funcionalidades iniciais e alargar o sistema para estar de acordo com a nova tecnologia, seja ela Java, Python, C#, entre outras.

É importante salientar que as indústrias ligadas à engenharia química estão cada vez mais recetivas a novas tecnologias, e esforçam-se por encontrar formas de modernizar os seus softwares mediante o aperfeiçoamento ou delineando novos meios para atingir os objetivos.

O objetivo desta dissertação foi exemplificar um processo de migração de software a partir de um ficheiro de Excel de uma reação de titulação para um software elaborado em Java e abordar as vantagens do mesmo em relação ao original.

Conclui-se que uma das grandes vantagens de novas tecnologias são a acessibilidade, dado que permite acesso em qualquer dispositivo com uma ligação à internet e permitir que vários utilizadores em simultâneo consultem, alterem e insiram os dados sem ocorrerem conflitos entre os vários utilizadores.

Palavras-chave: tecnologia, Java, Python, C#, migração, software, Excel, acessibilidade, internet, utilizadores.

Abstract

In an increasingly technological society that at each day new technologies are developed many companies are faced with a problem of how to update their computer systems to support these new changes without losing the features that were implemented or the high cost that would have to return to design and install a new complete system

In response to this problem the possibility of migration technology that guarantee to maintain the initial features and extends the system to comply with the new technology demands, whether Java, Python, C#, among others.

It is important to note that industries related to chemical engineering are increasingly receptive to new technologies, and strive to find ways to upgrade their software by upgrading or outlining new ways to achieve the objectives.

The present master's dissertation on Chemical and Biochemical Engineering goal as to exemplify a software migration process from an Excel file from a titration reaction to an elaborate java software and its advantages over the original.

Was concluded that one of the great advantages of the new technology is the accessibility, as it allows access on any device with an Internet connection and allow multiple users simultaneously browse, modify and insert data without the issue of conflicts occur between the various users.

Keywords: technology, Java, Python, C#, migration, software, Excel, accessibility, Internet, users.

Índice Geral

1.	Enquadramento teórico e objetivos.....	1
2.	Introdução.....	3
2.1.	Morphis	3
2.2.	Migração	3
2.3.	Migração de Software.....	5
2.4.	Evolução Histórica das Linguagens de Programação.....	6
2.5.	SQL.....	7
2.6.	Java	8
2.7.	Linguagem de marcação	9
2.8.	Ferramentas da Qualidade	9
3.	Ferramentas Informáticas.....	13
3.1.	SQL Developer	13
3.2.	SQL Developer Data Modeler	13
3.3.	GIT	14
3.4.	Eclipse	15
3.5.	Apache Tomcat.....	16
4.	Metodologias	17
4.1.	Java	17
4.2.	Titulação	19
5.	Apresentação e Discussão dos Resultados.....	35
5.1.	Criação da BD	35
5.2.	Repositórios de Compostos	36
5.3.	Curva de Titulação e Determinação do PH.....	38
5.4.	Controle de Qualidade.....	41
6.	Conclusões	43
7.	Trabalho Futuro	44
8.	Bibliografia.....	47
Anexos	53	
	Anexo A – Cálculo Curva Titulação a partir de Equações	54

Anexo B –Função Auxiliar Cálculo PH a partir de Equações.....	59
Anexo C– Métodos para o Cálculo do pH através da determinação do Ponto de inflexão	60
Anexo D – Estudo da Dow Chemical	70

Índice de Figuras

Figura 1- Exemplo Carta de Controle.....	10
Figura 2-Zonas ISO 8258(1991)[33]	11
Figura 3- SQL Developer.....	13
Figura 4- SQL Developer Data Modeler.....	14
Figura 5- BitBucket Git	14
Figura 6-SmartGit Software.....	15
Figura 7-SourceTree Software	15
Figura 8-Eclipse.....	16
Figura 9- Apache Tomcat.....	16
Figura 10-Estrutura MVC [23].....	18
Figura 11- Fluxograma de Gravação da introdução de Marcadores ou dos Compostos	21
Figura 12- Fluxograma de Cartas de Controlo.....	22
Figura 13- Fluxograma da Simulação da Titulação.....	23
Figura 14- Fluxograma titulação a partir das equações.....	24
Figura 15-Eschema BD	35
Figura 16-Repositório de Compostos CurTiPot	36
Figura 17-Repositório de Marcadores CurTiPot.....	36
Figura 18- Repositório Compostos.....	37
Figura 19-Repositório Marcadores	37
Figura 20- Simulação CurTiPot	38
Figura 21-Calculo Curva de Titulação e PH.....	39
Figura 22-Curva de Titulação	39
Figura 23- Calculo Curva a partir de Equações	40
Figura 24- Gráfico CurTiPot	40
Figura 25-Pontos Curva Experimental	41
Figura 26-Inserção cartas de controle.....	42
Figura 27- Carta de Controlo.....	42

Índice de Tabelas

Tabela 1-Resultados Cp.....	12
Tabela 1-Resultados Cpk.....	12
Tabela 1- Símbolos Fluxograma	20
Tabela 2-Validação dos Resultados [47].....	39

Lista de abreviaturas, siglas e símbolos

BD - Base de Dados

IDE - Integrated Development Environment

BO - Business Object

SBO -Simple Business Object

DBO -Database Business Object

MVC - Model-View-Controller

BL- Business Logic

σ - Desvio Padrão

X- Media

1. Enquadramento teórico e objetivos

A presente dissertação foi realizada no âmbito de um estágio curricular com duração de seis meses, na SoftMorphis, Lda.

Esta empresa é responsável pela implementação e conversão de vários projetos de Migração Software de forma a modernizar as ferramentas informáticas para uma gama variada de indústrias.

Das várias linguagens de programação existentes foi selecionado o Java como sendo o indicado para proceder a migração do software.

Um dos objetivos deste trabalho foi a migração para uma aplicação semelhante de um ficheiro de Excel que permita a um utilizador traçar a curva de titulação em três situações distintas, de forma ao utilizador simular ou introduzir dados, que são:

- Simulação a partir da concentração e volume do titulante
- Introdução de dados experimentais
- Introdução de equações do tipo linear, logarítmica e exponencial para vários intervalos de PH

O segundo objetivo foi a determinação do pH a partir da obtenção das curvas.

O terceiro objetivo foi validar se a Framework da Morphis seria funcional para o tipo de aplicação que se pretendia implementar além de validar se as bibliotecas que a mesma utiliza tem a capacidade de tratamento das diferentes tipo de funcionalidades para a área de engenharia.

1. Enquadramento teórico e

2. Introdução

2.1. Morphis

Em 1995 uma equipa de jovens graduados liderados pelo fundador e CEO da Morphis começaram a desenvolver software com a missão de modernizar as ferramentas informáticas a nível mundial. Ao fim de 20 anos a empresa foi transformada numa nova companhia, tendo sido fundada em 2013 a SoftMorphis, LDA [2].

Rapidamente a Morphis obteve a atenção de grandes parceiros no qual destacam-se Microsoft, IBM, Oracle, Fujitsu e Cap Gemini, tem clientes em varias áreas desde saúde, seguros, telecomunicações, entre outros.

Dado a boa reputação estabelecida desde 1995 e que a procura mundial para a modernização de tecnologia a Morphis expandiu-se rapidamente. Atualmente tem escritórios em Inglaterra, Espanha, Brasil e nos Estados Unidos.

Os Serviços, o Software e as análises efetuadas pela Morphis ajudam os seus clientes a atualizarem os seus sistemas antiquados para sistemas web, integraveis som soluções cloud e mobile, libertando-os das dependêndicas de tecnologias e arquitecturas ultrapassadas..

2.2. Migração

A indústria de modernização de sistemas de informação contempla as empresas que realizam serviços ou disponibilizam produtos para auxiliar na conversão de um determinado sistema existente num outro que satisfaça os mesmos requisitos, mas que assenta sobre *standards* e tecnologias atuais no mundo das TI.

Tipicamente os sistemas alvo de modernização designam-se por sistemas ‘legados’ – sistemas com muito anos de existência, com o acumulado de funcionalidades e alterações efetuadas ao longo do tempo ou que não têm suporte por parte do fabricante, e que frequentemente apresentam os seguintes problemas:

- Falta de documentação;
- Desaparecimento das pessoas que o construíram inicialmente;
- Uso de tecnologias/arquiteturas obsoletas ou descontinuadas pelos fabricantes;
- Limitações funcionais face à realidade atual.

2. Introdução

As principais motivações de uma instituição (empresa, organismo público, etc.) desejar migrar o seu sistema são normalmente:

- Tecnologias legacy que podem estar descontinuadas e que não permitem evolução rápida no contexto atual das TI;
- User Interface (UI) antiquado/inadequado à realidade atual;
- Necessidade de disponibilidade global através da Web (Web Application);
- Inexistência de programadores no mercado para as tecnologias do sistema.

No processo de modernização, a instituição envolvida pretende obter um novo sistema no mínimo funcionalmente equivalente ao anterior, com baixo risco. Alguns exemplos de linguagens legacy que são migradas são: Oracle Forms e Cobol.

Tendo isto em conta, é possível fazer uma análise do perfil dos clientes que utilizam este tipo de serviço.

Os clientes de migrações exigem sempre grande celeridade no processo, dado que normalmente existe uma grande pressão interna ou do mercado para apresentar uma nova solução. Essa pressão pode resultar da necessidade de expansão ou adequação do seu negócio (core) a um contexto em evolução, ou a resposta à concorrência agressiva no mercado, quando se tratam de Independent Software Vendors (ISV).

As soluções apresentadas pela indústria das TI para uma modernização podem ser segmentadas em dois conjuntos:

- Soluções que utilizam ferramentas de conversão que realizem todo ou uma parte do processo automaticamente;
- Soluções de conversão manual ou reescrita de código (inclui novos requisitos ou mudanças estruturais), com recurso a um grande número de programadores, muitas vezes realizado por empresas com mão-de-obra barata (e.g. Índia, países de leste).

Questões e Problemas

Quando se procede a migração de um software científico as questões que se colocam são, para que linguagem queremos migrar? Que características a linguagem possui para ter capacidades idênticas ao nosso software?

No caso desta tese de mestrado a resposta à primeira questão já se encontra respondida, tendo sido optado pela linguagem Java. De seguida fala-se um pouco das diferentes características desta linguagem para software científico.

A linguagem Java [50][51][52] é uma linguagem de programação por objetos de fácil uso, que tem o potencial para estender o seu uso ao ramo de códigos científicos. Apesar deste potencial o Java tem algumas limitações.

2. Introdução

A primeira limitação está ligada à memória, dado que a mesma está definida no alocador de memória, que origina que as linhas de um vetor possam estar divididas nas posições de memória, o que prejudica a performance através do baixo comportamento da cache.

A segunda está ligada ao coletor de lixo, que é uma característica do Java. O Java utiliza um processo automático de gestão do coletor de lixo, que pode fazer o software ficar um pouco mais lento em situações de existência de um número elevado de múltiplas camadas no qual os vetores estão definidos, dado que cada linha do vetor é administrada de forma independente das outras existentes.

A terceira tem origem na dificuldade de otimizar programação em dados em paralelo quando está presente um número elevado de múltiplas camadas no qual os vetores estão definidos.

Inicialmente o Java não era recomendado para utilização em softwares científicos dado que não suportava o **IEEE Standard for Floating-Point Arithmetic**, isto é, o Java não respeitava as regras de precisão científica. A partir da Java Virtual Machine 1.2 foi introduzido algum suporte para este tipo de tratamento de dados para dois tipos, *float* e *double*. Apesar desta introdução ainda é necessário mais implementações da linguagem Java por parte da Oracle até ser totalmente complacente com as regras definidas para a precisão científica

2.3. Migração de Software

Existe um procedimento específico definido para que as migrações de software sejam feitas com sucesso, de forma à obtenção de um melhor produto para o cliente. Estas podem ser definidas de várias formas, mas essencialmente todas vão efetuar etapas idênticas até se chegar à fase de entrega para produção da aplicação final para o cliente. Em termos genéricos, as etapas designadas na Morphis são [2]:

- Analyse
- Transform
- Develop

A etapa designada Analyse está dividida em seis subetapas de forma a preparar-se o código para ser transformado para a linguagem pretendida [1].

A primeira subetapa designada Pattern Matching consiste em verificar padrões de linguagem que permitem uma identificação mais rápida da lógica e que possibilite de forma automática converter o código para a nova linguagem com o mínimo de trabalho de verificação necessário.

2. Introdução

A segunda subetapa, Program Slicing, consiste numa técnica que retorna o código original como ficheiro XML do código que foi gerado de forma a se poder consultar o mesmo em caso de necessidade.

Na terceira subetapa, Dependency Evaluation, são identificadas as relações a nível de dados, e entre sistemas para as preservar.

A quarta subetapa, Metrics Evaluation, está diretamente ligada à gestão de tempo, a migração de uma parte do produto consoante a complexidade de códigos, número de ficheiros a migrar para ter uma funcionalidade do programa, e o número de linhas.

Após esta fase é possível identificar técnicas que se devem utilizar, definir estratégias e riscos de erros na migração. Esta subetapa permite também uma melhor indicação ao cliente dos prazos que se compromete a entregar o produto.

A quinta subetapa, View Exploration, permite analisar quatro condições:

- ✓ Requerimentos de infraestruturas (Servidores, Tipo de Base de dados, etc).
- ✓ Código da aplicação inicial
- ✓ Procedimentos Hierárquicos
- ✓ Analisar o código a ser gerado

A sexta subetapa, Flow Visualization, apenas difere da anterior na forma de tratamento de dados, dado que nesta subetapa é utilizado suporte visual, como por exemplos gráficos.

A etapa, Transform, procede as referidas subetapas. Após a identificação de padrões e efetuar as configurações decididas nas subetapas anteriores, é utilizado um processo automático que converte o código da aplicação inicial em código da nova linguagem pretendida.

A etapa, Develop, procede à correção de todos os erros que possam ter ocorrido, procede aos testes e validação das funcionalidades para garantir que o software mantém todas as características, tendo por vezes de rescrever o código inicial. Nesta etapa efetua-se também a adição de código se o cliente pedir novas funcionalidades.

2.4. Evolução Histórica das Linguagens de Programação

As Linguagens de Programação antecedem o computador moderno, apesar de antes do aparecimento do mesmo, a linguagem era apenas em código.

De seguida apresenta-se o ano de lançamento e o nome de algumas das linguagens mais significativas desde a década de 40 [3][4]:

2. Introdução

- 1943-Plankalkül
- 1943-ENIAC coding system
- 1949- C-10
- 1954- FORTRAN
- 1958-LISP
- 1959- COBOL
- 1964-BASIC
- 1970- Pascal e Forth
- 1972- C e Prolog
- 1978- SQL
- 1983- C++
- 1985- Eiffel
- 1987- Perl
- 1991- Python
- 1991- Java e PL/SQL
- 1993-Ruby
- 1994 -ANSI Common Lisp
- 1995-JavaScrip e PHP
- 2000- C#
- 2008-JavaFX Script

2.5. SQL

O SQL [5] [6] (Structured Query Language, ou em português Linguagem de Consulta Estruturada) foi desenvolvido nos laboratórios da IBM na década de 70, e é usada como a linguagem de gestão padrão de Base de dados dado a simplicidade desta linguagem e facilidade de uso.

Esta linguagem tem como elemento diferenciador de outras linguagens de Base de dados na forma como consulta os dados dado que em SQL e especificado a forma do resultado pretendido e não o caminho para se obter o mesmo, além de ser uma linguagem declarativa.

O facto de esta ser declarativa e não procedimental também reduz o tempo necessário de aprendizagem da linguagem.

Desde o aparecimento desta linguagem foi criado um padrão para esta linguagem efetuado pela ANSI (1986) e pela ISO (1987) e tem sido continuamente revisto à medida que a tecnologia evolui de forma a dar resposta a mesma. Um exemplo desta revisão é o SQL:2003 que introduziu as características relacionadas com o XML, por exemplo.

Alguns exemplos de Linguagens de Base de dados do tipo procedimental são Pascal; PHP; Perl; Python; Visual Basic; Cobol; C, Mathematica; Matlab. [7]:

PL/SQL

PL/SQL(Procedural Language/ Structured Query Language)) é uma linguagem que estende o SQL utilizada na gestão de base de dados da Oracle Corporation que permite o manipular os dados [8][9][10].

Destaca-se esta linguagem dado que foi esta a linguagem de Base de Dados utilizada no decorrer desta tese de mestrado.

Transactional Model

Em SQL todas as alterações a uma tabela da Base de Dados, sejam elas a criação, atualização ou apagar os dados são uma transação. Uma transação consiste em unidades ou sequências de trabalho que tem associadas a si uma ordem lógica que é analisada e tratada por um programa da Base de Dados [39] [40].

Isto vai permitir que múltiplos utilizadores possam editar a mesma tabela de dados, dado que as instruções serão feitas de acordo com a ordem lógica definida [39] [40].

As transações têm quatro propriedades base que podem ser descritas pelo acrónimo ACID [39] [40].

A primeira etapa, Atomicity, o objetivo é garantir que todas as operações de uma transação foram concluídas com sucesso, em caso de falhar ocorre um rollback para retornar ao estado inicial a informação.

A segunda etapa, Consistency garante que a Base de Dados alterou com sucesso o estado com a conclusão com sucesso da transação.

A terceira etapa, Isolation, garante que uma transação opera independentemente e de forma autónoma de outras que estejam a ocorrer em simultâneo.

A ultima etapa, Durability garante que o resultado de uma transação ou o seu efeito fica definido em caso de um erro de sistema.

