

Universidade Federal do Rio de Janeiro

Escola Politécnica

Departamento de Eletrônica e de Computação

Software Educacional Gerenciador de Teias Alimentares

Autor:

Felipe Dias de Rezende Machado

Orientador:

Prof. Antônio Cláudio Gómez de Sousa, M. Sc.

Examinador:

Prof. Flávio Luis de Mello, D. Sc.

Examinador:

Prof. José Gabriel Rodriguez Carneiro Gomes, Ph.D.

DEL

Fevereiro de 2011

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
Escola Politécnica – Departamento de Eletrônica e de Computação
Centro de Tecnologia, bloco H, sala H-217, Cidade Universitária
Rio de Janeiro – RJ CEP 21949-900

Este exemplar é de propriedade da Universidade Federal do Rio de Janeiro, que poderá incluí-lo em base de dados, armazenar em computador, microfilmar ou adotar qualquer forma de arquivamento.

É permitida a menção, reprodução parcial ou integral e a transmissão entre bibliotecas deste trabalho, sem modificação de seu texto, em qualquer meio que esteja ou venha a ser fixado, para pesquisa acadêmica, comentários e citações, desde que sem finalidade comercial e que seja feita a referência bibliográfica completa.

Os conceitos expressos neste trabalho são de responsabilidade do(s) autor(es) e do(s) orientador(es).

AGRADECIMENTO

Agradeço primeiramente a Deus por me dar forças para prosseguir sempre. A meus pais Silvio Duarte Périco Machado e Rosane Dias de Rezende Machado que tanto me ajudaram nessa difícil tarefa de completar um curso de graduação na UFRJ. Ao meu irmão Jonathas por sua atenção e suas indispensáveis caronas. À minha cunhada Lucy Mary que tanto me apoiou nessa longa jornada. Ao orientador Antônio Cláudio Gómez de Sousa que disponibilizou seu tempo e atenção para que fosse possível o desenvolvimento desse projeto. Ao Ricardo Jullian e à Rejane Gadelha pelos conhecimentos adquiridos no LIpE, bem como a amizade construída durante esse período. Agradeço também a minha namorada Mariana Hammes Gonçalves por me dar força em sempre continuar e ajudar em conceitos de biologia. Aos companheiros de curso Daniel de Souza Dias, Gustavo dos Santos de Campos, Juliana Suleiman Kyriazis e a Rafael de Oliveira Rodriguez por me ajudarem a desenvolver um protótipo do software. Aos grandes amigos que sempre me deram força ao longo da vida: Bernardo, Diego, Natália, Thalita, Mateus, Flávio, Jander, Rui, Raphael, William, Suelen, João, entre outros, que se sintam citados.

RESUMO

O software em questão é do tipo educacional na área de conhecimento de biologia, especificamente no estudo de teias alimentares. Certas características são desejáveis para essa aplicação, tais como a interatividade e usabilidade. O objetivo é que o software seja aplicado em uma sala de aula a fim de poder reforçar o conteúdo do tema, além de despertar a curiosidade dos alunos.

Neste caso, foi escolhido um software que tenta reproduzir o comportamento de uma teia alimentar, que mesmo não sendo um ecossistema completo, ajuda na compreensão do equilíbrio delicado que existe entre as espécies. A modelagem dessa teia é complexa, e serão usados algoritmos matemáticos baseados no modelo Lotka-Volterra para populações.

Palavras-Chave: teias alimentares, educação, software educacional, biologia.

ABSTRACT

The software we propose is educational, for the field of biology, specifically for the study of food webs. Certain characteristics are desirable for this application, such as interactivity and usability. The objective is that the software be implemented in a classroom in order to enhance the content of the topic, besides fostering the curiosity of students.

In this case, we chose a software that tries to reproduce the behavior of a food web, which is not a complete ecosystem, but in the understanding of delicate balance that exists between species. The modeling of the web is complex, and will use mathematical algorithms based on the Lotka-Volterra population model.

Palavras-Chave: food webs, education, educational software, biology.

SIGLAS

UFRJ – Universidade Federal do Rio de Janeiro

GTA – Gerenciador de Teias Alimentares

IDE - Integrated Development Environment

ODBC - Open Data Base Connectivity

ERS – Especificação de Requisitos de Software

Sumário

Capítulo 1 - Introdução.....	1
1.1 – Tema	1
1.2 – Delimitação.....	1
1.3 – Justificativa.....	1
1.4 – Objetivos.....	2
1.5 – Metodologia.....	2
1.6 – Descrição	3
Capítulo 2 - Teias Alimentares	5
2.1 – Introdução	5
2.2 – Definições	5
2.3 – Dinâmica de teias alimentares	7
2.3.1 – Variação das populações	8
2.3.2 – Fluxo de energia	8
2.4 – Considerações para o projeto.....	9
Capítulo 3 - Especificação de Requisitos	10
3.1 – Introdução	10
3.2 – Considerações sobre o software	10
3.2.2 – Objetivos do software	10
3.2.3 – Público alvo	11
3.3 – Requisitos não funcionais.....	11
3.4 – Requisitos funcionais.....	12
3.4.1 – Casos de uso	12
3.4.2 – Classes de dados	14
3.4.3 – Dicionário de dados	15
3.4.3.1 – Teia	15