2.6. Java

A linguagem Java é uma linguagem Orientada a objetos desenvolvida pela Sun Microsystems (atualmente Oracle Corporation) e foi lançada em 1995.

Esta linguagem tem como características [11] [12] [13]:

- Orientação a objetos
- Correr em qualquer plataforma que tenha uma maquina virtual
- Recursos de Rede - Possui extensa biblioteca de rotinas que facilitam a cooperação com protocolos TCP/IP, como HTTP e FTP
- Segurança
- Copilada em bytecode [14] [15]

Os bytecode é uma forma intermediária que por sua vez será interpretado e executado por uma máquina virtual, sendo isto o que permite ao java correr em qualquer plataforma.

2.7. Linguagem de marcação

Estas Linguagens são um sistema de anotação de um texto para ser sintaticamente distinguível. Em informática esta linguagem consiste num conjunto de códigos aplicados com o objetivo de adicionar ficheiros específicos.

Alguns exemplos desta linguagem são [16]:

- HTML
- XHTML
- XML

XML

Esta linguagem tem como características [17] [18]:

Separação do conteúdo da formatação

- Simplicidade e legibilidade, tanto para humanos quanto para computadores
- Possibilidade de criação de tags sem limitação
- Criação de arquivos para validação de estrutura
- Interligação a base de dados distintos
- Concentração na estrutura da informação, e não na sua aparência

Destaca-se esta linguagem, dado que esta foi a utilizada no decorrer da presente tese de mestrado.

2.8. Ferramentas da Qualidade

As Sete Ferramentas Básicas de Qualidade são uma designação dada a um conjunto de sete técnicas de interpretação de resultados que permitem agir sob processos de produção.

As sete técnicas referidas são [44] [45]:

- Fluxograma
- Cartas de Controle
- Diagrama de causa-efeito
- Folhas de verificação
- Histogramas
- Gráficos de dispersão
- Diagrama de Pareto

No ambiente desta tese apenas será abordada a técnica Cartas de Controle.

Cartas de Controle

Cartas de Controles é um processo gráfico que segue a norma ISO 8258, estando representado um exemplo gráfico representativo das mesmas:

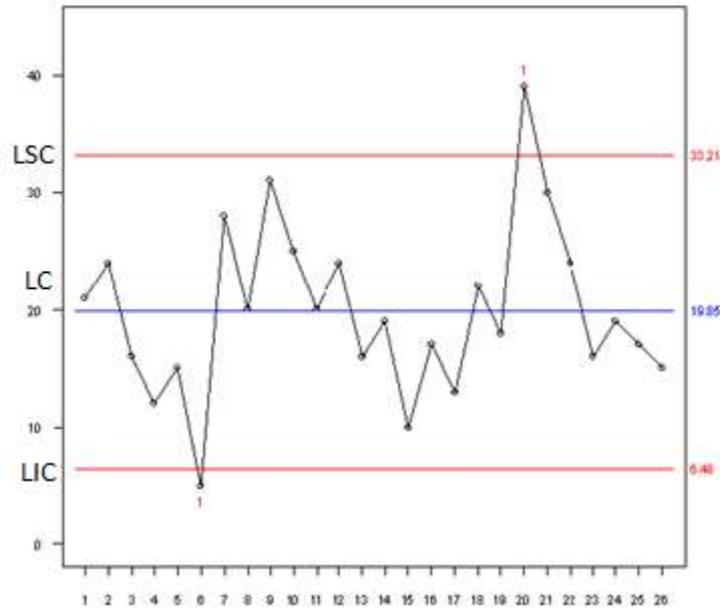


Figura 1- Exemplo Carta de Controle

Os gráficos das Cartas de Controle são compostos por:

- Limite Superior de Controle
- Limite Inferior de Controle
- Limite Central

Em algumas situações além dos três limites mencionados também se podem usar os seguintes limites opcionais, consoante indicação do cliente para o produto pretendido:

- Limite Superior de Especificação.
- Limite Inferior de Especificação.

ISO 8258(1991)

A ISO 8258(1991) determina as condições necessárias respeitar para se poder afirmar que o processo não se encontra sob controlo estatístico. Essas regras são:

3.Ferramentas Informáticas

- Regra 1 – Um qualquer ponto fora dos limites de controlo (limites).
- Regra 2 – Nove pontos consecutivos de um mesmo lado da linha central.
- Regra 3 – Seis pontos consecutivos em sentido ascendente ou descendente.
- Regra 4 – Catorze pontos crescendo e decrescendo alternadamente.
- Regra 5 – Dois de três pontos consecutivos na zona A, do mesmo lado da linha central.
- Regra 6 – Quatro de cinco pontos consecutivos na zona B ou A, do mesmo lado da linha central.
- Regra 7 – Quinze pontos consecutivos na zona C.
- Regra 8 – Oito pontos de ambos os lados da linha central, sem nenhum na zona C.

Análise das Cartas de Controlo: Norma ISO 8258 (1991)

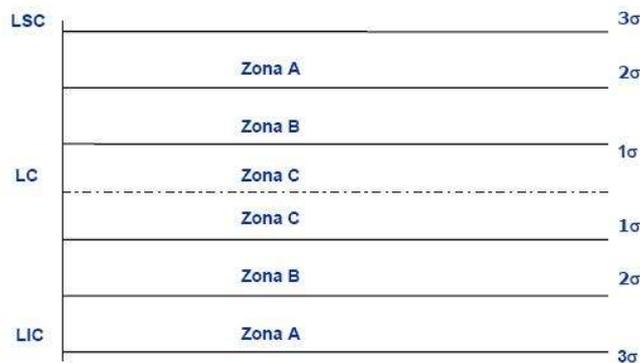


Figura 2-Zonas ISO 8258(1991)[33]

Índices de Capacidade(Cp e Cpk)

A capacidade de um processo[53] pode ser definida como sendo a capacidade de um processo de obter sempre o produto desejado durante um longo período de tempo e sob um determinado conjunto de condições. Isto permite verificar se um processo de fabricação específico é ou não viável e sustentável. Recorrendo a estes índices e possível tomar decisões sobre a forma como interagir com o processo de forma a se obter o produto desejado ou a manutenção ou não da produção de linhas de produção. Este limites são definidos a partir dos limites de especificação definidos para o processo.

O primeiro índice que vou abordar é o Cp que é o índice mais simple, dado não ser sensível as causas especias de alteração do processo e não ter em conta a centralizão do processo. Quanto maior este valor menor será a probabilidade de o processo estar fora das especificações pretendidas.

3.Ferramentas Informáticas

O Cp é dado pela seguinte Equação:

$$Cp = \frac{LSE - LIE}{6\delta}$$

Equação 1-Cálculo Cp

Na seguinte tabela estão representados as possíveis conclusões a partir do Cp

	Cp <1	1 ≤ Cp <1,33	Cp ≥1,33
Conclusão	Processo incapaz	Processo Aceitavel	Processo Capaz

Tabela 1-Resultados Cp

O segundo índice é o Cpk que tem em conta a centralização do processo e as causas especiais de alterar o processo. Este funciona como o ajuste do índice mencionado anteriormente para distribuições não centradas entre os limites de especificação

O Cpk é dado pela seguinte Equação:

$$Cpk = \text{MIN}\left(\frac{LSE - X}{3\delta}, \frac{X - LIE}{3\delta}\right)$$

Equação 2-Cálculo Cpk

Na seguinte tabela estão representados as possíveis conclusões a partir do Cpk

	Cpk <1	1 ≤ Cpk <1,33	Cpk ≥1,33
Conclusão	Processo incapaz	Processo Aceitavel	Processo Capaz

Tabela 2-Resultados Cpk

Alem das conclusões anteriores tambem podemos concluir se o processo esta ou não centrado , isto e possivel obter ao se verificar se os dois valores do Cpk(Cpk max, Cpk min) são o mesmo ou se esta mais deslocado para a direita ou esquerda

Com Base nestes metodos é possivel verificar se os processos estão sobre controlo estatistico ou se são ou não capazes, isto permite tomar ações sobre o processo de forma a maximizar a produção do produto pretendido com o minimo de produto não conforme.

3. Ferramentas Informáticas

Neste capítulo, serão descritos de forma sumária algumas das ferramentas utilizadas para a elaboração do software e a utilização do tipo de dados e respetivo tratamento.

3.1. SQL Developer

O SQL Developer é um software gráfico para gestão de base de dados Oracle que permite aos utilizadores e administradores da base de dados elaborar as suas tarefas [19].

Esta ferramenta pretende:

- Aumentar a produtividade
- Diminuir tempo
- Maximizar o retorno do investimento em base de dados Oracle

Na Figura 3 está representado o aspeto visual do SQL Developer com múltiplas Base de dados configuradas

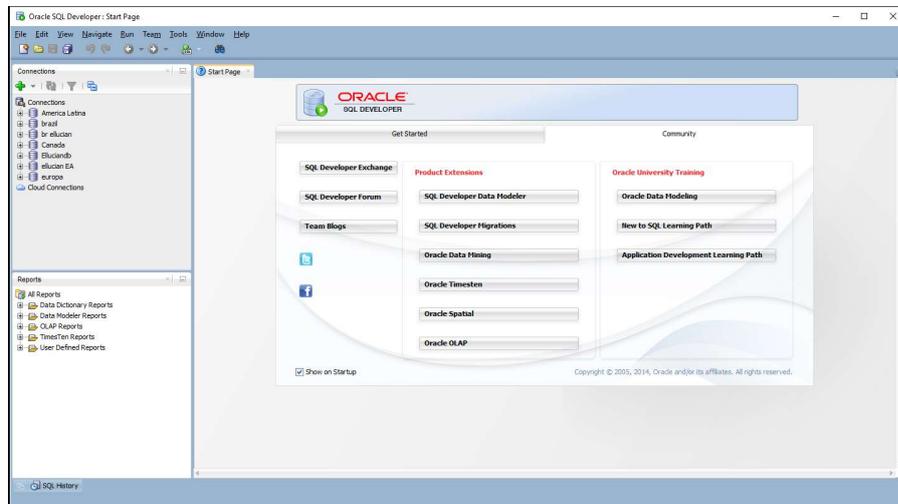


Figura 3- SQL Developer

3.2. SQL Developer Data Modeler

O Data Modeler é um software gratuito que permite aos utilizadores criarem, consultarem, modificarem e criarem modelos lógicos, relacionais, entre outros. Esta ferramenta pode ser utilizada em ambientes físicos ou em ambientes Cloud [20]. O Aspeto visual da configuração da BD utilizada e do Data Modeler está representado na figura 4.

3.Ferramentas Informáticas

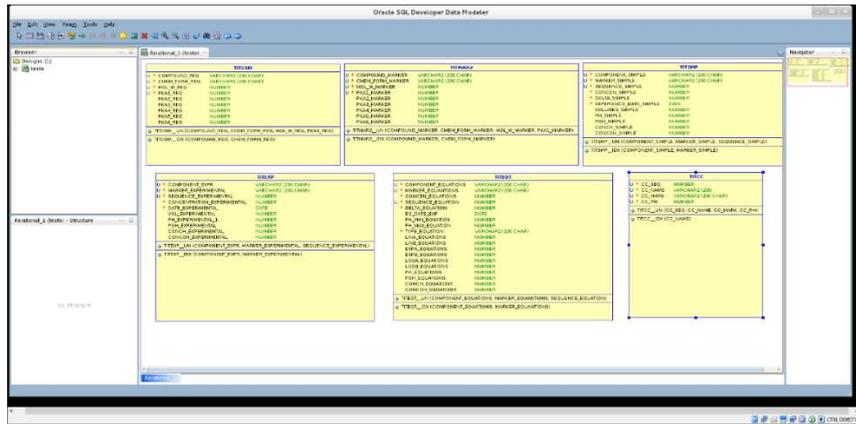


Figura 4- SQL Developer Data Modeler

3.3. GIT

O GIT é um sistema que permite gerir o projeto, que tem na cloud um repositório com uma cópia do nosso projeto a nível local. Este serviço foi desenvolvido em 2005 para sistemas Linux [21].

Existem vários sites que disponibilizam este serviço gratuitamente, no âmbito desta tese foi usado: <https://bitbucket.org/> , o aspeto visual deste serviço está representado na figura 5.

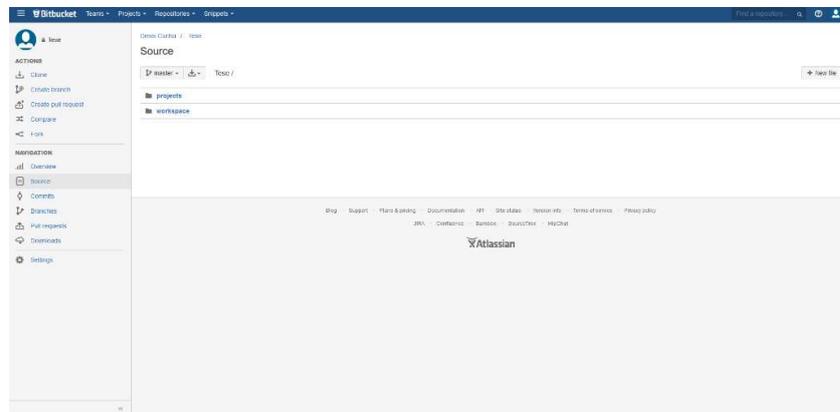


Figura 5- BitBucket Git

De forma a interagir com o GIT de uma forma mais eficiente existem vários softwares nos quais destaco:

O SmartGit é um software de interface de gestão do repositório GIT para Windows, Mac e Linux, sendo apenas gratuito para utilizadores sem fins comerciais, esse software está representado na figura 6 [22].

3.Ferramentas Informáticas

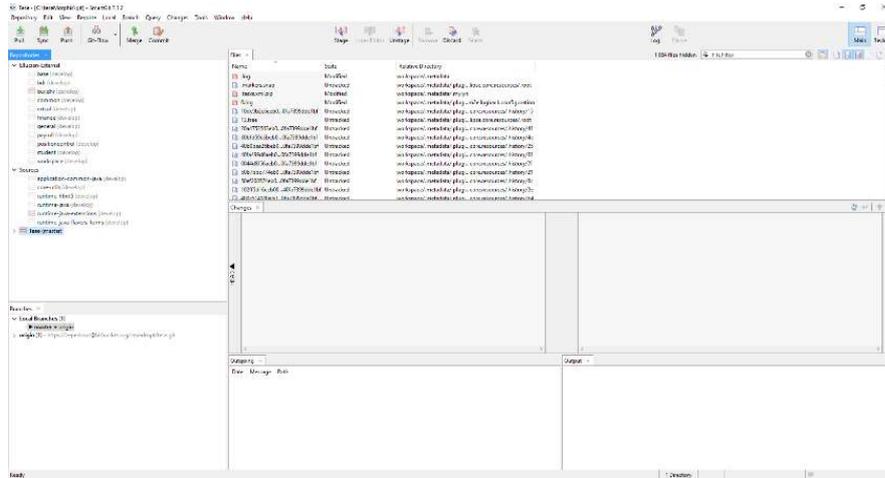


Figura 6-SmartGit Software

O segundo software é o SourceTree, que é semelhante ao SmartGit, e funciona em Windows e Mac fornecendo uma interface gráfica para o repositório GIT ou Hg. Este software tem como vantagem ser gratuito dado que é um software open Source e está representado na figura 7 [23].

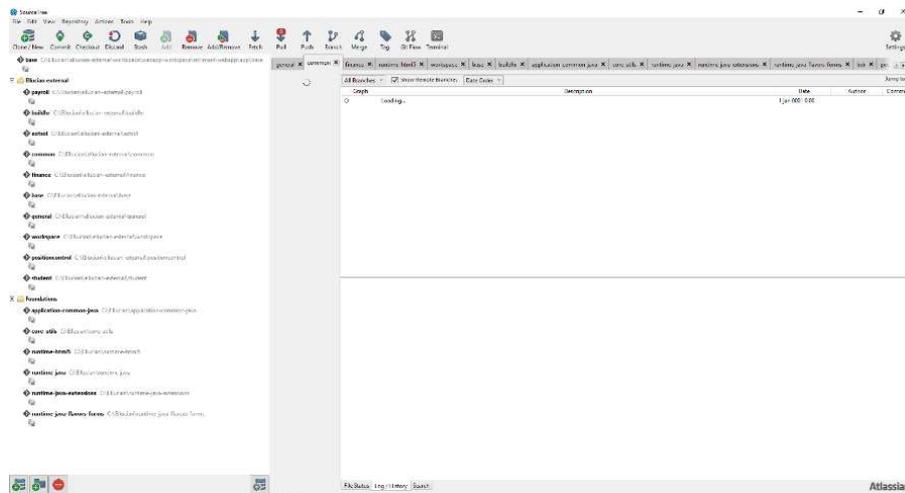


Figura 7-SourceTree Software

3.4. Eclipse

O Software Eclipse foi desenvolvido inicialmente pela IBM que optou por o doar como open source. Eclipse é um IDE para desenvolvimento em Java, mas que suporta outras linguagens através de Plugins (por exemplo: Python) [24] [25] [26] [27].

O interface visual deste IDE está representado na figura 8.

3.Ferramentas Informáticas

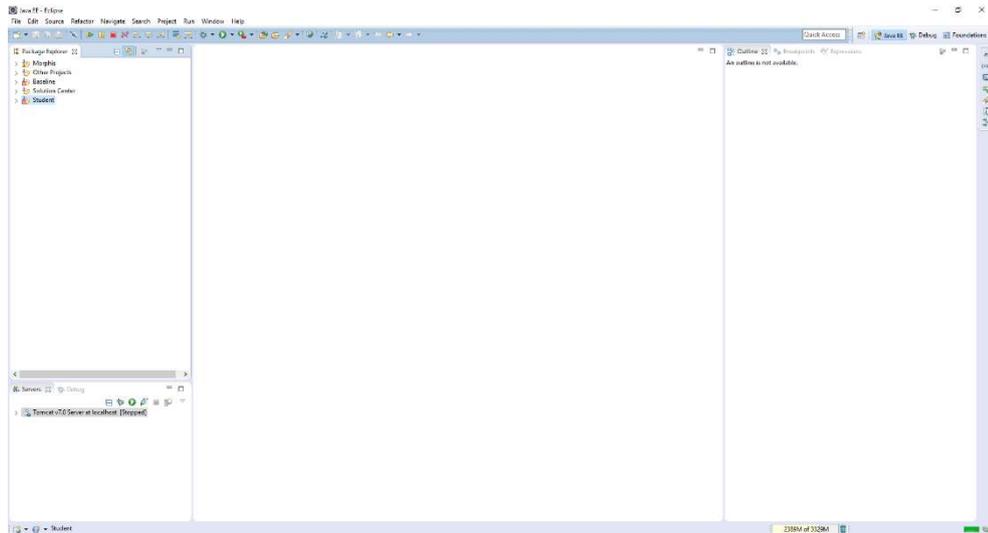


Figura 8-Eclipse

3.5. Apache Tomcat

O Tomcat é um servidor web java desenvolvido pelo Apache Software Foundation que o distribui como open source e encontra-se instalado no Eclipse, ficando simbolizado como está representado na figura 9 [29].

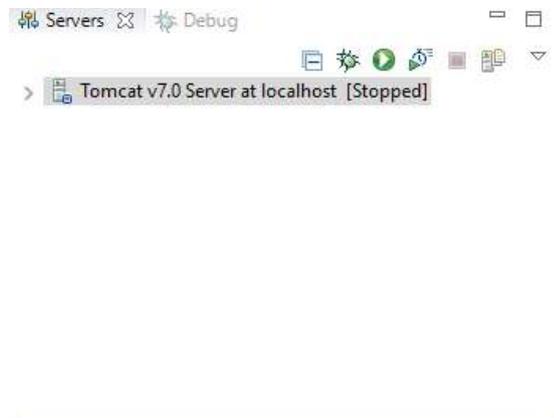


Figura 9- Apache Tomcat

4. Metodologias

O que é a metodologia, para os autores das Ciências sociais “metodologia é o estudo do método. Para os sociólogos, presume-se que seja estudar os métodos de fazer pesquisa sociológicas, de analisar o que pode ser descoberto através delas e o grau de confiabilidade do conhecimento assim adquirido, e de tentar aperfeiçoar estes métodos através da investigação fundamentada e da crítica de suas propriedades” (Becker, 1993) [28],

Para definir método e metodologias existem vários autores na literatura com diferentes interpretações:

- Gil (1994)
- Thiollent (1994)
- Jung (2004)

Neste trabalho adotou-se a consoante de Jung que diz:” Método é 'uma maneira de como se fazer algo'; desta forma tratando-se da prática científica é necessário, a existência e a aplicação de um método. Este método consiste num conjunto de etapas ordenadamente dispostas a serem executadas que tenham por finalidade a investigação de fenómenos naturais para a obtenção de conhecimentos.”

4.1. Java

Neste ponto pretende-se dar uma breve noção sobre o tipo de Objetos do Java (Simple Business Objects, Database Business Objects), e sobre o método Model-View-Controller que foi utilizado na elaboração desta tese.

Business Objects

Um BO é uma entidade que se encontra no software com multicamadas que funciona em simultâneo com o acesso aos dados e a Business Logic [35] [33] [34].

A BL por sua vez é a parte da aplicação que determina como pode ser criado, gravado, editado e visualizado os dados de acordo com as regras de negócio que estejam definidos na finalidade do software.

DBO

Um DBO são objetos em java que tem correspondência em colunas de uma BD, tendo de se configurar um manager para interagir com a BD de forma a obter, modificar e apagar dados de acordo com as especificações.

SBO

Um SBO são objetos em java que não tem correspondência em colunas de BD.

MVC

O MVC [37] [38] [36] é um padrão de arquitetura de software para implementação de interfaces para utilizadores, que é dividido em três partes que se encontram interligadas:

Model

O Model gere o comportamento e os dados do domínio da aplicação, responde aos pedidos de informação do estado originado pela View e responde as instruções de alteração de estado pelo Controller.

View

A View gere a maneira como se observa a informação obtida.

Controller

O Controller gere e interpreta todas as ações efetuadas pelos utilizadores (rato e teclado), alterando o modelo e as View de acordo com o pretendido.

O MVC pode ser exemplificado pela seguinte imagem:

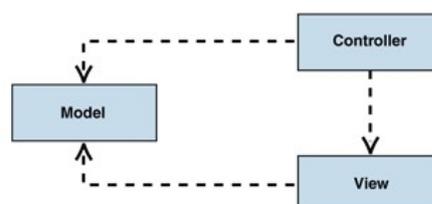


Figura 10-Estrutura MVC [23]

4.2. Titulação

A Titulação é uma técnica laboratorial de análise química quantitativa que permite determinar a concentração de uma solução de um reagente conhecido.

Este método consiste em fazer reagir um volume conhecido de uma solução (titulante) com as suas características conhecidas (concentração, volume e número de moles) com uma amostra a solução da qual se desconhece a reação (titulado) [30] [31] [32].

Formulas Matemáticas

As equações matemáticas representadas são algumas das equações correspondentes às que foram utilizadas posteriormente de forma a ser calculado a curva de titulação e o valor de PH.

Equação para cálculo da Concentração

$$C = \frac{n}{v}$$

Equação 3-Cálculo Concentração

Equações para cálculo do pH

$$pH = -\log([H^+])$$

Equação 4- Cálculo pH 1

$$pH = pKA + \log\left(\frac{[A^-]}{[HA]}\right)$$

Equação 5- Cálculo pH 2

$$pH = \frac{1}{2}[Pka(x) + Pka(x + 1)]$$

Equação 6- Cálculo pH 3

Equações para cálculo do pOH

$$pOH = -\log([OH^-])$$

Equação 7- Cálculo pOH

Equações que relaciona o pH com o pOH

$$14 = pH + pOH$$

Equação 8- Relação entre pH e pOH

Constante de equilíbrio de acidez

$$K_A = \frac{[A^-] \times [H_3O^+]}{[HA]}$$

Equação 9- Constante de Equilíbrio

Equação para cálculo da Concentração de H⁺

$$[H^+] = 10^{-pH}$$

Equação 10- Cálculo H⁺

Equação para cálculo da Concentração de OH⁻

$$[OH^-] = 10^{-pOH}$$

Equação 11- Cálculo OH⁻

Diagramas de Decisão

Nesta secção serão mostrados alguns dos diagramas de decisão que foram necessários elaborar de forma a se analisar a sequencia correta para o código ser elaborado.

Os Diagramas de decisão ou Fluxogramas são compostos por símbolos que representam as diferentes etapas a efetuar. Esses símbolos são:

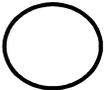
	Terminador
	Introdução Manual
	Conexão
	Decisão
	Processo
	Input/Output

Tabela 3- Símbolos Fluxograma

Nas Figuras 11,12,13 e 14 estão representados os fluxogramas de: Gravação e introdução de Marcadores e Compostos, Cartas de Controlo, Simulação da Titulação e titulação a partir das equações.