3.4.3.2 – Organismo.....	15
3.4.3.3 – Organismo da teia	15
3.4.3.4 – Relação teia.....	16
3.5 – Interfaces	16
3.5.1 – Interfaces de software	16
3.5.2 – Interfaces gráficas	17
3.6 – Conclusão	17
Capítulo 4 - Implementação	18
4.1 – Introdução	18
4.2 – Métodos e ferramentas.....	18
4.3 – Classes reais.....	19
4.3.1 – Módulo de dados	20
4.3.2 – Módulo de processamento	21
4.3.3 – Módulo de entradas e saídas	22
4.4 – Problemas e soluções.....	27
Capítulo 5 - Correção de Especificação de requisitos	28
5.1 – Introdução	28
5.2 – Interfaces gráficas	28
5.3 – Conclusão	29
Capítulo 6 - Correção de Implementação	30
6.1 – Introdução	30
6.2 – Ferramentas	30
6.3 – Algoritmo da teia	30
6.4 – Gráfico	32
6.5 – Instalador	33
Capítulo 7 - Resultados	34
7.1 – Introdução	34

7.2 – Quantitativo	35
7.3 – Qualitativo	36
7.4 – Avaliação	37
7.5 – Conclusão	37
Capítulo 8 - Conclusão	38
8.1 – Métodos e ferramentas utilizados	38
8.2 – Conhecimentos adquiridos	39
8.3 – Propostas futuras	40
Referências Bibliográficas.....	41

Apêndice A – Documento de Planejamento

Apêndice B – Documento de Especificação de Requisitos de Software (ERS)

Apêndice C – Documento de Projeto

Apêndice D – Documento de Projeto de Testes

Apêndice E – Manual do Usuário

Anexo A – Normas da documentação

Anexo B – Fichas de avaliação de software

Lista de Figuras

Figura 1 - Exemplo de cadeia alimentar	6
Figura 2 - Exemplo de teia alimentar	6
Figura 3 - Pirâmide ecológica de energia	8
Figura 4 - Casos de uso	13
Figura 5 - Classes de dados	14
Figura 6 - Divisão de módulos	19
Figura 7 - Classe Bd	20
Figura 8 - Classes de dados (Reais)	21
Figura 9 - Classe Simulador	21
Figura 10 - Classes de saída	22
Figura 11 - Interface gráfica do software – Aba fluxograma	23
Figura 12 - Interface gráfica - Aba imagem	24
Figura 13 – Classes interface gráfica 1	25
Figura 14 - Classes interface gráfica 2	26
Figura 15 - Interface gráfica para o gráfico	29
Figura 16 - Fluxo do algoritmo da teia	31
Figura 17 - Divisão dos módulos - inserção do gráfico	32
Figura 18 - Classes de saída - com gráfico	32

Lista de Tabelas

Tabela 1 - Tabela de resultados.....	35
--------------------------------------	----

Capítulo 1

Introdução

1.1 – Tema

O presente trabalho consiste no desenvolvimento de um software educacional na área de biologia, especificamente tratando do conceito de teias alimentares. Esse software pretende causar uma melhora no aprendizado do tema abordado, sendo usado não só como apoio em aulas ministradas por professores regulares de biologia como também pelos próprios alunos em casa. Utilizando técnicas de Engenharia de Software obteve-se uma primeira versão do software com a respectiva documentação.

1.2 – Delimitação

O software restringe-se à área de teias alimentares, tratando apenas de seus aspectos mais simples e básicos, como a consideração de poucas características dos organismos, tais como peso, taxa de mortalidade e quanto necessita de alimentação. Essa restrição se deve ao fato do software ser voltado para o ensino básico, não sendo necessária uma estrutura muito complexa.

1.3 – Justificativa

Quando se fala em software educacional destacam-se alguns pensamentos divergentes. Algumas pessoas acham que o software educacional irá substituir o professor, outras acreditam no auxílio que esses softwares poderiam fornecer ao aprendizado do aluno. Em parte isso ocorre devido à dificuldade da quebra do tradicional estilo de aula por parte dos professores, que consiste no professor explicando aos alunos utilizando o quadro negro como instrumento. Já os que acreditam no software educacional apostam em uma possível melhora no aprendizado dos alunos fazendo uso dessa ferramenta.

Utilizando somente a audição, a absorção de informação não é tão grande, podendo causar uma dificuldade de aprendizado. Com a visão, há a possibilidade de uma memorização melhor. Porém se for capaz de interagir com o conteúdo, gerando erros e acertos, a capacidade de aprendizado é ampliada e possivelmente o conteúdo será aprendido.

Há um problema sobre a especificidade do software educacional. Caso seja muito genérico pode não atingir objetivo algum, e se for muito específico será utilizado somente para o objetivo que foi projetado. Nesse sentido pode-se observar que deveria existir um número grande de softwares educacionais específicos para cada área.

Com essas considerações, o projeto utilizou o estudo de teias alimentares, para o desenvolvimento do software. Tem como base o funcionamento da teia e como ocorrem as interações entre os organismos existentes nela. As características dos organismos são alteradas pelos alunos proporcionando um melhor entendimento do conhecimento, além da verificação do fato de uma pequena alteração poder gerar uma grande mudança em todos os organismos presentes na teia.

1.4 – Objetivos

O objetivo geral do projeto é desenvolver uma ferramenta educacional gerenciadora de teias alimentares. Como objetivos específicos podem ser citados a criação de documentação técnica e um manual do usuário. O software tem como objetivos a melhoria na atratividade do assunto para os alunos e a possibilidade do software ser usado de forma autônoma como exercício.

1.5 – Metodologia

Para a realização do projeto foram usadas as teorias de Engenharia de Software para o desenvolvimento em cascata do software, incluindo etapas como planejamento, especificação de requisitos, desenvolvimento de manual do usuário, projeto, implementação e testes.

Na fase de planejamento foram especificadas as dificuldades, as possíveis soluções, restrições, gerenciamento de dificuldades e medidas de controle. Na especificação de requisitos foram abordados temas como os casos de uso implementados, a delimitação do escopo, definição de interfaces e a descrição de entidades. No manual, foi realizada uma descrição das funções do software com o auxílio de imagens e linguagem natural. No projeto foram especificados os grandes módulos do software, implementados na fase seguinte. Depois da implementação, planos de testes foram desenvolvidos para mitigar os defeitos ou até mesmo corrigir na sua totalidade. Após todo o processo, uma primeira versão do software foi gerada com sua respectiva documentação.

O foco do software está na usabilidade, de forma que fique atrativo e de fácil utilização para os alunos. O aluno entrará com dados, como peso, taxa de mortalidade, de quanto o organismo se alimenta e o software fará os cálculos das populações mostrando, em números, quanto há dos organismos em cada rodada, podendo haver extinção da espécie ou até uma possível superpopulação. Esses dados fornecidos pelos alunos serão armazenados em um banco de dados instalado na própria máquina.

1.6 – Descrição

No **capítulo 2** é apresentada uma descrição sobre **teias alimentares**, incluindo explicações sobre conceitos e definições utilizados no desenvolvimento do projeto.

A seguir, no **capítulo 3** está feita a **especificação de requisitos** do software, que serve de parâmetro delimitador para a fase de implementação, apresentando casos de uso, classes de dados e requisitos não funcionais. Importante dizer que essa foi uma primeira especificação de requisitos, posteriormente, uma alteração (será explicada nas seções seguintes) foi feita para promover uma melhoria no projeto.

O **capítulo 4** abrange o processo da **implementação** do software, explicitando suas divisões e classes.

O **capítulo 5** explica uma segunda **especificação de requisitos**, onde ocorreram alterações necessárias para o bom funcionamento do sistema. Da mesma forma como foi feito anteriormente, o **capítulo 6** visa demonstrar como foi **implementada** a modificação realizada na especificação de requisitos.

No **capítulo 7** são apresentados **resultados** quantitativos e qualitativos com testes realizados em turmas de 7º, 8º e 9º anos do ensino fundamental.

Encerrando o projeto, o **capítulo 8** mostra as **conclusões** do projeto, citando possíveis implementações, avaliando as ferramentas e métodos utilizados e qual a contribuição pessoal e profissional proporcionada por esse projeto.

No **apêndice** encontra-se a **documentação** gerada durante o desenvolvimento do software. Há também um **CD** contendo o código fonte e a mesma documentação.

Em **anexo** estão as normas utilizadas para desenvolver a documentação e o modelo das fichas de avaliação do software utilizadas por alunos e professores.

Capítulo 2

Teias Alimentares

2.1 – Introdução

Este capítulo se propõe a definir uma teia alimentar, mostrar o funcionamento dela, assim como definir alguns termos que foram usados no software, na documentação, ou até mesmo termos utilizados durante esse documento.

Algumas considerações foram tomadas para o desenvolvimento do software, visto que este deveria ser voltado para o ensino fundamental (do 6º ao 9º anos) e médio. Estas considerações foram impostas com a finalidade de simplificar e focar no ensino de teias alimentares.

2.2 – Definições

Uma cadeia alimentar pode ser considerada como uma sequência de seres vivos que se alimentam uns dos outros (Linhares e Gewandsznajder [4]), possibilitando a transferência da energia dos produtores para os consumidores subsequentes. Os produtores são seres autotróficos, ou seja, produzem o seu próprio alimento (matéria orgânica) a partir de matéria inorgânica. Os consumidores são chamados de heterotróficos, necessitando assim de outro ser para se alimentar.

Partindo daí, surge o conceito de níveis tróficos, que segundo Lindeman (apud Ricklefs [1]) são determinados pela existência de produtores, herbívoros e carnívoros. Cada nível trófico constitui uma categoria onde os seres se encaixam na cadeia alimentar de acordo com a sua forma de alimentação. A Figura 1 mostra uma cadeia alimentar de três níveis, onde a embaúba está no primeiro nível trófico, sendo ela a produtora, o macaco está no segundo nível trófico, consituindo o herbívoro ou consumidor primário, e no terceiro nível está a onça, considerada carnívora ou consumidora secundária. Segundo Lawton (apud Lima & Selles [2]), caso ocorra de um ser se alimentar de dois outros seres de níveis tróficos diferentes, será chamado de onívoro.