4. Metodologias

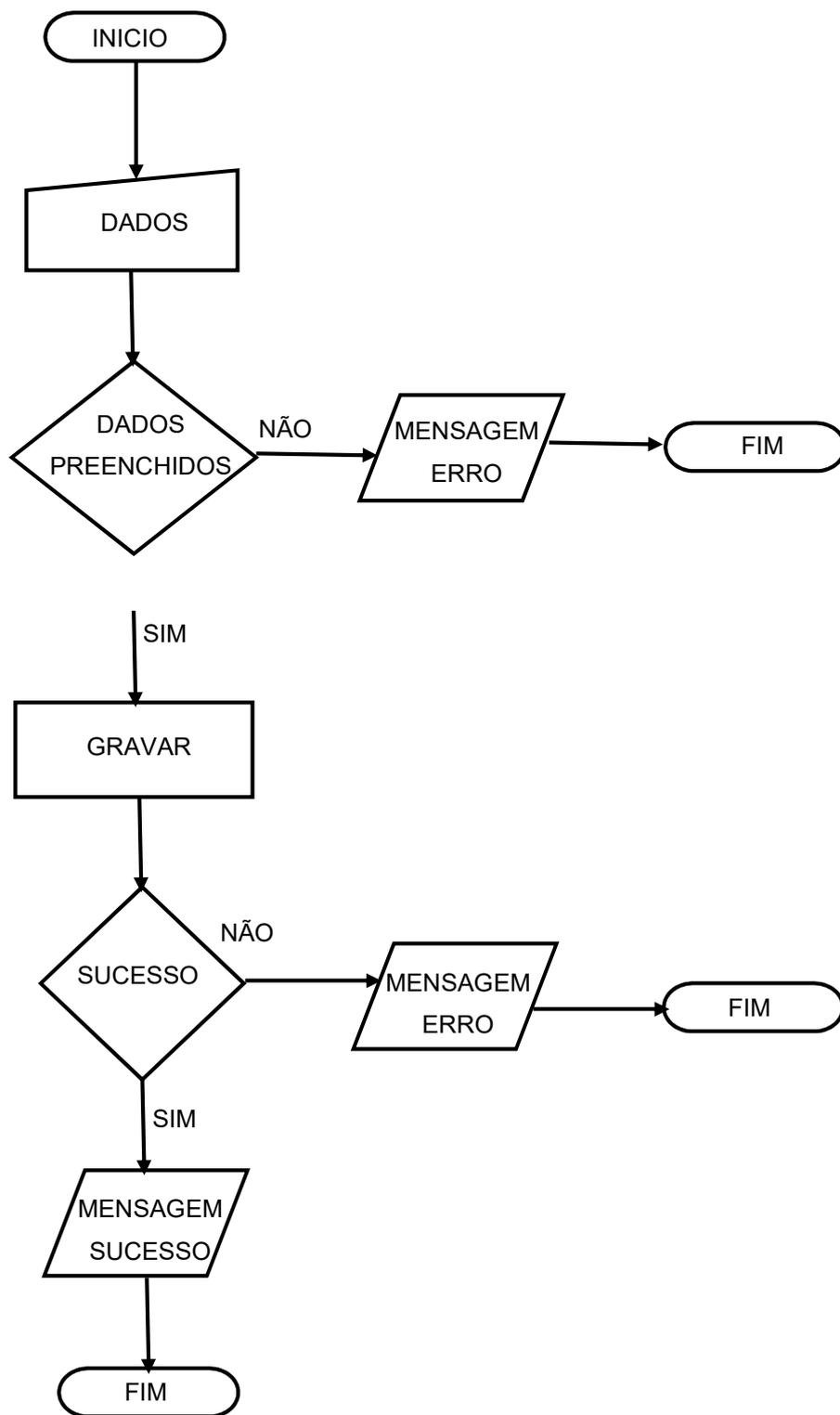


Figura 11- Fluxograma de Gravação da introdução de Marcadores ou dos Compostos

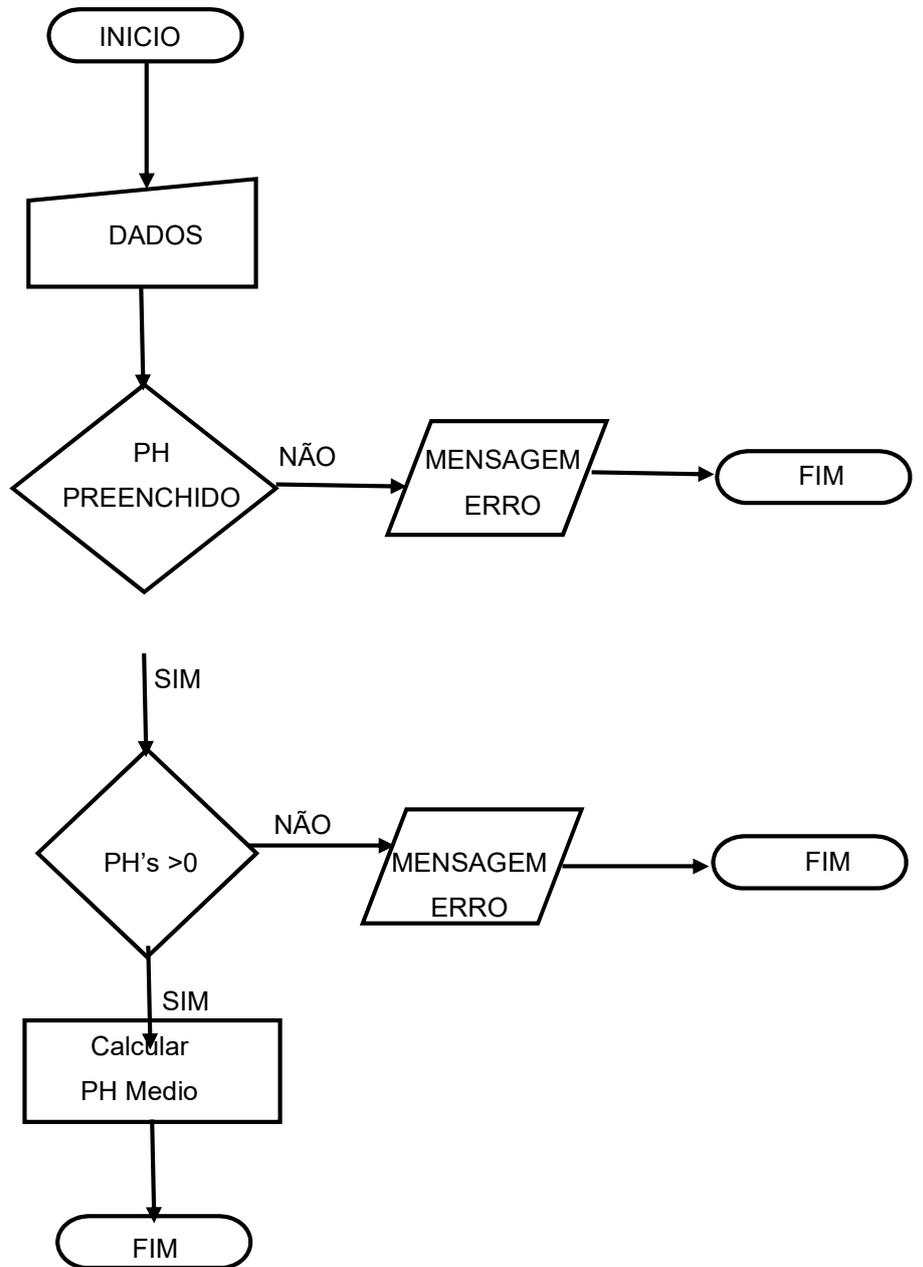


Figura 12- Fluxograma de Cartas de Controle

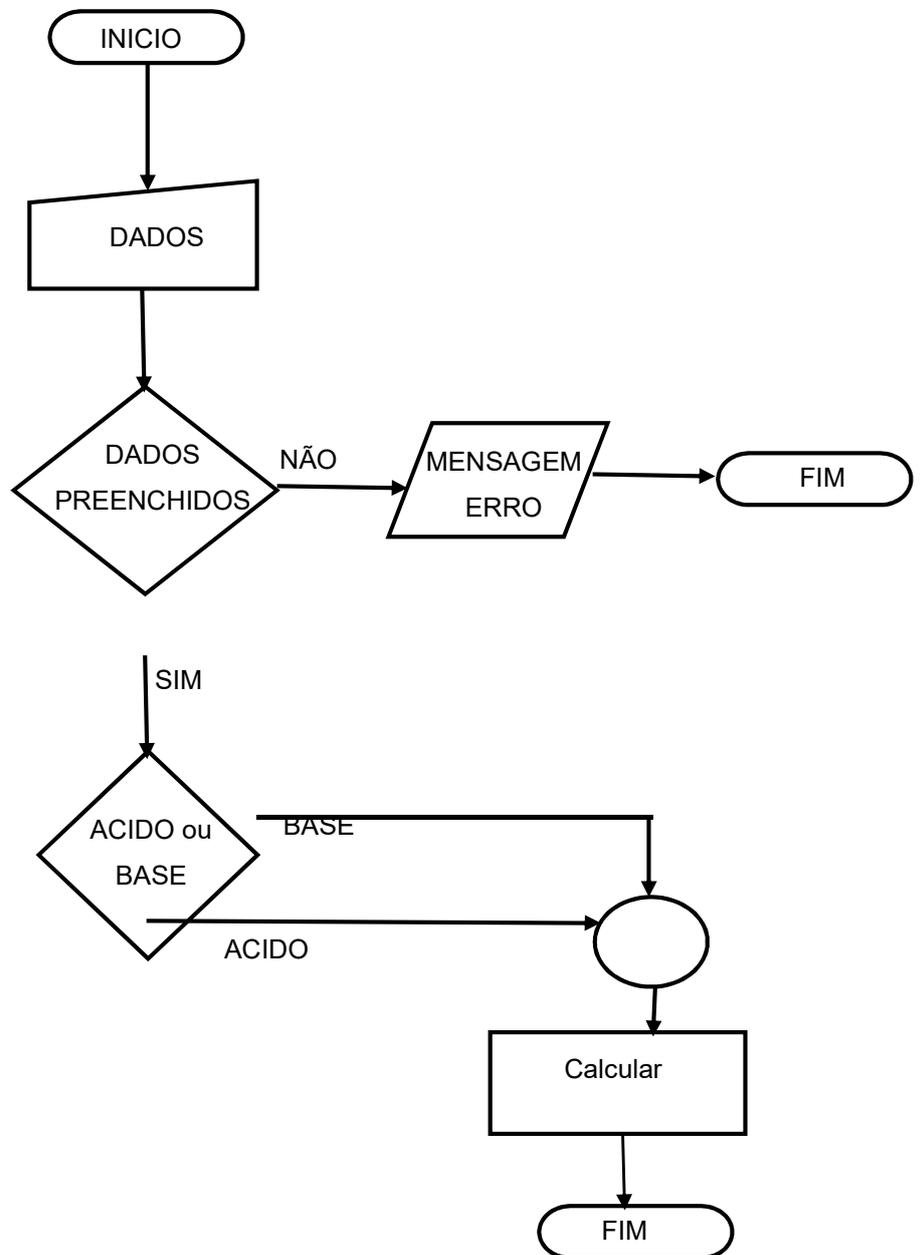


Figura 13- Fluxograma da Simulação da Titulação

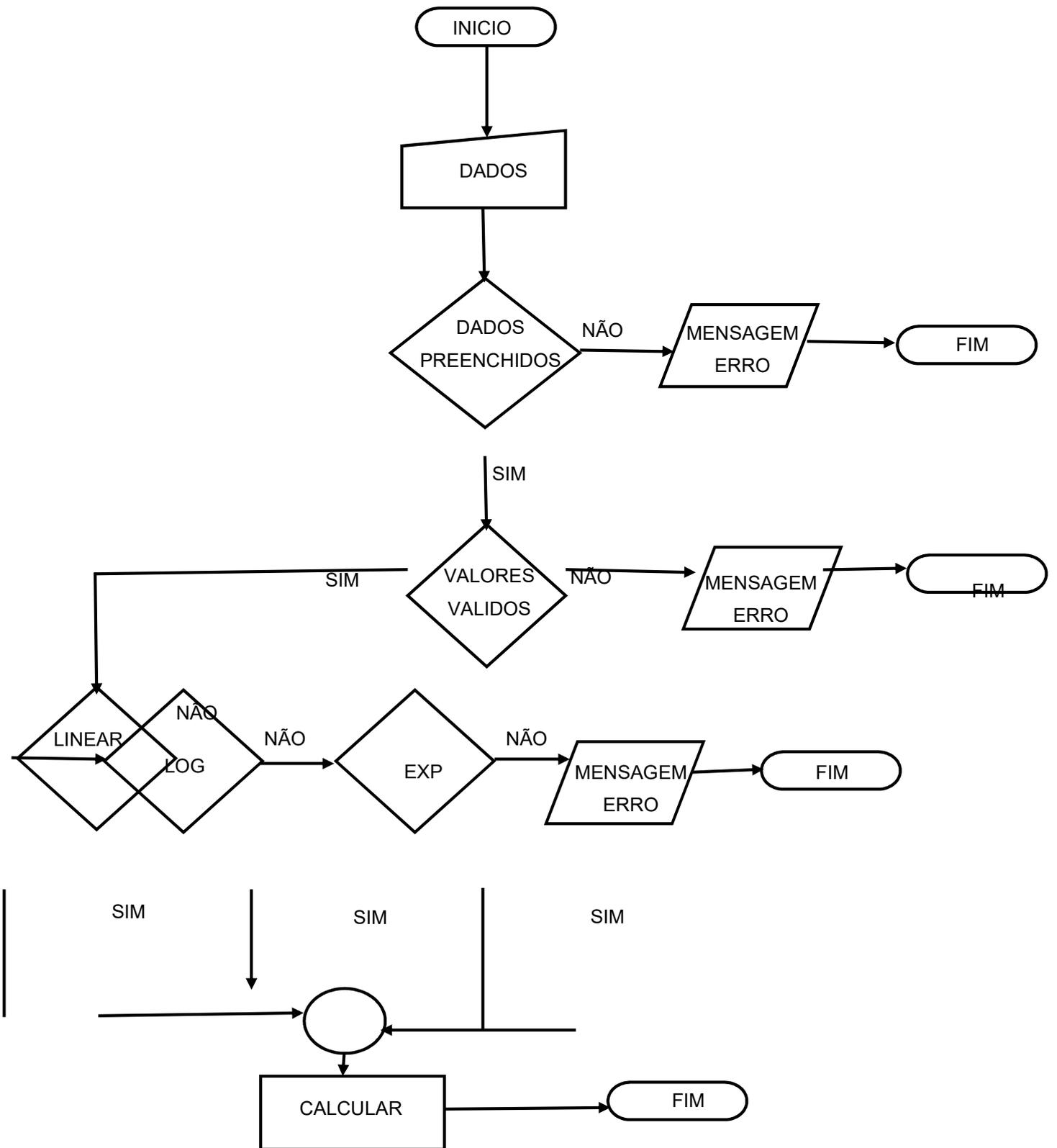


Figura 14- Fluxograma titulação a partir das equações

4. Metodologias

Estes diagramas têm em conta as validações dados introduzidos, as ações que efetua e as suas posteriores validações

Métodos Utilizados na Migração

Neste ponto vai ser colocado o código que define as equações acima mencionadas em alguns dos métodos mais importantes de forma a calcular as diferentes funcionalidades.

Cálculo pH e pOH

```
public void lib_calculo(BigDecimal ctitulado, BigDecimal ph,
BigDecimal poh, NString n){

    Double aux=null;
    Double ctitu=ctitulado.doubleValue();

    lib_log(aux, ctitu );

    BigDecimal aux2= new BigDecimal(aux);

    BigDecimal pw= new BigDecimal(14);

    if(n.equals("PH")){

        ph=aux2;
        poh=pw.subtract(ph);
    }
    else{

        poh=aux2;
        ph=pw.subtract(poh);
    }

}
```

Este método baseia-se no cálculo do pH através da indicação por parte do utilizador se a solução a ser titulada é um ácido ou uma base.

Cálculo logaritmo

```
public void lib_log(Double aux, Double ctitulado ){

    aux=-(Math.log10(ctitulado));

}
```

Este método utiliza a biblioteca do java para calcular o valor do logaritmo. Foi necessário utilizar os métodos originais do java dado atualmente a biblioteca matemática da Morphis não ter tratamento para este tipo de função

Cálculo da simulação da Curva de titulação

```

    @ActionTrigger(action="WHEN-BUTTON-PRESSED",
item="TIT2_CALC_BTN")
    public void titcalcsimple_buttonClick()
    {

TIT2Adapter tit2Element=(TIT2Adapter)
this.getFormModel().getTIT2().getRowAdapter(true);

TITSMPAdapter tit3Element=(TITSMPAdapter)
this.getFormModel().getTITSMP().getRowAdapter(true);

if(tit3Element.getcompsimple().isNull() || tit3Element.getintvolsimple().
.isNull() ||

tit3Element.getconcsolsimple().isNull() || tit3Element.gettitsmpsolt().i
sNull() || tit3Element.getvolsolution().isNull() || tit3Element.getconsolt
it().isNull()){
errorMessage("Missing Data in Required Fields: Coumpound Name, Interval
Volume, Titration Solution Type, Titration and Solution
Concentracion, Titration solution volume");
        throw new ApplicationException();
    }

    NNumber vol=new NNumber(0);
    BigDecimal ctitulante= new
BigDecimal(toStr(tit3Element.getconcsolsimple()).getValue());

        if(tit3Element.gettitsmpsolt().equals("PH")){
            goItem("TIT2.VOLUMES_SIMPLE2");

            while(tit2Element.getphcalcsimple2().isNull() || tit2Element.getph
calcsimple2().less(14)){

                BigDecimal nH= new BigDecimal(0);
                BigDecimal nOH= new BigDecimal(0);

                nH=tit3Element.getconsoltit().getValue().multiply(tit3Element.ge
tvolsolution().getValue()).divide(toNumber(1000).getValue());
                BigDecimal vcontit= new
BigDecimal(0);

                while(tit2Element.getphcalcsimple2().less(toNumber(7)) ||
tit2Element.getphcalcsimple2().isNull()){

                    nextRecord();

                    if(toNumber(vcontit).equals(0)){

                        tit2Element.setvolcalcsimple2(toNumber(vcontit));

                        tit2Element.setphcalcsimple2(toNumber(0));

                        tit2Element.setpohcalcsimple2(toNumber(14));

                        tit2Element.setconchcalcsimple2(toNumber(1));

                        tit2Element.setconcohcalcsimple2(toNumber(0.000000000000001));

                        vcontit=vcontit.add(tit3Element.getintvolsimple().getValue());
                    }
                }
            }
        }
    }

```

4. Metodologias

```

                                else{

nOH=tit3Element.getconcsolsimple().getValue().multiply(vcontit).divide
(toNumber(1000).getValue());
                                BigDecimal v=
vol.getValue();

                                if (toNumber(nH).notEquals(toNumber(nOH)) &&
toNumber(nH).greater(toNumber(nOH))) {
                                BigDecimal n= new
BigDecimal(0);

                                v=vcontit.add(tit3Element.getvolsolution().getValue());

                                n=nH.subtract(nOH);

tit2Element.setconchcalcsimple2(trunc(((toNumber(n).divide(toNumber(v)
)).multiply(toNumber(1000).getValue()),7));
                                Double aux=null;
                                Double
aux2=null;
                                Double
ctitu=tit2Element.getconchcalcsimple2().getValue().doubleValue();
                                aux=-
(Math.log10(ctitu));

tit2Element.setphcalcsimple2(toNumber(aux));

tit2Element.setpohcalcsimple2(toNumber(14).subtract(tit2Element.getphc
alcsimple2()));

                                aux2=
Math.pow(toNumber(10).getValue().doubleValue(),
tit2Element.getpohcalcsimple2().getValue().doubleValue());

tit2Element.setconcohcalcsimple2(trunc(toNumber(aux2).divide(toNumber(
1000)),7));

tit2Element.setvolcalcsimple2(toNumber(vcontit));

vcontit=vcontit.add(tit3Element.getintvolsimple().getValue());

                                }else{

                                if (toNumber(nH).notEquals(toNumber(nOH)) &&
toNumber(nH).lesser(toNumber(nOH))) {

tit2Element.setphcalcsimple2(toNumber(7));

tit2Element.setpohcalcsimple2(toNumber(7));

tit2Element.setconchcalcsimple2(toNumber(0.0000001));

tit2Element.setconcohcalcsimple2(toNumber(0.0000001));

tit3Element.setphcalcsimple(toNumber(7));

tit3Element.setpohcalcsimple(toNumber(7));

```

4. Metodologias

```
tit3Element.setconchcalcsimple(toNumber(0.0000001));
tit3Element.setconcohalcsimple(toNumber(0.0000001));
tit2Element.setvolcalcsimple2(toNumber(vcontit));
tit3Element.setvolcalcsimple(toNumber(vcontit));
}
}

if(toNumber(nH).equals(toNumber(nOH))) {

tit2Element.setvolcalcsimple2(toNumber(vcontit));
tit2Element.setphcalcsimple2(toNumber(7));
tit2Element.setpohcalcsimple2(toNumber(7));
tit2Element.setconchcalcsimple2(toNumber(0.0000001));
tit2Element.setconcohalcsimple2(toNumber(0.0000001));
tit2Element.setvolcalcsimple2(toNumber(vcontit));
tit3Element.setvolcalcsimple(toNumber(vcontit));
tit3Element.setphcalcsimple(toNumber(7));
tit3Element.setpohcalcsimple(toNumber(7));
tit3Element.setconchcalcsimple(toNumber(0.0000001));
tit3Element.setconcohalcsimple(toNumber(0.0000001));
vcontit=vcontit.add(tit3Element.getintvolsimple().getValue());
}
break;
}

}

while(tit2Element.getphcalcsimple2().lesser(14)) {
nextRecord();

nOH=tit3Element.getconcsolsimple().getValue().multiply(vcontit).divide
(toNumber(1000).getValue());
Double aux3=null;
Double aux4=null;
NNumber
aux5=toNumber(nOH).subtract(toNumber(nH));

tit2Element.setvolcalcsimple2(toNumber(vcontit));
```

4. Metodologias

```
tit2Element.setconcohalcsimple2(trunc(toNumber(aux5).divide(vcontit).
multiply(toNumber(1000).getValue()),7));
                                Double ctitu2=
tit2Element.getconcohalcsimple2().getValue().doubleValue();
                                aux3=- (Math.log10(ctitu2));

tit2Element.setpohcalcsimple2(trunc(toNumber(aux3),10));

tit2Element.setpohcalcsimple2(toNumber(14).subtract(aux3));
                                aux4=
Math.pow(toNumber(10).getValue().doubleValue(),
tit2Element.getpohcalcsimple2().getValue().doubleValue());

tit2Element.setconchcalcsimple2(trunc(toNumber(aux4),7));

if(tit2Element.getpohcalcsimple2().greaterOrEquals(toNumber(13.8))){

tit2Element.setpohcalcsimple2(toNumber(14));

tit2Element.setpohcalcsimple2(toNumber(0));

tit2Element.setconchcalcsimple2(toNumber(0));

tit2Element.setconcohalcsimple2(toNumber(0.000000000000001));

tit2Element.setvolcalcsimple2(toNumber(vcontit));
                                return;
                                }

vcontit=vcontit.add(tit3Element.getintvolsimple().getValue());
                                }

                                }
                                else{

                                goItem("TIT2.VOLUMES_SIMPLE2");

                                while(tit2Element.getpohcalcsimple2().isNull()||tit2Element.getpoh
calcsimple2().greater(0)){
                                        BigDecimal nH= new BigDecimal(0);
                                        BigDecimal nOH= new BigDecimal(0);

                                        nOH=tit3Element.getconsoltit().getValue().multiply(tit3Element.g
etvolsolution().getValue()).divide(toNumber(1000).getValue());
                                        BigDecimal vcontit= new BigDecimal(0);

                                        while(tit2Element.getpohcalcsimple2().isNull()||tit2Element.getpoh
calcsimple2().greater(7)){
                                                nextRecord();
                                                if(toNumber(vcontit).equals(0)){

tit2Element.setvolcalcsimple2(toNumber(vcontit));
```

4. Metodologias

```
tit2Element.setphtcalcsimple2(toNumber(14));
tit2Element.setpohcalcsimple2(toNumber(0));
tit2Element.setconchcalcsimple2(toNumber(0.000000000000001));
tit2Element.setconcohcalcsimple2(toNumber(1));
vcontit=vcontit.add(tit3Element.getintvolsimple().getValue());
    }
    else{
        nH=tit3Element.getconcsolsimple().getValue().multiply(vcontit).d
        ivate(toNumber(1000).getValue());
        BigDecimal v= vol.getValue();
        if(toNumber(nH).notEquals(toNumber(nOH))
        && toNumber(nOH).greater(toNumber(nH))){
            BigDecimal n= new BigDecimal(0);
            v=vcontit.add(tit3Element.getvolsolution().getValue());
            n=nOH.subtract(nH);
            tit2Element.setconcohcalcsimple2(trunc(((toNumber(n).divide(toNu
            mber(v))).multiply(toNumber(1000).getValue()),7));
            Double aux=null;
            Double aux2=null;
            Double
            ctitu=tit2Element.getconcohcalcsimple2().getValue().doubleValue();
            aux--(Math.log10(ctitu));
            tit2Element.setpohcalcsimple2(toNumber(aux));
            tit2Element.setphtcalcsimple2(toNumber(14).subtract(tit2Element.g
            etpohcalcsimple2()));
            aux2=
            Math.pow(toNumber(10).getValue().doubleValue(),
            tit2Element.getphtcalcsimple2().getValue().doubleValue());
            tit2Element.setconchcalcsimple2(trunc(toNumber(aux2).divide(toNu
            mber(1000)), 7));
            tit2Element.setvolcalcsimple2(toNumber(vcontit));
            vcontit=vcontit.add(tit3Element.getintvolsimple().getValue());
        }
        else{
            if(toNumber(nH).notEquals(toNumber(nOH))
            && toNumber(nOH).lesser(toNumber(nH))){
                tit2Element.setphtcalcsimple2(toNumber(7));
                tit2Element.setpohcalcsimple2(toNumber(7));
                tit2Element.setconchcalcsimple2(toNumber(0.0000001));
                tit2Element.setconcohcalcsimple2(toNumber(0.0000001));
                tit3Element.setphtcalcsimple(toNumber(7));
```

4. Metodologias

```
tit3Element.setpohcalcsimple(toNumber(7));
tit3Element.setconchcalcsimple(toNumber(0.0000001));
tit3Element.setconcohalcsimple(toNumber(0.0000001));
tit2Element.setvolcalcsimple2(toNumber(vcontit));
tit3Element.setvolcalcsimple(toNumber(vcontit));
    }
    }
    if(toNumber(nH).equals(toNumber(nOH))) {

    tit2Element.setvolcalcsimple2(toNumber(vcontit));
    tit2Element.setphcalcsimple2(toNumber(7));
    tit2Element.setpohcalcsimple2(toNumber(7));
    tit2Element.setconchcalcsimple2(toNumber(0.0000001));
    tit2Element.setconcohalcsimple2(toNumber(0.0000001));
    tit2Element.setvolcalcsimple2(toNumber(vcontit));
    tit3Element.setvolcalcsimple(toNumber(vcontit));
        tit3Element.setphcalcsimple(toNumber(7));

    tit3Element.setpohcalcsimple(toNumber(7));
    tit3Element.setconchcalcsimple(toNumber(0.0000001));
    tit3Element.setconcohalcsimple(toNumber(0.0000001));
    vcontit=vcontit.add(tit3Element.getintvolsimple().getValue());

    break;
    }

    }

    }

while(tit2Element.getphcalcsimple2().greater(0))

        nextRecord();

    nH=tit3Element.getconcsolsimple().getValue().multiply(vcontit).d
    ivide(toNumber(1000).getValue());
        Double aux3=null;
        Double aux4=null;
        NNumber
aux5=toNumber(nH).subtract(toNumber(nOH));

    tit2Element.setvolcalcsimple2(toNumber(vcontit));
```

4. Metodologias

```
tit2Element.setconhcalcsimple2(trunc(toNumber(aux5).divide(vcon
tit).multiply(toNumber(1000).getValue()),7));
Double ctitu2=
tit2Element.getconhcalcsimple2().getValue().doubleValue();
aux3=-(Math.log10(ctitu2));

tit2Element.setphcalcsimple2(trunc(toNumber(aux3),10));

tit2Element.setpohcalcsimple2(toNumber(14).subtract(aux3));
aux4=
Math.pow(toNumber(10).getValue().doubleValue(),
tit2Element.getpohcalcsimple2().getValue().doubleValue());

tit2Element.setconcohalcsimple2(trunc(toNumber(aux4),7));

while(tit2Element.getphcalcsimple2().greater(0.3)){
tit2Element.setphcalcsimple2(toNumber(0));

tit2Element.setpohcalcsimple2(toNumber(14));

tit2Element.setconhcalcsimple2(toNumber(0.000000000000001));

tit2Element.setconcohalcsimple2(toNumber(0));

tit2Element.setvolcalcsimple2(toNumber(vcontit));
return;
}
}
}
```

Este método executa as validações dos dados a serem introduzidos, e executa com base no tipo de solução, base ou ácidos, um conjunto de etapas que permite calcular os valores de pH, pOH, e concentrações em função do volume. Dado este método utilizar ciclos do tipo while e em algumas situações originar ciclos infinitos, dado que não atingia os valores de 14 ou de 0, foi necessário definir que o software assume o valor 14 ou 0 respectivamente após atingir o valor de 13.8 se a solução for acida ou de 0.3 se for base.

Para o método de cálculo a partir de equações foi feito um método de forma análoga, que está remetido para o Anexo A.

Equação para cálculo do pH a partir dos dados experimentais

```

@ActionTrigger(action="WHEN-BUTTON-PRESSED", item="TIT4_INS_BTN")
public void titinsexp_buttonClick()
{
    Tit7Adapter tit7Element=(Tit7Adapter)
this.getFormModel().getTit1().getRowAdapter(true);
    Tit4Adapter tit4Element=(Tit4Adapter)
this.getFormModel().getTit1().getRowAdapter(true);
    ArrayList<Point2D> list=new ArrayList<Point2D>();
    PointData point=
((TestFormFormController) this.getParentController()).new
PointData(null, 0);
    firstRecord();
    while(!isInLastRecord(true)){
        Point2D p =
((TestFormFormController) this.getParentController()).new
Point2D(tit4Element.getVol().toDouble(),
tit4Element.getPh().toDouble());
        list.add(p);
    }

    ((TestFormFormController) this.getParentController()).xyph(list,
point);
    tit7Element.setchartPH1(toNumber(point.notablePoint.y));
    tit4Element.setphcalcexp2(toNumber(point.notablePoint.y));
}

```

Este método baseia-se em analisar os pontos experimentais que o utilizador introduz e decompõe-nos em pontos (x,y) permitindo ao método auxiliar calcular através dos declives o ponto de inflexão.

O código para as funções auxiliares necessárias foi colocado no Anexo B, dado serem muito extensas [41] [42] [43].

4. Metodologias

5. Apresentação e Discussão dos Resultados

Neste capítulo, vai ser demonstrada a criação da base de dados e as diferenças entre um ficheiro de Excel de controle de titulação em relação ao software em Java que foi migrado e a utilização de Bases de Dados.

5.1. Criação da BD

Para proceder-se a criação a BD utilizou-se o SQL Developer Data Modeler, tendo criado um modelo relacional entre as diferentes tabelas necessárias.

Para esse efeito foi necessário definir que as tabelas necessárias, as colunas primárias e a forma de garantir a unicidade dos dados introduzidos.

Foi necessário criar seis tabelas para representar os dados, sendo que as primeiras duas da seguinte lista têm de ser inicialmente preenchidas com os dados referentes a compostos utilizados.

TITCOM- Tabela onde têm de ser criados os registos dos compostos a serem titulados com as suas características (Nome, Peso Molecular, Nome, e os pka).

TITMAR - Tabela onde têm de ser criados os registos compostos que irão servir de marcadores da titulação com as suas características (Nome, Peso Molecular, Nome, e os pka).

TITSMP-Tabela para guardar os dados que simulam uma curva de titulação a partir do volume e concentração de uma solução.

TITEQT-Tabela para guardar os dados que simulam uma curva de titulação a partir de equações da reta.

TITEXP-Tabela que guarda os dados experimentais de uma determinada experiência e traça a curva da mesma.

TITCC-Tabela que guarda os PH experimentais, introduzidos pelos utilizadores ou introduzido através do cálculo dos dados TITEXP.

De seguida está o esquema da BD utilizada na figura 33:

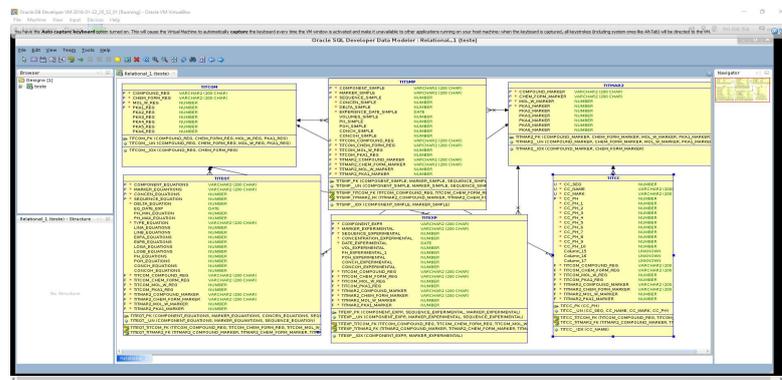


Figura 15-Eschema BD

5.2. Repositórios de Compostos

Nos Ficheiros Excel os Compostos são colocados em folhas de cálculo, podendo ser ou não divididos em tipos de compostos (Compostos a titular de Marcadores).

De seguida apresenta-se um exemplo de uma folha de compostos e marcadores numa folha de calculo, tendo-se usado o CurTiPot que estão presente na Figura 16 e 17 [46]:

Ácido ou Base	pKa = -log da constante de dissociação do ácido						Formula	Massa molecular g/mol
	pKa ₁	pKa ₂	pKa ₃	pKa ₄	pKa ₅	pKa ₆		
ORDEM ALFABÉTICA								
Acetamida	0,53						C ₂ H ₅ NO	59,067
Ácido acético	4,757						CH ₃ COOH	60,052
Ácido acetoacético	3,58						C ₄ H ₇ O ₃	102,089
Ácido acrílico	4,25						C ₃ H ₄ O ₂	72,063
Ácido asfórico	4,43	5,41					CBH ₃ NO ₄	145,143
Ácido 4-aminobenzóico	2,501	4,874					C ₇ H ₇ NO ₂	137,138
Ácido 2-aminobenzóico	2,108	4,948					C ₇ H ₇ NO ₂	137,138
Ácido 2-aminobutanoico	2,29	9,83					C ₆ H ₉ NO ₂	103,120
Ácido 8-aminohexanoico	4,373	10,504					CBH ₁₃ NO ₂	131,174
Ácido 5-aminopentanoico	4,27	10,786					CBH ₁₁ NO ₂	117,147
Ácido arsénico	2,24	6,96	11,5				HSAsO ₄	
Ácido arsenoso	9,22						HSAsO ₃	
Ácido ascórbico	4,1						CBH ₈ O ₆	
Ácido aspártico	1,99	3,9	10,002				C ₄ H ₇ NO ₄	
Ácido barbítico	4,91						C ₄ H ₄ N ₂ O ₃	128,090
Ácido benzeneossulfínico	0,7						CBH ₆ O ₂ S	
Ácido benzoico	4,19						C ₇ H ₆ O ₂	
Ácido bórico	9,236						HSBO ₃	
Ácido bromídrico	-9	12,74	13,8				HBr	
Ácido butanoico	4,83						C ₄ H ₈ O ₂	
Ácido 3-butenóico	4,34						C ₄ H ₆ O ₂	
Ácido carbónico	6,352	10,329					H ₂ CO ₃	
Ácido cianídrico	9,21						HCN	
Ácido cianídrico	3,48						HCNO	
Ácido cítrico	3,128	4,761	6,396				HSOC ₃ H ₅ O ₇	
Ácido clorídrico	-7						HCl	
Ácido cloroacético	2,865						ClCH ₂ COOH	
Ácido 2-clorobenzoico	2,92						C ₇ H ₅ ClO ₂	
Ácido 3-clorobenzoico	3,82						C ₇ H ₅ ClO ₂	
Ácido 4-clorobenzoico	3,98						C ₇ H ₅ ClO ₂	
Ácido cromoico	-1,2	6,51					H ₂ CrO ₄	
Ácido 2,4-diaminobutanoico	1,85	8,24	10,44				C ₄ H ₁₀ N ₂ O ₂	
Ácido dicloroacético	1,3						ClCH ₂ COOH	
Ácido dicloroacético	2,8						C ₂ H ₂ Cl ₂ O ₂	
Ácido dipicróico	2,16	4,76					C ₇ H ₅ N ₂ O ₄	
Ácido etilendiametetraacetato (EDTA)	-1,21	1,5	2,21	3,11	6,75	11,03	C ₁₀ H ₁₆ N ₂ O ₈	
Ácido ferriacético	4,28						CBH ₆ O ₂	
Ácido fluorídrico	3,17						HF	
Ácido fórmico	3,745						HCOOH	
Ácido fosfórico	2,148	7,199	12,35				H ₃ PO ₄	
Ácido m- fáltico	3,54	4,6					CBH ₆ O ₄	166,140
Ácido o- fáltico	3,51	4,62					CBH ₆ O ₄	166,140
Ácido p- fáltico	2,95	5,408					CBH ₆ O ₄	166,140
Ácido fumárico	3,053	4,484					C ₄ H ₄ O ₄	116,070
Ácido glicérico	3,52						CBH ₈ O ₄	
Ácido glicólico	3,831						HOCH ₂ COOH	
Ácido itálico	3,18						C ₇ H ₆ O ₄	

Figura 16-Repositório de Compostos CurTiPot

Ácido ou Base	pKa = -log da constante de dissociação do ácido						Formula	Massa molecular g/mol	pKa
	pKa ₁	pKa ₂	pKa ₃	pKa ₄	pKa ₅	pKa ₆			
ORDEM ALFABÉTICA									
Acetamida	0,53						C ₂ H ₅ NO	59,067	
Ácido acético	4,757						CH ₃ COOH	60,052	
3-Nitrofenol	8,39						CBH ₅ NO ₃		
4-Nitrofenol	7,15						CBH ₅ NO ₃		
Isoprenalina	8,64	9,7					CBH ₁₁ NO ₃		
Oxalacetamida	10,6						C ₁ BH ₃ N		
Papaverina	5,4						C ₂₀ H ₂₁ NO ₄		
Peróxido de hidrogénio	11,65						H ₂ O ₂		
Picarpina	6,87						C ₁₁ H ₁₈ N ₂ O ₂	208,259	
Piperazina	9,83	5,56					C ₄ H ₁₀ N ₂		
Piperidina	7,53	11,123					C ₅ H ₁₁ N	85,150	
Prina	5,229						CBH ₉ N	79,100	
Primidina	6,35						C ₁ H ₃ N ₃		
Procatecol	9,4	12,8					CBH ₆ O ₂	110,100	
Propidina	11,27						C ₄ H ₉ N		
Prolina	1,952	10,64					C ₅ H ₉ NO ₂	115,130	
Propilamina	10,586						CH ₃ CH ₂ CH ₂ NH ₂		
Purina	2,3	8,66					C ₅ H ₄ N ₄	120,110	
Quina	4,13	8,52					C ₂₀ H ₂₄ N ₂ O ₂	324,420	
Quinolina	4,9						CBH ₇ N	129,160	
Resorcinol	9,3	11,06					CBH ₆ O ₂	110,100	
Sacarina	11,68						C ₇ H ₅ N ₃ O ₃ S		
Serina	2,19	9,05					C ₃ H ₇ NO ₃	105,090	
Strychna	8,26						C ₂₁ H ₂₂ N ₂ O ₂	334,410	
Tiazol	2,44						C ₃ H ₃ N ₃ S		
Tiramina	9,74	10,52					CBH ₁₁ NO		
Treana	2,17	5,19	10,47				CBH ₁₁ NO ₃		
Treonina	2,088	9,1					C ₄ H ₉ NO ₃	119,120	
Trietanolamina	7,762						HOCH ₂ CH ₂ CH ₂ NH ₂		
Trietamina	10,715						CH ₃ CH ₂ CH ₂ NH ₂		
Trimetlamina	9,8						CH ₃ CH ₃ NH ₂		
Triptofano	2,35	9,54					C ₁₁ H ₁₂ N ₂ O ₂		
Tri(hidroxiometil)-aminometano = tris	8,075						HOCH ₂ CH ₂ CH ₂ NH ₂		
Ureia	0,1						CH ₄ N ₂ O		
Valina	2,286	9,718					C ₅ H ₁₁ NO ₂		
INDICADORES									
Azul de Timol (pKa 1,7)	1,7	8,9					C ₂₇ H ₃₀ O ₅ S	466,590	1,7
Alaranjado de Metila (pKa 3,5)	-1	3,47					C ₁₄ H ₁₄ N ₃ NaO ₃ S	327,330	3,47
Verde de Bromocresol (pKa 4,8)	-1	4,8					C ₂₁ H ₁₄ Br ₄ O ₃ S	690,010	4,8
Vermeho de Metila (pKa 5,1)	5,1						C ₁₅ H ₁₅ N ₃ O ₂	269,300	5,1
Azul de Bromotimol (pKa 7,1)	7,1						C ₂₇ H ₂₈ B ₂ O ₅ S	624,380	7,1
Vermeho de fenol (pKa 7,7)	1,2	7,7					C ₁₈ H ₁₄ O ₅ S	354,380	7,7
Azul de Timol (pKa 8,9)	1,7	8,9					C ₂₇ H ₃₀ O ₅ S	466,590	8,9
Fenolftaleina (pKa 9,7)	9,5	9,7					C ₂₀ H ₁₄ O ₄	318,320	9,7
Alizarina amarela R (pKa 11)	11						C ₁₃ H ₆ N ₃ O ₅ S	287,230	11

Figura 17-Repositório de Marcadores CurTiPot

5. Apresentação e Discussão de Resultados

Estas folhas de calculo sem fazer macros adicionais vão permitir a inserção de compostos duplicados, além desta situação estes dados terão de ser colocados em cada Excel o que leva a uma dispersão dos dados se não estiverem num repositório online. Isto levanta a questão de múltiplos utilizadores usarem em simultâneo e inserirem dados.

De forma a facilitar o uso do utilizador optou-se por dividir as tabelas de marcadores e compostos no software transformado tendo ficado como demonstrado na Figura 18 e 19:



Compound	Chemical Formula	Molecular Mass (g/mol)	PKA1	PKA2	PKA3	PKA4	PKA5	PKA6
CHLOROPHIL	CHLOROPHIL	39.025		7.07				
Acid Content	MG	38.481		7.0				



Compound	Chemical Formula	Molecular Mass (g/mol)	PKA1	PKA2	PKA3	PKA4	PKA5	PKA6
C29H30O2	C29H30O2	418.52		9.5	9.7			
Acid de Total	C29H30O2	418.53		9.7	8.9			

Figura 18- Repositório Compostos



Compound	Chemical Formula	Molecular Mass (g/mol)	PKA1	PKA2	PKA3	PKA4	PKA5	PKA6
C29H30O2	C29H30O2	418.52		9.5	9.7			
Acid de Total	C29H30O2	418.53		9.7	8.9			



Compound	Chemical Formula	Molecular Mass (g/mol)	PKA1	PKA2	PKA3	PKA4	PKA5	PKA6
C29H30O2	C29H30O2	418.52		9.5	9.7			
Acid de Total	C29H30O2	418.53		9.7	8.9			

Figura 19- Repositório Marcadores

Esta versão vai permitir a todos os utilizadores um acesso centralizado em que todos poderão aceder e modificar, ficando imediatamente disponível. Tem como grande vantagem a separação clara do tipo de compostos e o facto de como se utiliza o SQL Transactional Model as tabelas poderem ser editadas em simultâneo por múltiplos utilizadores.

5.3. Curva de Titulação e Determinação do PH

Com base nas tabelas anteriores e com base em valores tabelados dos diferentes compostos, aliados a dados de volumes e concentrações determina-se as Curvas de titulação e o pH de uma solução.

Simulação a Partir da Concentração e Volumes

Em versões de Excel as folhas de calculo ficam muito complexas dado necessitarem de terem todos os dados disponíveis localmente. Aliado a isto é necessário a aplicação de macros em algumas situações para se proceder as validações necessárias.