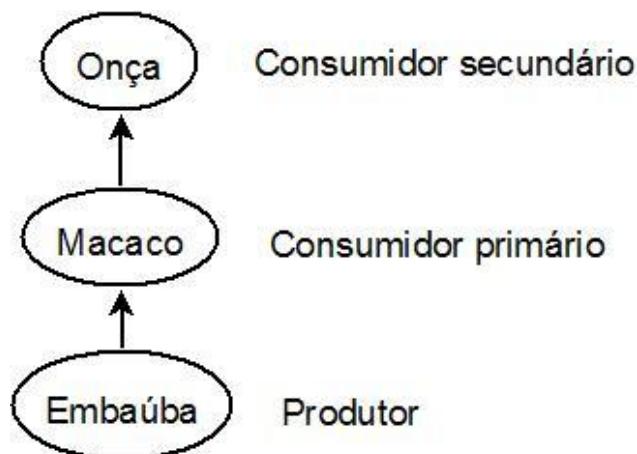


Figura 1 - Exemplo de cadeia alimentar com respectivos níveis tróficos.

O termo teia alimentar surge quando há a interligação de várias cadeias alimentares, e pelo menos um organismo está presente em duas cadeias ao mesmo tempo (Figura 2). Como foi explicado acima, os seres também são encaixados em níveis tróficos diferentes de acordo com sua forma de alimentação. Cabe também elucidar que ao tratar de teias alimentares cada organismo representa, na verdade, uma população, um conjunto de organismos de mesma espécie vivendo ao mesmo tempo, no mesmo local (Ricklefs [1]).

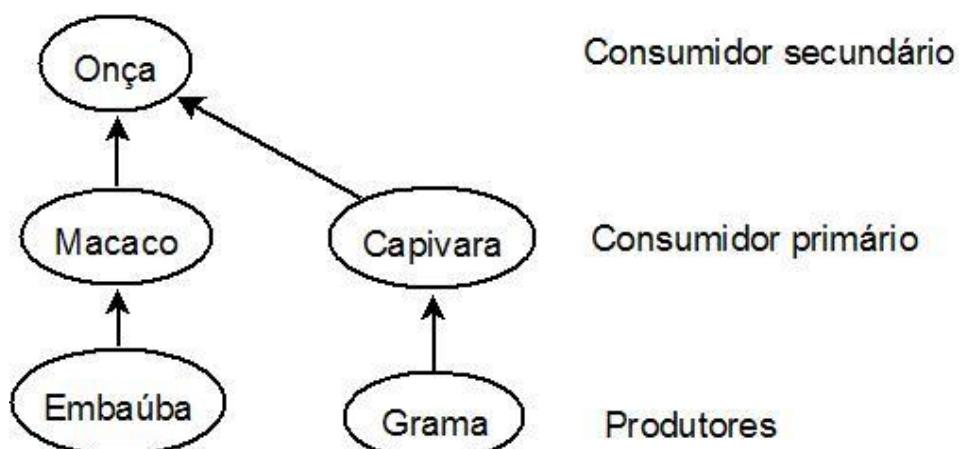


Figura 2 - Exemplo de teia alimentar.

Segundo Ricklefs [1], as interações entre os seres vivos se encaixam em três diferentes tipos: predador-presa, herbívoro-planta e parasito-hospedeiro. Para todas essas interações há a caracterização de benefícios (positivo) e malefícios (negativo)

entre os organismos envolvidos. O predador, o herbíboro e o parasita são beneficiados enquanto a presa, a planta e o hospedeiro são prejudicados.

A interação predador-presa caracteriza uma predação. Para esse caso o predador caça a presa, mata e se alimenta dela. Uma característica importante dessa interação é a morte da presa, causando a diminuição da população.

Parasito-hospedeiro se assemelha ao predador-presa no sentido de um organismo ser beneficiado e o outro prejudicado. Porém, nesse caso, é importante para o parasita que o hospedeiro se mantenha vivo, e dessa forma não causa uma diminuição na população, mesmo que a existência do parasita venha a prejudicar a sobrevivência do hospedeiro. O parasitismo, caracterização da interação parasito-hospedeiro, pode ocorrer nos diferentes níveis tróficos da cadeia alimentar.

Semelhante ao modelo predador-presa existe o modelo herbíboro-planta. Porém, nesse caso não há caça, ou seja, as plantas não se movem e dessa forma os herbívoros se alimentam mais facilmente que os carnívoros. Nesse caso pode ocorrer a retirada total da planta, agindo assim como um predador, ou uma retirada parcial, agindo como parasita.

2.3 – Dinâmica de teias alimentares

“[...] Sem saber, poderíamos matar um animal importante, um pequeno pássaro, uma barata; mesmo uma flor, assim destruindo um elo importante, numa espécie em crescimento.” [3].

“[...] Esmagar certas plantas pode provocar alterações infinitesimais. Um pequeno erro aqui pode se multiplicar ao longo de sessenta milhões de anos e deixar tudo fora de proporção.” [3].

Os textos acima conseguem elucidar a idéia de uma teia alimentar. Essa idéia é dada no sentido de que uma pequena modificação na população de qualquer indivíduo da teia alimentar pode gerar um grande desequilíbrio em toda a teia.

2.3.1 – Variação das populações

As populações das teias alimentares não interagem de forma constante, assim podem ocorrer variações populacionais bruscas ou sutis. Um aumento ou uma diminuição brusca num número de indivíduos de uma população pode alterar toda a teia alimentar, já que ela está baseada nas relações citadas anteriormente. É possível perceber dessa forma que a quantidade de presas dependerá da quantidade de predadores, e vice-versa. Uma superpopulação de predadores pode extinguir toda a população de sua presa, e como pensamento inverso, a diminuição da presa ocasionará uma redução dos predadores.