Na Figura 20 apresenta-se uma folha de calculo, tendo se usado o CurTiPot[46] para demonstrar:

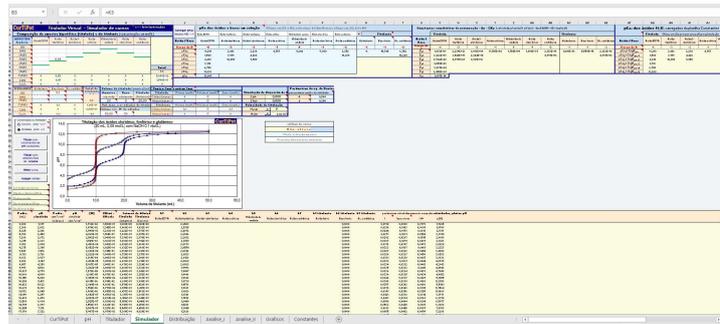


Figura 20- Simulação CurTiPot

Na versão transformada só é necessário fornecer os dados específicos para a experiencia (Volumes, Concentração, etc.) dado que os restantes já se encontram associados na BD para o composto selecionado.

Esse facto simplifica o aspeto da visualização do software como pode ser verificado na Figura 21.

5. Apresentação e Discussão de Resultados

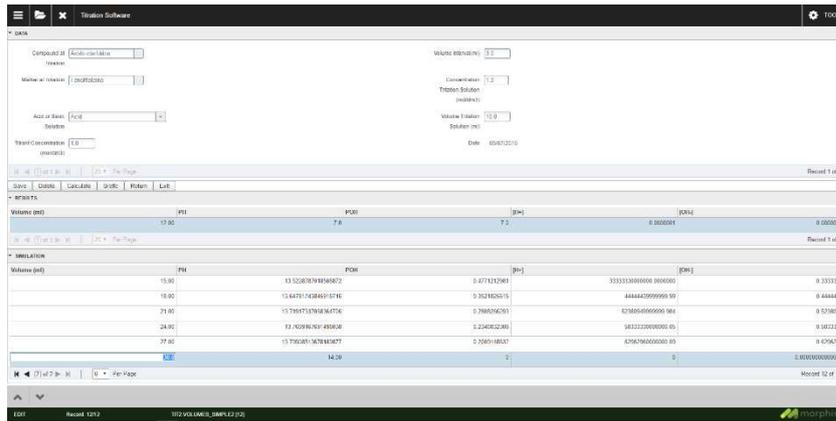


Figura 21-Cálculo Curva de Titulação e PH

A Partir dos dados anteriores obteve-se a Curva de Titulação representada na Figura 22.

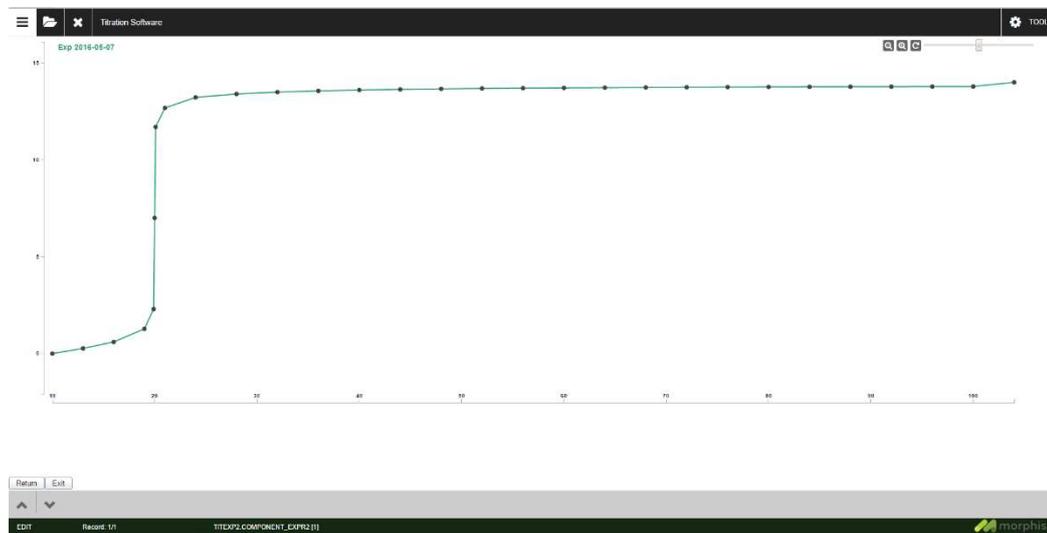


Figura 22-Curva de Titulação

Estes resultados foram validados a partir dos dados da tabela 2 [47].

Vol HCL (ml)	Vol NAOH (ml)	Vol Total (ml)	HCL (mol)	NAOH (mol)	Diferença (mol)	[H+] (g/mol)	PH	[OH-] (g/mol)	POH
10	0,0	10,0	0,01	0,0000	0,01	1	0,00	1E-14	14,00
10	3,0	13,0	0,01	0,0030	0,007	0,5385	0,27	2E-14	13,73
10	6,0	16,0	0,01	0,0060	0,004	0,25	0,60	4E-14	13,40
10	9,0	19,0	0,01	0,0090	0,001	0,0526	1,28	1,9E-13	12,72
10	9,9	19,9	0,01	0,0099	1,00E-04	0,005	2,30	1,99E-12	11,70
10	10,0	20,0	0,01	0,0100	0	1,00E-07	7,00	1E-07	7,00
10	10,1	20,1	0,01	0,0101	1,00E-04	2,00E-12	11,70	0,004975	2,30
10	11,0	21,0	0,01	0,0110	0,001	2,10E-13	12,68	0,047619	1,32
10	14,0	24,0	0,01	0,0140	0,004	6,00E-14	13,22	0,167	0,78

Tabela 4-Validação dos Resultados [47]

5. Apresentação e Discussão de Resultados

A versão transformada além de ser visualmente mais simples, também identifica diretamente de forma explícita o valor a qual o pH corresponde a 7, isto é o momento em que o numero de moles de acido é idêntico ao numero de moles de base, o volume a qual isso ocorre e as respectivas concentrações

Simulação a Partir de Equações

Esta simulação foi adicionada para demonstrar a capacidade de variabilidade que permite a um software em java ter mantendo todas as características de modelos mais simples, foi verificado neste ponto a necessidade de aumento das capacidades da Biblioteca da Framework da Morphis

Para esse efeito foi necessário alterar o aspeto visual do modelo mais simples, pedindo a introdução dos dados necessários para o calculo de 3 tipos de equações (Lineares, logarítmicas, exponenciais), o aspeto visual pode ser observado na figura 23

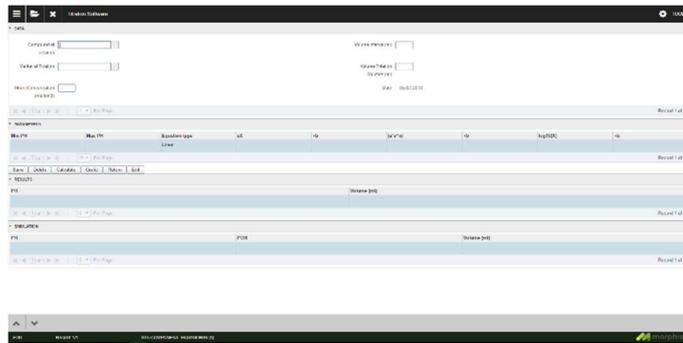


Figura 23- Calculo Curva a partir de Equações

Curva Experimental

A Curva Experimental permite ao utilizador introduzir os valores experimentais manualmente e traçar a curva, utilizando métodos para determinação de pH através do declive da curva ou de derivação.

Na figura 24 está representada a forma como no o CurTiPot [46] ficando definido esta funcionalidade

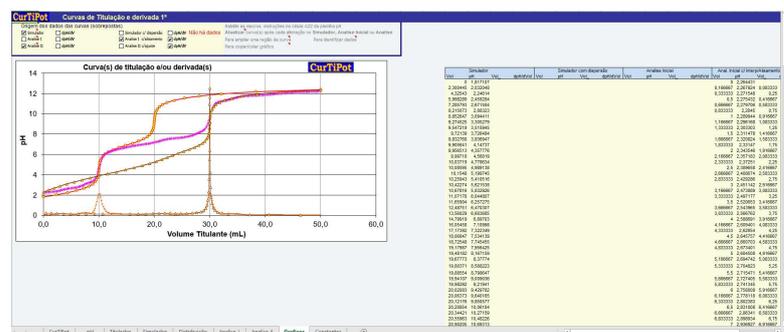
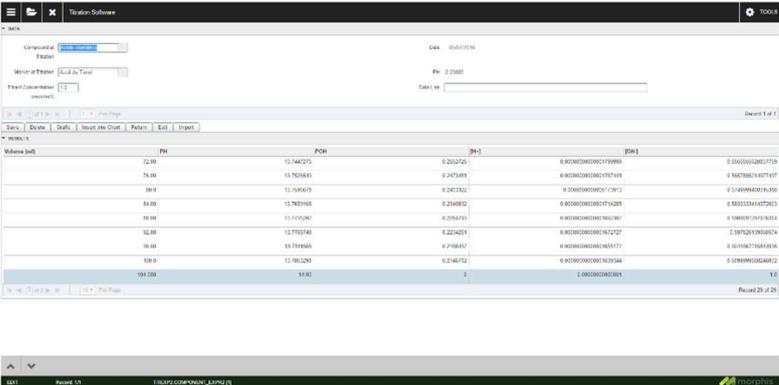


Figura 24- Gráfico CurTiPot

5. Apresentação e Discussão de Resultados

Na versão transformada o início da funcionalidade é igual, dado que é necessário introduzir os valores, o que diferencia este é a opção de o software permitir diretamente o calculo e inserção do valor obtido numa carta de controle.

Está representado a versão transformada de introdução de dados na figura 25, o gráfico não está aqui representado dado ter o mesmo aspeto visual do gráfico da simulação de curvas de titulação atrás referido.



Volume (ml)	pH	PCH	[OH ⁻]	[OH ⁻]
0.00	12.744272	0.252725	0.0000000000000000	0.0000000000000000
25.00	12.702869	0.243389	0.0000000000000000	0.0000000000000000
50.00	12.700472	0.242702	0.0000000000000000	0.0000000000000000
75.00	12.700968	0.242852	0.0000000000000000	0.0000000000000000
100.00	12.719280	0.250701	0.0000000000000000	0.0000000000000000
125.00	12.775248	0.282428	0.0000000000000000	0.0000000000000000
150.00	12.711868	0.238803	0.0000000000000000	0.0000000000000000
175.00	12.701290	0.238702	0.0000000000000000	0.0000000000000000
199.99	12.701290	0.238702	0.0000000000000000	0.0000000000000000
200.00	14.00	0	0.0000000000000000	0.0000000000000000

Figura 25-Pontos Curva Experimental

5.4. Controle de Qualidade

As cartas de controle servem para validar se um processo estatístico se encontra sob controle ou não. Dado que um dos objetivos desta tese era que fosse funcional optou-se por colocar uma forma de validar os dados experimentais obtidos, recorrendo por isso a cartas de controle [44][45].

Esta secção tem 2 formas de adicionar dados traçar a carta de controle:

1. Do método da Curva Experimental.
2. pH medio (Introduzir pH de múltiplas experiencias efetuadas para o mesmo caso de teste).

Na Figura 26 está representado visualmente o local de inserção do método de traçar as cartas de controle.

5. Apresentação e Discussão de Resultados

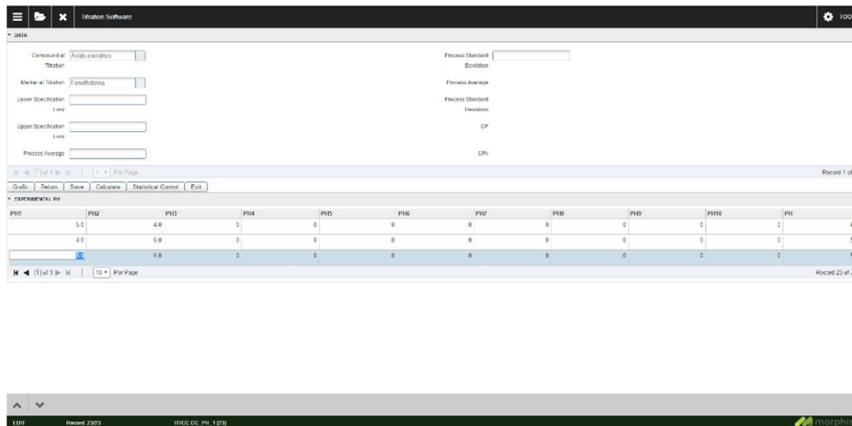


Figura 26-Inserção cartas de controle

Carta de Controle

A partir dos valores previamente inseridos e possível traçar a carta de controle que tem o seguinte aspecto que esta representado na figura 27:

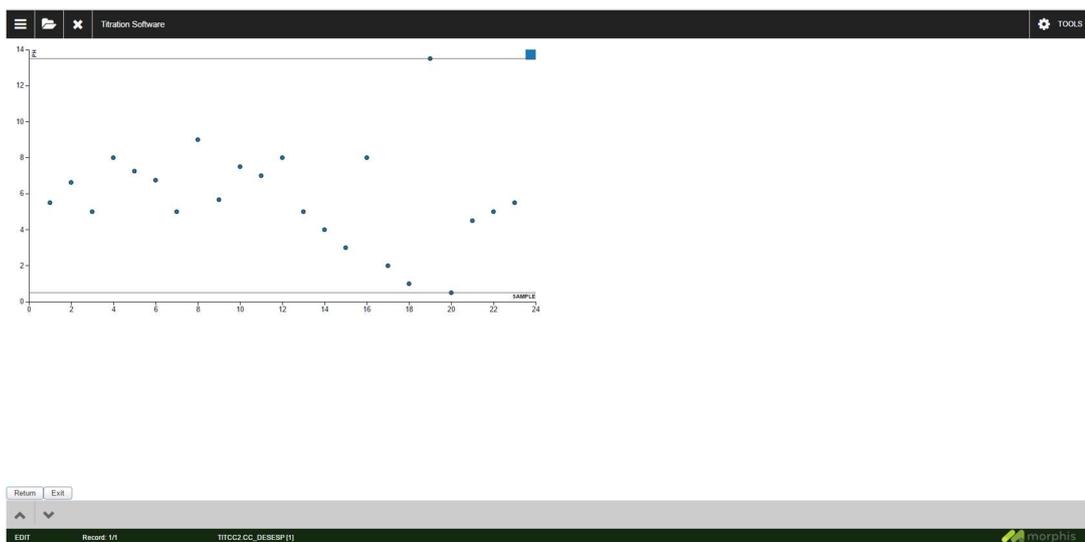


Figura 27- Carta de Controle

A partir da carta traçada vai ser possível aferir a presença de causas especiais de variação permitindo desta forma a decisão de medidas corretivas.

6. Conclusões

Esta dissertação teve como objetivo a migração de software de reações de titulação tendo em conta a conversão de Excel para Java.

Ao elaborar esta tese foi possível concluir que a framework da Morphis (Foundations + Frames) é também adequada para aplicações na área da Engenharia, nomeadamente no tratamento e visualização de dados. Verificou-se também que é necessário aumentar a capacidade da biblioteca da framework da Morphis em termos de tratamento matemático dos dados.

Em termos de migração conclui-se na elaboração da mesma que as vantagens do software em Java em relação ao Excel são:

- Acessibilidade a utilizadores invisuais recorrendo ao JAWS;
- Possibilidade de aceder em qualquer dispositivo à aplicação (Portátil, Telemóvel, Tablet);
- Não necessitar de ter ficheiros a nível do utilizador, dado ser uma aplicação web;
- Permitir que múltiplos utilizadores editem em simultâneo os dados sem ocorrência de conflitos.

Em relação às desvantagens do software em Java em relação ao Excel são:

- O custo do servidor para incorporar a base de dados.

Também foi possível concluir, que as folhas de calculo por vezes tornam o processo mais complicado no entendimento por parte do utilizador, sendo necessário procurar o local onde estão os dados, como utiliza-los e a forma como se encontram distribuídos.

Os novos softwares são visualmente são mais simples, dado que a organização pode ser efetuada em blocos distintos simplificando a visão da aplicação a nível de utilizador final. Isso permite ao utilizador a consulta de dados, efetuar os cálculos de uma forma intuitiva, sem ter grande conhecimento de como o software funciona. A versão transformada também identifica diretamente de forma explicita o valor a qual o PH corresponde a 7, isto é o momento em que o numero de moles de acido é idêntico ao numero de moles de base, o volume a qual isso ocorre e as respetivas concentrações quando se simula.

7. Trabalho Futuro

Como trabalho futuro seria interessante proceder-se a migração de outras funcionalidades além da titulação de forma a fazer-se uma aplicação funcional web completa para poder ser integrada em laboratórios, dado que:

- Permite uma utilização de todos os utilizadores, na medida em que esta tecnologia permite a utilização do software JAWS sem restrições, facilitando o acesso a utilizadores com limitações visuais.
- Permite uma melhor gestão de dados, dado que possibilita múltiplos utilizadores a fazer alterações em simultâneo.

Relativamente à funcionalidade da titulação seria interessante num futuro trabalho, implementar métodos para mais tipos de equações de forma a permitir ao utilizador uma maior variedade de opções.

Relativamente a framework da Morphis seria útil no futuro utilizar as ferramentas da mesma para possibilitar o tratamento dos dados através de vários tipos de gráficos que permitam a manipulação dos dados por parte dos utilizadores, além de interligar outras funcionalidades que permitam interpretação simultânea dos resultados. No decorrer desta implementação para novo tratamentod e graficos seria util dotar a fremwork da morphis das capacidades necessarias a nivel de calculos para a area de Engenharia.