As causas para as flutuações do número de indivíduos de uma população podem ser diversas. Morte, natalidade e dispersão são algumas dessas causas.

2.3.2 – Fluxo de energia

Na teia alimentar o fluxo de energia flui dos produtores até os últimos consumidores (Figura 3). Porém, nessa passagem de um nível trófico para outro, apenas parte da energia é transmitida, outra parte é perdida durante transformações bioquímicas (Ricklefs [1]). Essa energia passada nunca chega a 100% e pode variar muito dependendo do organismo que está servindo de alimento.

“[...] Os herbívoros assimilam cerca de 80% da energia nas sementes e de 60% a 70% na vegetação jovem. A maioria dos pastadores de folhas e brotos (elefantes, gado, gafanhoto) extraem de 30% a 40% da energia da comida para si próprios. Os milípedes [...] assimilam somente 15% da energia contida em sua dieta.” (Ricklefs[1])

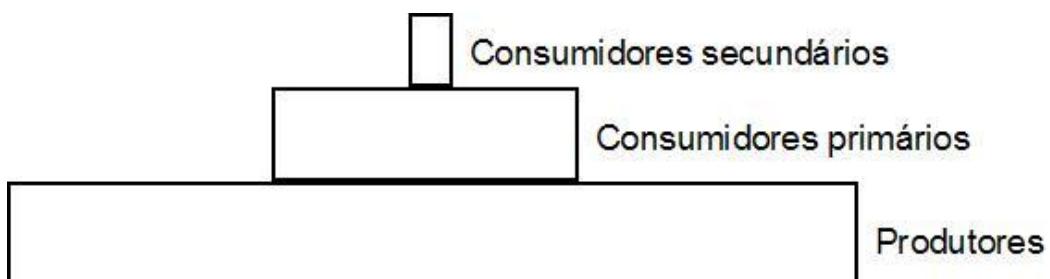


Figura 3 - Pirâmide ecológica de energia

2.4 – Considerações para o projeto

Analisando o que foi explicado anteriormente pode-se perceber que existem muitas variáveis no tratamento de teias alimentares, considerações de meio ambiente, características físicas de cada organismo, a relação entre eles, são alguns exemplos. Porém, o objetivo do software é servir como ferramenta didática para a fixação e auxílio no ensino de teias alimentares, e tais objetivos poderiam não ser alcançados caso a complexidade do software fosse muito grande. Para tal, o número de considerações foi diminuído a algumas características apenas, porém, o suficiente para explorar os conceitos do tema.

As características do meio ambiente (sol, chuva, vento, nutrientes, entre outros) foram todas agrupadas em uma única característica, condições para a teia. As características físicas e fisiológicas dos organismos foram reduzidas a peso, quanto o organismo come para sobreviver e taxa de mortalidade. Como forma de interação foi escolhida somente a predação.

Com essas simplificações foi possível desenvolver um software simples e que conseguisse assegurar os objetivos propostos.

Capítulo 3

Especificação de Requisitos

3.1 – Introdução

Este capítulo é destinado à descrição da especificação de requisitos, e com isso apresenta algumas informações do conteúdo do documento ERS (Especificação de Requisito de Software), que está no apêndice, e mostra como foram pensadas as características do software.

Essa análise de requisitos foi utilizada como guia para a próxima fase, a implementação do software. Nesta etapa são definidas as classes de dados, os casos de uso, as interfaces do software, entre outras funcionalidades que são descritas nos tópicos a seguir.

3.2 – Considerações sobre o software

3.2.2 – Objetivos do software

Como já foi explicado no **capítulo 1**, o software possui objetivos específicos que devem ser considerados no instante de seu projeto, tanto quanto na sua implementação. Foi dito que os objetivos são a melhoria na atratividade do assunto para os alunos e a possibilidade do software ser usado de forma autônoma como exercício. Em ambos os objetivos é recorrente a importância da aparência do software, e essa característica, juntamente com a facilidade de utilização, foram guias no processo de desenvolvimento.

3.2.3 – P blico alvo

O software considerou a exist ncia de dois p blicos distintos: os alunos e os professores. Pensou-se nesses dois p blicos para desenvolver um software agrad vel para ambos, tanto o professor que est  utilizando o software para auxiliar no ensino, quanto o aluno que est  aprendendo atrav s dele.

No perfil dos alunos considerou-se que est o no ensino b sico, englobando o ensino fundamental (do 6^o ao 9^o) e o ensino m dio. Levou-se em considera o tamb m que o aluno deve conseguir utilizar o software sem o aux lio de nenhum instrutor, ou professor, utilizando somente o manual como consulta, desde que o aluno j  detenha o conhecimento de teias alimentares.

Para os professores desejou-se inserir no software certa flexibilidade de a es, possibilitando assim um aumento da gama de assuntos que podem ser adotados. Esses assuntos podem at  n o ter liga o alguma com as teias alimentares, basta apenas ao professor uma boa imagina o e disposi o para usar o software, descobrindo tais possibilidades.

3.3 – Requisitos n o funcionais

Nesse software foram escolhidos tr s requisitos n o funcionais: usabilidade, manutenibilidade e confiabilidade. Foram escolhidos esses requisitos para serem levados em considera o, pois se encaixavam com os objetivos do software e do projeto.