7. Trabalho Futuro

8. Bibliografia

[1] “ Dow Chemical Paper ”, [Online], acessado 16 de Março de 2016, disponível em:

http://www.chemicalprocessing.com/assets/Media/MediaManager/wp_06_032_wbf_benton.pdf

[2] “ Morphis ”, [Online], acessado 16 de Março de 2016, disponível em:

<https://morphis-tech.com/>

[3] “Historia Linguagens Programação”, [Online], acessado 16 de Março de 2016, disponível em:

https://pt.wikipedia.org/wiki/Hist%C3%B3ria_das_linguagens_de_programa%C3%A7%C3%A3o

[4] “Historia Linguagens Programação”, [Online], acessado 16 de Março de 2016, disponível em:

http://www.dsc.ufcg.edu.br/~pet/jornal/novembro2012/materias/historia_da_computacao.html

[5] I. Din, Akeel (1994) “Structured Query Language - A Practical Introduction”

[6] Preston Prescott (2015)- SQL for Beginners: Learn the Structured Query Language for the Most Popular Databases including Microsoft SQL Server, MySQL, MariaDB, PostgreSQL, and Oracle Paperback

[7] “ Programação Procedimental “, [Online], acessado 16 de Março de 2016, disponível em:

https://pt.wikipedia.org/wiki/Programa%C3%A7%C3%A3o_procedural

[8] “ PL/SQL “, [Online], acessado 16 de Março de 2016, disponível em:

<https://pt.wikipedia.org/wiki/PL/SQL>

[9] Feuerstein Steven, Pribyl, Bill (2005) “Oracle PL/SQL Programming, 4th Edition”

[10] Feuerstein Steven (2007) “Oracle PL/SQL Best Practices, 2nd Edition”

8. Bibliografia

[11] “ Java “, [Online], acessado 16 de Março de 2016, disponível em:

https://pt.wikipedia.org/wiki/Java_%28linguagem_de_programa%C3%A7%C3%A3o%29

[12] “ Java “, [Online], acessado 16 de Março de 2016, disponível em:

<http://www.programacaoprogessiva.net/2012/08/comece-programar-linguagem-de.html>

[13] Eckel, Bruce “Thinking in Java”, Fourth Edition, [Online], acessado 16 de Março de 2016, disponível em:

http://sd.blackball.lv/library/Thinking_in_Java_4th_edition.pdf

[14] “ ByteCode “, [Online], acessado 16 de Março de 2016, disponível em:

https://pt.wikipedia.org/wiki/Bytecode_Java

[15] “ ByteCode ” Engel ,Joshua -Programming for the Java Virtual Machine

[Online], acessado 20 de Março de 2016, disponível em:

https://books.google.pt/books?id=dg7bj_e-SAMC&pg=PA68&source=gbs_selected_pages&cad=2#v=onepage&q=bytecode&f=false

[16] “ Linguagem de Marcação “, [Online], acessado 16 de Março de 2016, disponível em:

https://pt.wikipedia.org/wiki/Linguagem_de_marca%C3%A7%C3%A3o

[17] Key, Sam (2015) XML Programming Success in a Day: Beginner's Guide to Fast, Easy, and Efficient Learning of XML Programming (XML, XML Programming, Programming, XML Guide, ... XSL, DTD's, Schemas, HTML5, JavaScript)

[18] Fawcett, Joe, Ayers, Liam, Quin, R.E.(2012)- “Beginning XML, 5th Edition”

[19] “ SQL Developer “, [Online], acessado 16 de Março de 2016, disponível em:

<http://www.oracle.com/technetwork/developer-tools/sql-developer/what-is-sqldev-093866.html>

[20] “ SQL Developer Data Modeler “, [Online], acessado 16 de Março de 2016, disponível em:

<http://www.oracle.com/technetwork/developer-tools/datamodeler/overview/index.html>

8. Bibliografia

[21] “ GIT “, [Online], acessado 16 de Março de 2016, disponível em:

https://en.wikipedia.org/wiki/Git_%28software%29

[22] “ SMARTGIT “, [Online], acessado 16 de Março de 2016, disponível em:

<http://www.syntevo.com/smartgit/>

[23] “ SourceTree “, [Online], acessado 16 de Março de 2016, disponível em:

<https://www.sourcetreeapp.com/>

[24] “ Eclipse “, [Online], acessado 16 de Março de 2016, disponível em:

<https://eclipse.org/org/>

[25] “ IDE “, [Online], acessado 16 de Março de 2016, disponível em:

https://pt.wikipedia.org/wiki/Ambiente_de_desenvolvimento_integrado

[26] “ IDE “, [Online], acessado 16 de Março de 2016, disponível em:

https://pt.wikipedia.org/wiki/Apache_Tomcat

[27]- Vogel,Lars (2013) “Eclipse IDE: Eclipse IDE based on Eclipse 4.2 and 4.3”

[28] Abílio, Ramon Simões () “PROPOSTA DE UMA METODOLOGIA DE MIGRAÇÃO DE SOFTWARE PROPRIETÁRIO PARA SOFTWARE LIVRE COM FOCO NO USUÁRIO”
.[Online], acessado 16 de Março de 2016, disponível em:

http://repositorio.ufla.br/bitstream/1/5422/1/MONOGRAFIA_Proposta_de_uma_metodologia_de_migra%C3%A7%C3%A3o_de_software_propriet%C3%A1rio_para_software_livre_com_foco_no_usu%C3%A1rio.pdf

[29] “ Tomcat “, [Online], acessado 16 de Março de 2016, disponível em:

https://pt.wikipedia.org/wiki/Apache_Tomcat

[30] “ Titulação “, [Online], acessado 16 de Março de 2016, disponível em:

<http://mundoeducacao.bol.uol.com.br/quimica/conceito-ph-poh.htm>

[31] L. Good, Daniel, R. Ball, Scoot, W. Reger, David(2010) “Chemistry Principles and Practice-3rd Edition”

- [32] Chang, R., Química, 8ªed., McGraw-Hill, Lisboa, 2005
- [33] “ BO “, [Online], acessido 16 de Março de 2016, disponível em:
https://en.wikipedia.org/wiki/Business_object
- [34] Howson, Cindi “BusinessObjects XI (Release 2): The Complete Reference”
- [35] “ Business Logic “, [Online], acessido 16 de Março de 2016, disponível em:
https://en.wikipedia.org/wiki/Business_logic
- [36] “ MVC “, [Online], acessido 16 de Março de 2016, disponível em:
<http://www.oracle.com/technetwork/articles/javase/index-142890.html>
- [37] “ MVC “, [Online], acessido 16 de Março de 2016, disponível em:
<https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>
- [38] “ MVC “, [Online], acessido 16 de Março de 2016, disponível em:
<https://msdn.microsoft.com/en-us/library/ff649643.aspx>
- [39] “ SQL- Transactions “, [Online], acessido 16 de Março de 2016, disponível em:
<http://www.tutorialspoint.com/sql/sql-transactions.htm>
- [40] “ SQL- Transactions “, Sheldon, Robert, Richardson, Rob, and Davis, Tony “SQL Server Source Control Basics” [Online],
- [41] “ Método de Calculo pH através pontos inflexão “, [Online], acessido 16 de Março de 2016, disponível em:
<http://www.shodor.org/~jmorrell/interactivate/org/shodor/util11/DataSetUtils.java>
- [42] “ Método de Calculo pH através pontos inflexão “, [Online], acessido 16 de Março de 2016, disponível em:
<http://www.shodor.org/~jmorrell/interactivate/org/shodor/data/Point2D.java>
- [43] “ Método de Calculo pH através pontos inflexão “, [Online], acessido 16 de Março de 2016, disponível em:
<http://www.shodor.org/~jmorrell/interactivate/org/shodor/data/PointData.java>
- [44] Pereira, Z.L. and Requeijo, J.G. (2008), Qualidade: Planeamento e Controlo Estatístico de Processos, FCT-UNL e Prefácio, Lisboa

[45] “ ISO 8258(1991) “ [Online], acessado 20 de Março de 2016, disponível em:

https://pt.wikipedia.org/wiki/Carta_de_controlo

[46] “ CurTiPot “ Rolf Gutz ,Ivano Gebhardt Instituto de Química da Universidade de São Paulo [Online], acessado 16 de Março de 2016, disponível em:

<http://www.iq.usp.br/gutz/Curtipot.html>

[47] “Validação de Resultado“ [Online], Apontamentos de Química 1- Reações ácido-base, pagina 19, acessado 22 de Março de 2016, disponível em:

<http://www2.dq.fct.unl.pt/cadeiras/quimica1/material/teoria/livroQ1-2-pH.pdf>

[48], M.A. Jackson (1975)-“Principles of Program Design 1st Edition”

[49] McLaughlin ,Brett , “Building Java Enterprise Applications”, [Online], acessado 16 de Março de 2016, disponível em:

<ftp://ftp.bupt.edu.cn/pub/Documents/Programming/Java-Related/O'Reilly%20-%20Building%20Java%20EA%20vol%20I%20Architecture.pdf>

[50] Hale, Dave, “The Java and C++ platforms for scientific computing”, Center for wave Phenomena, Colorado School of Mines, Golden CO 80401, USA, [Online], acessado 16 de Março de 2016, disponível em:

<http://www.cwp.mines.edu/Documents/cwpreports/cwp547.pdf>

[51] Henk J. Sips and Kees van Reeuwijk, “Java for Scientific Computation: Prospects and Problems”, Delft University of Technology, the Netherland, [Online], acessado 16 de Março de 2016, disponível em:

<http://www.ds.ewi.tudelft.nl/pubs/papers/scicomp01.pdf>

[52] Villacis, Juan, “A Note on the Use of Java in Scientific Computing”, Indiana University, Bloomington IN 47405, Computer Science Department. [Online], acessado 16 de Março de 2016, disponível em:

<https://www.extreme.indiana.edu/cat/papers/sigapps.pdf>

[53] CEP - Controle Estatístico de Processo

<http://www.datalyzer.com.br/site/suporte/administrador/docs/arquivos/doc57/57.html>

Anexos

Anexo A – Cálculo Curva Titulação a partir de Equações

```
@ActionTrigger(action="WHEN-BUTTON-PRESSED",
item="TIT3_CALC_BTN")
public void titcalceq_buttonClick()
{
    //tenho de alterar o metodo para ter um bloco virtual para
    calcular e na bd so gravar o ph

    TITEQTAdapter tit3Element=(TITEQTAdapter)
this.getFormModel().getTITEQT().getRowAdapter(true);
    TIT3Adapter tit4Element=(TIT3Adapter)
this.getFormModel().getTIT3().getRowAdapter(true);

    if(tit3Element.getcomepeq().isNull()||tit3Element.getintvoleq().i
sNull()){

        errorMessage("Missing Data in Required Fields:
Compound Name, Interval Volume and Titration Concentracion");
        throw new ApplicationException();
    }

    if(tit3Element.gettypepeq().isNull()){
        errorMessage("Please define the type of equation you
desire to work with");
        throw new ApplicationException();
    }

    if(tit3Element.gettypepeq().equals("OUT")){
        errorMessage("This software its not available for
your simulation at the moment");
        throw new ApplicationException();
    }

    if(tit3Element.getexpaeq().isNull() &&
tit3Element.getexpbeq().isNull() && tit3Element.getlinaeq().isNull() &&
tit3Element.gelinbeq().isNull() &&
tit3Element.getlogaeq().isNull() && tit3Element.getlobeq().isNull()){

        errorMessage("Please define the parameters for your
equation");
        throw new ApplicationException();
    }

    NNumber phmax=NNumber.getNull();
    NNumber phmin=NNumber.getNull();
    NBool flag=NBool.False;

    while(!isInLastRecord(true)){
        if(phmax.isNull())
            tit3Element.getphmaxeq();

        if(phmax.greater(tit3Element.getphmaxeq())){
            tit3Element.getphmaxeq();
            flag=NBool.True;
        }
    }
}
```

```

        if(phmin.isNull())
            tit3Element.getphmineq();

        if(phmin.lesser(tit3Element.getphmineq())){
            tit3Element.getphmineq();
            flag=NBool.True;
        }

        nextRecord();
    }
    firstRecord();

    if(flag.getValue()){
        errorMessage("Please insert the interval in PH order
from PH=0 to PH=14 to proceed");
        throw new ApplicationException();
    }

    if(phmax.lesser(14)|| phmin.greater(0)){
        NNumber alertBtn = toNumber(showAlert("PHEQ"));

        if(alertBtn.equals(MessageServices.BUTTON1)){
        }
        else{
            return;
        }
    }

    }
    NNumber vol=new NNumber(0);
    BigDecimal ctitulante= new
BigDecimal(toStr(tit3Element.getconconsoleq()).getValue());
    NNumber phi=NNumber.getNull();
    goItem("TIT3.PH_EQUATIONS3");
    while(true){

        if(tit3Element.gettypeeq().equals("LIN")){
            NNumber phmi=tit3Element.getphmineq();
            NNumber phma=tit3Element.getphmaxeq();
            NNumber moles= NNumber.getNull();

            while(phmi.lesserOrEquals(phma)){

                if(tit3Element.getlinaeq().equals(toNumber(0))&&
tit3Element.gelinbeq().equals(0)|| tit3Element.gelinbeq().equals(0) ){
                    errorMessage("The Parameters cannot
be 0");
                    throw new ApplicationException();
                }

                if((tit3Element.getlinaeq().isNotNull()
|| tit3Element.getlinaeq().notEquals(0) )&&
(tit3Element.gelinbeq().isNotNull()
&&tit3Element.gelinbeq().notEquals(0) ) ){

```

```

        tit4Element.setconchcalceq3(trunc(toNumber(phmi.subtract(tit3Element.getlinbeq()))).divide(tit3Element.getlinaeq()),7));

        tit4Element.setconcohalceq3(trunc(toNumber(toNumber(1).subtract(tit4Element.getconchcalceq3())),7));
        tit4Element.setphcalceq3(phmi);

        tit4Element.setpohcalceq3(toNumber("14").subtract(phmi));
    }
    else{

        tit4Element.setconchcalceq3(trunc(toNumber(phmi.divide(tit3Element.getlinaeq()))),7));

        tit4Element.setconcohalceq3(trunc(toNumber(toNumber(1).subtract(tit4Element.getconchcalceq3())),7));
        tit4Element.setphcalceq3(phmi);

        tit4Element.setpohcalceq3(toNumber("14").subtract(phmi));
    }

    if(phmi.greaterOrEquals(14)){

        tit4Element.setconchcalceq3(toNumber(0));

        tit4Element.setconcohalceq3(toNumber(1));

        tit4Element.setphcalceq3(toNumber(14));

        tit4Element.setpohcalceq3(toNumber("0"));
        break;
    }
    else{
        phmi=phma.add(0.1);
        duplicateRecord();
    }
}

if(tit3Element.gettypeeq().equals("EXP")){
    NNumber phmi=tit3Element.getphmineq();
    NNumber phma=tit3Element.getphmaxeq();
    NNumber moles= NNumber.getNull();

    while(phmi.lesserOrEquals(phma)){

        if(tit3Element.getexpaeq().equals(toNumber(0))&&
tit3Element.getexpbeq().equals(0)||
tit3Element.getexpaeq().equals(0)){
            errorMessage("The Parameters cannot
be 0");

            throw new ApplicationException();
        }

        if((tit3Element.getexpaeq().isNotNull()
|| tit3Element.getexpaeq().notEquals(0) )&&
(tit3Element.getexpbeq().isNotNull() &&
tit3Element.getexpbeq().notEquals(0))){

```

```

                Double aux=
(phmi.subtract(tit3Element.getexpbeq()).divide(tit3Element.getexpaeq(
)).toDouble());

        tit4Element.setconchcalceq3(toNumber(Math.log10(aux)));

        tit4Element.setconcohalceq3(toNumber(1).subtract(tit4Element.ge
tconchcalceq3()));
                tit4Element.setphcalceq3(phmi);

        tit4Element.setpohcalceq3(toNumber("14").subtract(phmi));
                }
                else{
                Double aux=
(phmi).divide(tit3Element.getexpaeq()).toDouble();

        tit4Element.setconchcalceq3(toNumber(Math.log10(aux)));

        tit4Element.setconcohalceq3(toNumber(1).subtract(tit4Element.ge
tconchcalceq3()));
                tit4Element.setphcalceq3(phmi);

        tit4Element.setpohcalceq3(toNumber("14").subtract(phmi));
                }

                if(phmi.greaterOrEquals(14)){

        tit4Element.setconchcalceq3(toNumber(0));

        tit4Element.setconcohalceq3(toNumber(1));

        tit4Element.setphcalceq3(toNumber(14));

        tit4Element.setpohcalceq3(toNumber("0"));
                break;
                }
                else{
        phmi=phma.add(0.1);
        duplicateRecord();
                }
        }
        if(tit3Element.gettypeeq().equals("LOG")){

        NNumber phmi=tit3Element.getphmineq();
        NNumber phma=tit3Element.getphmaxeq();
        NNumber moles= NNumber.getNull();

        while(phmi.lesserOrEquals(phma)){

                if(tit3Element.getlogaeq().equals(toNumber(0)) &&
tit3Element.getlobeq().equals(0) || tit3Element.getlogaeq().equals(0)){
                errorMessage("The Parameters cannot
be 0");

                throw new ApplicationException();
                }