A usabilidade foi um dos requisitos, pois o software deve ser de f cil utiliza o, mesmo para pessoas que jamais utilizaram softwares com caracter sticas semelhantes. Para isso as interfaces gr ficas devem ser simples e intuitivas, e o *menu* deve conter as informa es e fun es necess rias para utiliza o correta do sistema. Essa caracter stica  a mais importante do software, e para controle o software foi testado e avaliado ap s a implementa o com algumas turmas, a fim de verificar se esta caracter stica foi satisfeita ou n o.

Com a possibilidade de o software ser de c digo aberto, a manutenibilidade torna-se uma caracter stica relevante. Nesse caso a manuten o deve ser f cil, de forma a corrigir rapidamente os erros encontrados e, al m disso, a implementa o de uma

nova funcionalidade deve ser simples de ser aplicada. Para alcançar tais objetivos, o código fonte deve estar bem organizado e devidamente comentado para facilitar aos interessados.

Por fim, um requisito que é fundamental em praticamente todos os projetos de engenharia de software, a confiabilidade. O sistema deve ser capaz de fornecer os serviços necessários conforme o especificado. Mesmo não se tratando de um sistema crítico, cuja confiabilidade deve ser bem alta, a confiabilidade é intimamente ligada à aceitação do produto. Se o software não realiza o que se propõe, tende a ser rejeitado pelos usuários (Sommerville [5]). Para o controle desse requisito foi utilizado o mesmo teste e avaliação feito para a usabilidade.

3.4 – Requisitos funcionais

3.4.1 – Casos de uso

Para essa versão do software foram escolhidos alguns casos de uso. Para versões posteriores do software há a possibilidade de aumento, ou até mesmo alteração em alguns deles, seja na forma de acesso ou no conjunto de ações às quais está associado.

Os casos de uso podem ser vistos no diagrama, assim como sua hierarquia de acesso:

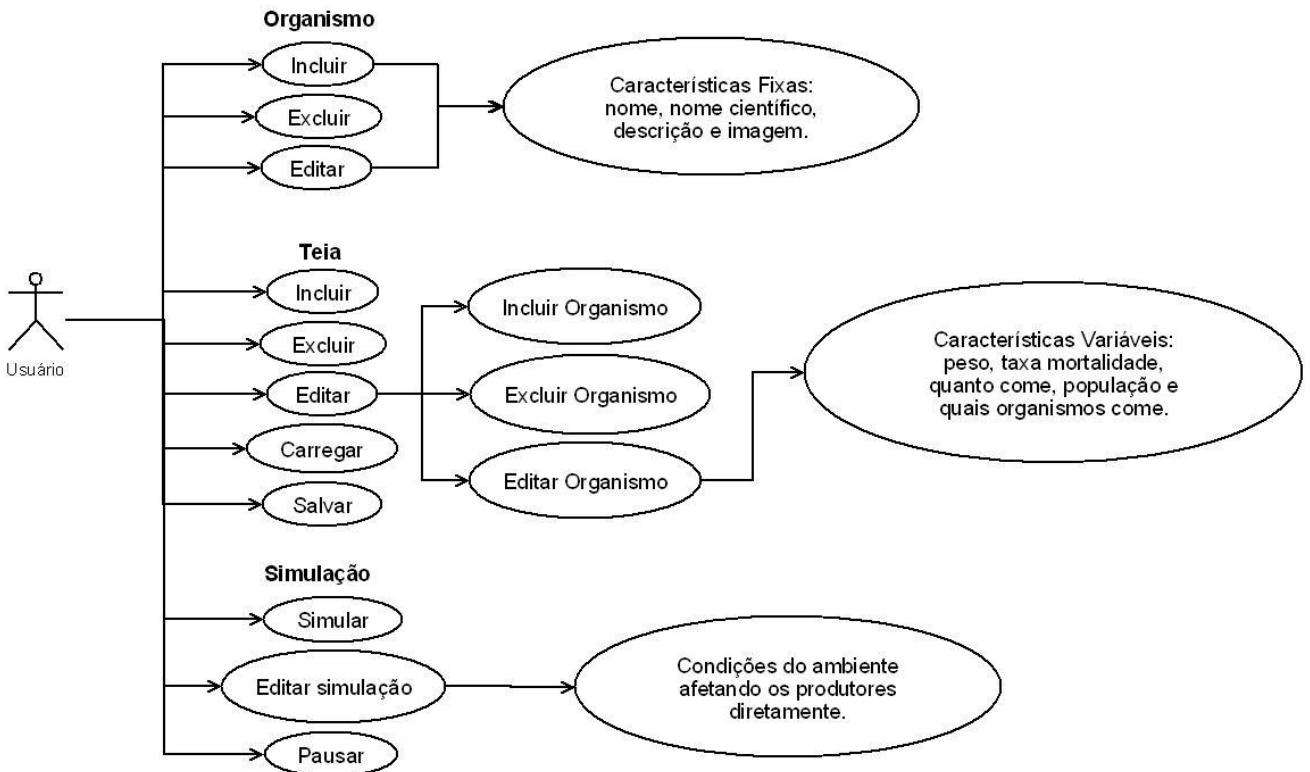


Figura 4 - Casos de uso.

Os casos de uso acima foram pensados para criar operações simples, diretas e eficazes. Pode-se dividir os casos de uso em três categorias: organismos, teia e simulação.

No caso dos organismos, o software disponibilizará ao usuário a inclusão de novos organismos no sistema, além da edição e exclusão dos já existentes.

Na categoria de teia, há a possibilidade de criar novas teias, salvá-las, carregá-las posteriormente e excluí-las. Ainda nesse escopo, porém atuando na parte funcional, poderão ser incluídos organismos nas teias, bem como suas características e relações com outros organismos. Os organismos incluídos podem ser excluídos também, assim como suas relações. É importante citar que para cada teia os organismos podem possuir características diferentes, por exemplo: em uma teia o coelho pesa 5 kg, e na outra pode pesar 50 kg.

Como última categoria encontra-se a simulação. Nessa categoria há possibilidades como iniciar a simulação, pausar a simulação e alterar as características do ambiente no qual a teia está imersa. Quando a simulação estiver em pausa, todos os dados da teia podem ser alterados, e após essa alteração a simulação poderá ser iniciada com os novos dados alterados.

Para a verificação completa dos procedimentos de cada caso de uso, utilizar o documento no apêndice “ERS – Especificação de Requisitos de Software”.

3.4.2 – Classes de dados

A modelagem dos dados desse software apresentou uma grande dificuldade ao ser feita. Dificuldade essa apoiada pela falta de experiência em modelagem e também pela dificuldade do problema. Como solução foi proposto o diagrama abaixo, contendo as classes de dados nas quais o software será baseado.

Para executar a modelagem foi proposto que um organismo pudesse pertencer a diferentes teias com características diferentes, porém com algumas características fixas: nome, nome científico, descrição e imagem. Para as características variáveis foram considerados: peso, taxa de morte, quantidade de comida ingerida por rodada, assim como a população.

Também foram consideradas características específicas de cada relação: valor de preferência para alimentar-se da presa, e quanto da presa foi comido pelo predador.

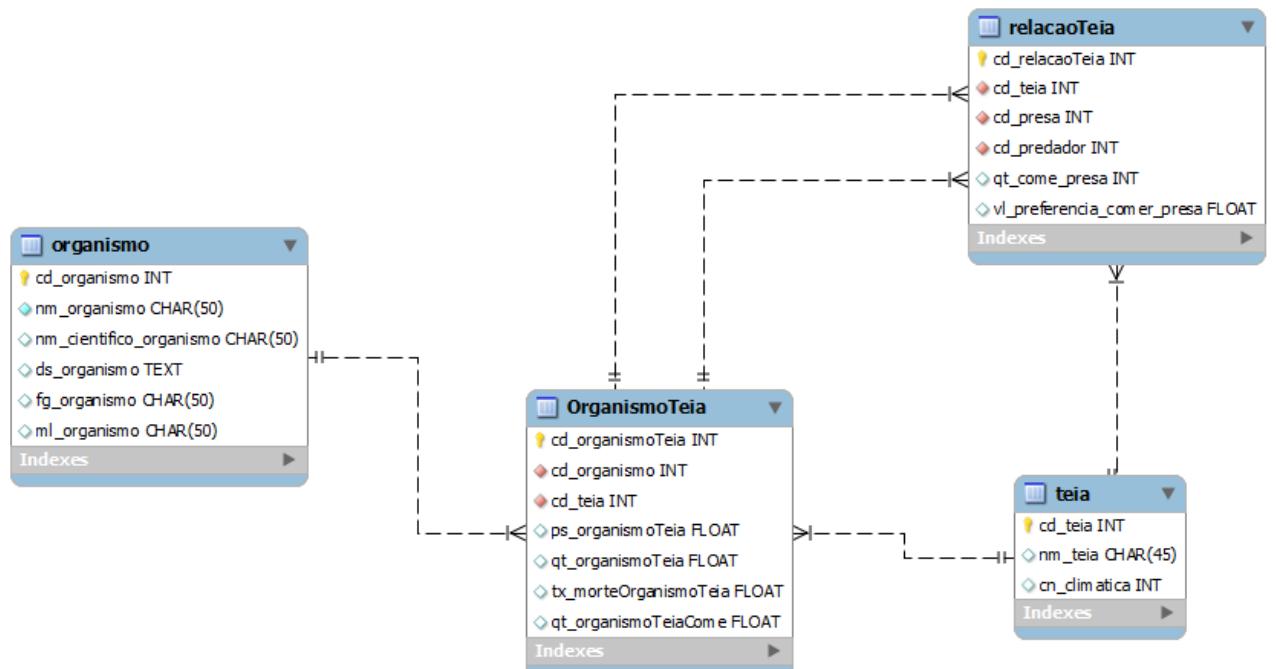


Figura 5 - Classes de dados

3.4.3 – Dicionário de dados

3.4.3.1 – Teia

Estrutura principal, que contém as informações de uma teia criada pelo usuário:

- cd_teia - Código usado para a identificação de uma teia;
- nm_teia - Nome da teia utilizada;
- cn_climáticas - Valor de condições de simulação da teia. É usado para que seja possível variar condições de simulação pelo aluno, assim gerando resultados distintos.

3.4.3.2 – Organismo

Estrutura que contém as informações de cada organismo presente no software. Não é dependente de teia:

- cd_organismo - Código usado para identificar o organismo;
- nm_organismo - Nome atribuído ao organismo;
- nm_cientifico - Nome científico atribuído ao organismo;
- fg_organismo - Endereço da figura correspondente ao organismo;
- ml_organismo - Modelo do organismo (utilização futura);
- ds_organismo - Descrição breve sobre o organismo.