```

```

        if((tit3Element.getLogaeq().isNotNull()
|| tit3Element.getLogaeq().notEquals(0) )&&
(tit3Element.getlobeq().isNotNull() &&
tit3Element.getlobeq().notEquals(0))){
            Double aux=
(phmi.subtract(tit3Element.getlobeq())).divide(tit3Element.getLogaeq()
).toDouble();

            tit4Element.setconchcalceq3(toNumber(Math.exp(aux)));

            tit4Element.setconcohcacalceq3(toNumber(1).subtract(tit4Element.ge
tconchcalceq3()));

            tit4Element.setphcalceq3(phmi);

            tit4Element.setpohcalceq3(toNumber("14").subtract(phmi));

        }
        else{
            Double aux=
(phmi).divide(tit3Element.getLogaeq()).toDouble();

            tit4Element.setconchcalceq3(toNumber(Math.exp(aux)));

            tit4Element.setconcohcacalceq3(toNumber(1).subtract(tit4Element.ge
tconchcalceq3()));

            tit4Element.setphcalceq3(phmi);

            tit4Element.setpohcalceq3(toNumber("14").subtract(phmi));

        }

        if(phmi.greaterOrEquals(14)){

            tit4Element.setconchcalceq3(toNumber(0));

            tit4Element.setconcohcacalceq3(toNumber(1));

            tit4Element.setphcalceq3(toNumber(14));

            tit4Element.setpohcalceq3(toNumber("0"));

            break;

        }
        else{
            phmi=phma.add(0.1);
            duplicateRecord();
        }
    }

}

if(isInLastRecord()){
    break;
}

}

detPh();
}

```

Anexo B –Função Auxiliar Cálculo PH a partir de Equações

```
private void detPh () {  
  
    TIT3Adapter tit4Element=(TIT3Adapter)  
this.getFormModel().getTIT3().getRowAdapter(true);  
    TITEQTAdapter tit3Element=(TITEQTAdapter)  
this.getFormModel().getTITEQT().getRowAdapter(true);  
  
    NNumber ph1=NNumber.getNull();  
    NNumber ph2=NNumber.getNull();  
    NNumber aux1=NNumber.getNull();  
    NNumber aux2=NNumber.getNull();  
    NNumber aux3=NNumber.getNull();  
    NNumber aux4=NNumber.getNull();  
    firstRecord();  
    while(!isInLastRecord()) {  
        ph1=tit4Element.getphcalceq3();  
  
        if(ph1.equals(toNumber(7))) {  
  
tit3Element.setconchcalceq(tit4Element.getconchcalceq3());  
tit3Element.setconcohalceq(tit4Element.getconcohalceq3());  
tit3Element.setphcalceq(tit4Element.getphcalceq3());  
tit3Element.setpohcalceq(tit4Element.getpohcalceq3());  
            break;  
        }  
        if(ph2.isNull()|| ph2.equals(ph1)) {  
            nextRecord();  
        }  
        ph2=tit4Element.getphcalceq3();  
  
        if (ph2.equals(toNumber(7))) {  
  
tit3Element.setconchcalceq(tit4Element.getconchcalceq3());  
tit3Element.setconcohalceq(tit4Element.getconcohalceq3());  
tit3Element.setphcalceq(tit4Element.getphcalceq3());  
tit3Element.setpohcalceq(tit4Element.getpohcalceq3());  
            break;  
        }  
        if(ph2.greater(toNumber(7))&&  
ph1.lesser(toNumber(7))) {  
            tit3Element.setphcalceq(toNumber(7));  
  
            tit3Element.setpohcalceq(toNumber(7).multiply(tit4Element.getphcalceq3().divide(ph2)));  
  
            tit3Element.setconchcalceq(toNumber(7).multiply(tit4Element.getpohcalceq3().divide(ph2)));  
  
            break;  
        }  
    }  
}
```

Anexo C– Métodos para o Cálculo do pH através da determinação do Ponto de inflexão

```
public class Point2D implements Serializable {
    /**
     *
     */
    private static final long serialVersionUID =
5148711533404696702L;
    public double x, y;

    public Point2D() {
        x = 0.0;
        y = 0.0;
    }

    public Point2D(double x, double y) {
        this.x = x;
        this.y = y;
    }

    public class PointData implements Serializable{
        /**
         *
         */
        private static final long serialVersionUID = -
8602441356443935556L;

        /**
         * The type of point represented in the data set:
local min, local max,
         * inflection point, or end point.
         */
        protected int pointType;

        /**
         * The logical coordinates of the special point.
         */
        protected Point2D notablePoint;

        public PointData (Point2D pt, int ptType) {
            notablePoint = pt;
            pointType = ptType;
        }

        /**
         * @return the notablePoint
         */
        public Point2D getNotablePoint() {
            return notablePoint;
        }

        /**
         * @param notablePoint the notablePoint to set
         */
        public void setNotablePoint(Point2D notablePoint) {
            this.notablePoint = notablePoint;
        }
    }
}
```

```

    /**
     * @return the pointType
     */
    public int getPointType() {
        return pointType;
    }

    /**
     * @param pointType the pointType to set
     */
    public void setPointType(int pointType) {
        this.pointType = pointType;
    }

    // public String toString() {
    //     return "Found point "+notablePoint.x+ ",
    // "+notablePoint.y+" as an "+toReadableType()+".";
    // }
    //
    // public String toReadableType() {
    //     switch (pointType) {
    //         case TestF.LOCAL_MIN : return "local minimum";
    //         case DataSetUtils.LOCAL_MAX : return "local
    maximum";
    //         case DataSetUtils.INFLECTION_POINT : return
    "inflection point";
    //         case DataSetUtils.END_POINT : return "end
    point";
    //         default: return "not specified";
    //     }
    // }

}

public class DataSetUtils {

    // Directions to search
    public static final int SEARCH_LEFT = 0;

    public static final int SEARCH_RIGHT = 1;

    // Point types to search for
    public static final int LOCAL_MIN = 0;

    public static final int LOCAL_MAX = 1;

    public static final int INFLECTION_POINT = 2;

    public static final int END_POINT = 3;

    public static final int ANY = 4;

    // Utility to extract one-dimensional data from a
    two-dimensional point
    // array.
    public static final int X = 0;

    public static final int Y = 1;

```

```

        public Point2D[]
findAllMinMaxInflectionPoints(Point2D[] dataSet) {

        Vector controlPts = new Vector();
        controlPts.addElement(dataSet[0]); // Always
add the first endpoint.

        int curIndex = 0;
        while (curIndex < dataSet.length - 1) {
            Point2D nextPoint =
findNextInflectionOrLocalMinMax(dataSet,
curIndex, SEARCH_RIGHT,
ANY).getNotablePoint();
            controlPts.addElement(nextPoint);

// What index of the data set corresponds to the next control point?

            int oldIndex = curIndex;
            for (int i = curIndex; i <
dataSet.length; i++) {
                Point2D pt = dataSet[i];
                if (Math.abs(pt.x - nextPoint.x) <
1E-4
                    && Math.abs(pt.y -
nextPoint.y) < 1E-4) {
                    curIndex = i;
                    break;
                }
            }
            if (curIndex == oldIndex) {
                System.out
                    .println("DEV ERROR:
DataSetUtils.findAllMinMaxInflectionPoints(Point2D[]). Error fetching
next min/max/inflection/edge.");
                /* System.out
                    .println("Could not
locate next min/max/inflection/edge after point "
curIndex);
                    +
                    System.out.println("Data Set");
                    for (int i = 0; i < dataSet.length;
i++) {
                        System.out.println("Point " +
i + ": " + dataSet[i].x
                            + ", " +
dataSet[i].y);
                    }
                    */
                break;
            }
        }

// Make sure that the last point is included in
the data set.
        Point2D lastControlPoint =
(Point2D)controlPts.lastElement();
        Point2D lastPoint = dataSet[dataSet.length-1];
        if ((Math.abs(lastPoint.x-lastControlPoint.x) >
1E-6) ||

```

```

        (Math.abs(lastPoint.y-
lastControlPoint.y) > 1E-6)) {
            controlPts.addElement(new
Point2D(lastPoint.x, lastPoint.y));
        }

        Point2D[] retValue = new
Point2D[controlPts.size()];
        for (int i = 0; i < retValue.length; i++) {
            retValue[i] = (Point2D)
controlPts.elementAt(i);
        }
        return retValue;
    }

    /**
     * Given a data set and a starting index of the data
     set, finds the nearest
     * local limimim, local maximim, inflection point, or
     the edge of the data
     * set if none of the previous three exist.
     *
     * @param dataSet
     *         The data set to search.
     * @param startingIndex
     *         The index of the data set to begin
     searching from.
     * @param direction
     *         Direction to search. Valid arguments:
     DataSetUtils.SEARCH_LEFT
     *         and DataSetUtils.SEARCH_RIGHT.
     * @param pointType
     *         Type of point to search for. Valid
     arguments:
     *         DataSetUtils.LOCAL_MIN,
     DataSetUtils.LOCAL_MAX,
     *         DataSetUtils.INFLECTION_POINT, or
     DataSetUtils.ANY.
     * @return The point to search for and the type of
     point it represents
     *         encapsulated in an
     org.shodor.data.PointData object. If a local
     *         min, max, or inflection point cannot be
     found, the type of point
     *         returned is designated
     DataSetUtils.END_POINT since it represents
     *         the edge of the data set.
     * @throws java.lang.IllegalArgumentException
     */
    public PointData
findNextInflectionOrLocalMinMax(Point2D[] dataSet,
                                int startingIndex, int direction, int
pointType)
                                throws java.lang.IllegalArgumentException
    {
        if (direction != SEARCH_LEFT && direction !=
SEARCH_RIGHT) {
            throw new
java.lang.IllegalArgumentException(

                "DataSetUtils.findNextInflectionOrLocalMinMax() improper usage.

```

```

Direction must be DataSetUtils.SEARCH_LEFT or
DataSetUtils.SEARCH_RIGHT");
    }
    if (pointType != LOCAL_MIN && pointType !=
LOCAL_MAX
&& pointType != INFLECTION_POINT &&
pointType != ANY) {
        throw new
java.lang.IllegalArgumentException(
    "DataSetUtils.findNextInflectionOrLocalMinMax() improper usage.
pointType must be DataSetUtils.LOCAL_MIN, DataSetUtils.LOCAL_MAX,
DataSetUtils.INFLECTION_POINT, or DataSetUtils.ANY.");
    }
    if (startingIndex < 0 || startingIndex >=
dataSet.length) {
        throw new IllegalArgumentException(
    "DataSetUtils.findNextInflectionOrLocalMinMax() improper usage.
Your starting index must be an index of the data set. startingIndex =
"
+ startingIndex
+ ",
dataSetLength = "
+
dataSet.length);
    }
    Point2D[] subset = new Point2D[0];
    if (direction == SEARCH_LEFT) {
        if (startingIndex <= 1) { // They are
searching left from the left
point.
            return new PointData(dataSet[0],
END_POINT);
        }
        /*
        * If starting index is 2, then we want a
2-element array and fill
        * with points 0 and 1.
        */
        subset = new Point2D[startingIndex];
        for (int i = 0; i < subset.length; i++) {
            subset[i] = dataSet[i];
        }
    }
    if (direction == SEARCH_RIGHT) {
        if (startingIndex >= dataSet.length - 2)
{ // They are searching
            // right from the right
            // most point or next to
            // right most point.
            // Return the right end
            // point.
            return new
PointData(dataSet[dataSet.length - 1], END_POINT);
        }
        /*
        * If starting index is 2 and the length
of the global data set is

```

```

fill with points 3 and 4.
2 - 1 = 2.
startingIndex - 1];
subset with dataSet
0 and 1 so we add an
subset.
startingIndex + 1];
}
}
return search(subset, direction, pointType);
}
protected PointData search(Point2D[] subset, int
direction,
int pointType) {
// To check for local min, max, inflection,
three points are needed.
int curIndex = (direction == SEARCH_LEFT) ?
subset.length - 1 : 0;
int nextIndex = (direction == SEARCH_LEFT) ?
subset.length - 2 : 1;
int followingIndex = (direction == SEARCH_LEFT)
? subset.length - 3 : 2;
if (direction == SEARCH_LEFT && followingIndex
<= 0) {
return new PointData(subset[0],
END_POINT);
}
if (direction == SEARCH_RIGHT && followingIndex
>= subset.length - 1) {
return new PointData(subset[subset.length
- 1], END_POINT);
}
int indexStep = (direction == SEARCH_RIGHT) ? 1
: -1;
while (followingIndex >= 0 && followingIndex <=
subset.length - 1) {
// Search for the min, max, or
inflection.
int nextType = (direction == SEARCH_LEFT)
? checkMinMax(
subset[nextIndex], subset[curIndex])
:
checkMinMax(subset[curIndex], subset[nextIndex],

```

```

subset[followingIndex]);
    if (nextType == LOCAL_MIN
        && (pointType == LOCAL_MIN ||
pointType == ANY)) {
        /*
        * Return the "middle" of the three
        * average closest to the true min,
        *
        */
        return new
PointData(subset[nextIndex], LOCAL_MIN);
    }
    if (nextType == LOCAL_MAX
        && (pointType == LOCAL_MAX ||
pointType == ANY)) {
        /*
        * Return the "middle" of the three
        * average closest to the true min,
        *
        */
        return new
PointData(subset[nextIndex], LOCAL_MAX);
    }
    nextType = -1;
    if (pointType == INFLECTION_POINT ||
pointType == ANY) {
        // We need 5 data points to test
        for an inflection point.
        if (direction == SEARCH_RIGHT
            && ((followingIndex +
1) < subset.length)
            && ((curIndex - 1) >=
0)) {
                nextType =
checkInflectionPoint(subset[curIndex - 1],
subset[nextIndex],
subset[followingIndex], subset[followingIndex + 1]);
            }
            // We need 5 data points to test
            for an inflection point.
            if (direction == SEARCH_LEFT &&
((followingIndex - 1) >= 0)
                && ((curIndex + 1) <
subset.length)) {
                    nextType =
checkInflectionPoint(subset[followingIndex - 1],
subset[followingIndex], subset[nextIndex],
subset[curIndex],
subset[curIndex + 1]);
                }
            }
        if (nextType == INFLECTION_POINT
            && (pointType ==
INFLECTION_POINT || pointType == ANY)) {
                /*

```

```

* Return the "middle" of the three
points which should be on
max, or inflection point.
*/
return new
PointData(subset[nextIndex], INFLECTION_POINT);
}
curIndex += indexStep;
nextIndex += indexStep;
followingIndex += indexStep;
}
// If we get here, we did not find a local min,
max, or inflection point
// before reaching the edge of the graph.
if (direction == SEARCH_LEFT && followingIndex
< 0) {
return new PointData(subset[0],
END_POINT);
}
return new PointData(subset[subset.length - 1],
END_POINT);
}

protected int checkMinMax(Point2D p1, Point2D p2,
Point2D p3) {
double slope1 = calcSlope(p1, p2);
double slope2 = calcSlope(p2, p3);
if (slope1 * slope2 <= 0.0) { // Local min /
max
return (slope2 > slope1) ? LOCAL_MIN :
LOCAL_MAX;
}
/*
* The three points do not form a local min or
local max.
*/
return -1;
}

protected int checkInflectionPoint(Point2D p1,
Point2D p2,
Point2D p3, Point2D p4, Point2D p5) {
double slope1 = calcSlope(p1, p2);
double slope2 = calcSlope(p2, p3);
double slope3 = calcSlope(p3, p4);
double slope4 = calcSlope(p4, p5);
boolean functionDecreasing = (slope1 < 0.0 &&
slope2 < 0.0
&& slope3 < 0.0 && slope4 < 0.0);
boolean functionIncreasing = (slope1 > 0.0 &&
slope2 > 0.0
&& slope3 > 0.0 && slope4 > 0.0);
if (!functionDecreasing && !functionIncreasing)
return -1; // It cannot be an inflection
point. The function must
// always be increasing or decreasing through
all five
// points.

```

```

// For the next test, the absolute value of the
slope is decreasing // for slopes 1 and 2 while increasing for
slopes 3 and 4, there is an // inflection point.

```

```

if (Math.abs(slope2) < Math.abs(slope1)
      && Math.abs(slope4) >
Math.abs(slope3))
    return INFLECTION_POINT;

```

```

// For the next test, the absolute value of the
slope is increasing // for slopes 1 and 2 while decreasing for
slopes 3 and 4, there is an // inflection point.

```

```

if (Math.abs(slope2) > Math.abs(slope1)
      && Math.abs(slope4) <
Math.abs(slope3))
    return INFLECTION_POINT;

```

```

// We have not determined that there is an
inflection point.
return -1;
}

```

```

/**
 * Extracts from an array of points in 2 dimensions
and forms a * one-dimensional array of data in the dimension
specified.
 *
 * @param data
 *         The Point2D[] array of points
 * @param dimension
 *         The dimension to extract (Valid
arguments: DataSetUtils.X or
 *         DataSetUtils.Y)
 * @return The 1-dimensional data of the Point2D[]
array in the dimension
 *         specified.
 */

```

```

public double[] toOneDimension(Point2D[] data, int dimension) {
    if (dimension != X && dimension != Y) {
        System.out

        .println("DataSetUtils.toOneDimension Error. Improper usage.
First argument is a Point2D[] to extract. Second argument is
DataSetUtil.X or DataSetUtils.Y depending on which dimension to
extract.");

        return null;
    }
    double[] retValue = new double[data.length];
    for (int i = 0; i < retValue.length; i++) {
        retValue[i] = (dimension == X) ?
data[i].x : data[i].y;
    }
    return retValue;
}

protected double calcSlope(Point2D p1, Point2D p2) {
    return ((p2.y - p1.y) / (p2.x - p1.x));
}
}
}

```

Anexo D – Estudo da Dow Chemical