3.4.3.3 – Organismo da teia

Estrutura que contém os dados específicos de um organismo que está incluído em uma teia. Foi criada para permitir que um organismo possa ter diferentes atributos em diferentes teias:

- cd_organismoTeia – Código usado para identificar o organismo da Teia;
- cd_teia - Código usado para a identificação de uma teia;
- cd_organismo - Código usado para identificar o organismo;
- ps_organismo - Peso do organismo em determinada teia (em kg);

- qt_organismoTeia - Quantidade inicial de um organismo (população);
- qt_OrganismoTeiaCome - Quantidade que o organismo necessita para sobreviver em determinada teia (em kg por dia);
- tx_morteOrganismoTeia - Taxa em que a população do organismo de uma determinada teia irá diminuir por rodada (variando de 0 até 1, em porcentagem).

3.4.3.4 – Relação teia

Estrutura que contém as relações entre os organismos que compõe a teia.

- cd_relacaoTeia – Código usado para a identificação de uma relação de teia;
- cd_teia - Código usado para a identificação de uma teia;
- cd_presa - Código usado para a identificação de um organismo da teia;
- cd_predador - Código usado para a identificação de um segundo organismo da teia;
- qt_come_presa – Quantidade consumida pelo predador da presa (utilização futura).

3.5 – Interfaces

3.5.1 – Interfaces de software

Pensando em implementações futuras e em como seria feita a persistência dos dados, foi escolhido um banco de dados. Então, dessa forma, o software apresenta uma interface com o banco de dados escolhido para esse projeto, MySQL, utilizando ODBC (gerenciador de fontes de dados). Essa escolha possibilitará uma implementação do software utilizando redes, possuindo um servidor conectado ao banco de dados e as outras máquinas clientes fazendo o acesso por ele.

3.5.2 – Interfaces gráficas

As interfaces foram pensadas de forma a preencherem os requisitos não funcionais especificados, ou seja, deveriam ser fáceis e intuitivas na sua utilização. Para verificação das interfaces, olhar o documento do apêndice “ERS – Especificação de Requisitos de Software”, onde se podem encontrar figuras das interfaces, bem como uma explicação sobre cada uma delas. Importante citar que o documento contém a interface para o gráfico, que não consta da versão especificada neste capítulo. O gráfico foi inserido posteriormente e mostrado no capítulo de complementação de especificação de requisito do software.

3.6 – Conclusão

Partindo dessa especificação elaborou-se um projeto de software, que será descrito no próximo capítulo. Com esse projeto foi gerada uma versão do software, posteriormente melhorada, que está documentada nos capítulos de correção.

Capítulo 4

Implementação

4.1 – Introdução

O presente capítulo apresenta a estrutura planejada para a implementação, e como essa implementação foi feita, descrevendo ferramentas utilizadas e alguns problemas encontrados com suas respectivas soluções. As seções descritas nesse capítulo visam elucidar melhor esse projeto.

4.2 – Métodos e ferramentas

Seguiu-se o ciclo de vida cascata desde o início do projeto. Partindo do documento de especificações de requisitos, desenvolveu-se outro documento de projeto que especifica como implementar o que foi especificado.

No desenvolvimento desse software foram usadas algumas ferramentas que auxiliaram o gerenciamento e a implementação.

Para o gerenciamento foram escolhidas ferramentas de controle de versão, no caso foi usado o TortoiseSVN v1.6.10, com o intuito de proteção de versões do software que funcionam, assim como gerenciar o que foi alterado em determinada versão. Ainda para gerenciamento foi utilizado um repositório de arquivos na internet. Esse repositório tem o nome de Dropbox, é uma ferramenta que permite que os dados do seu computador estejam sincronizados com os dados do site, além disso, é possível acessar os dados de qualquer lugar. Para a confecção de diagramas foi utilizado o software DIA v0.96.1, que possui uma gama de funcionalidades para fluxogramas, diagramas de classes, entre outros, além de ser gratuito e multiplataforma.

Na implementação foi utilizada a IDE do DevC++ v4.9.9.2, facilitando assim a compilação na linguagem C++, a organização dos arquivos de projeto e a instalação de bibliotecas. Foi utilizada a biblioteca a WxWidgets v2.6 para o desenvolvimento das interfaces, comunicação com o banco de dados, bem como outras funções implementadas. Como já foi dito anteriormente, o banco de dados utilizado foi o

MySQL v5.1.53 e para a conexão ODBC foi utilizado o conector Connector/ODBC v5.1.8.

4.3 – Classes reais

O software foi dividido em módulos para facilitar a visibilidade do sistema, e como consequência, facilitar a criação das classes que foram necessárias para desenvolver o software. As classes reais incluem as classes de dados, que foram citadas no capítulo anterior sobre a especificação dos requisitos do software, com as classes utilizadas para interface e outras funcionalidades. Serão mostradas as classes que foram implementadas, com seus respectivos métodos e atributos.

A divisão de módulos foi escolhida de forma a separar as classes que fariam o processamento, as classes de entrada e saída e classes de dados. A Figura 6 mostra as estruturas que são compostas por uma ou mais classes.

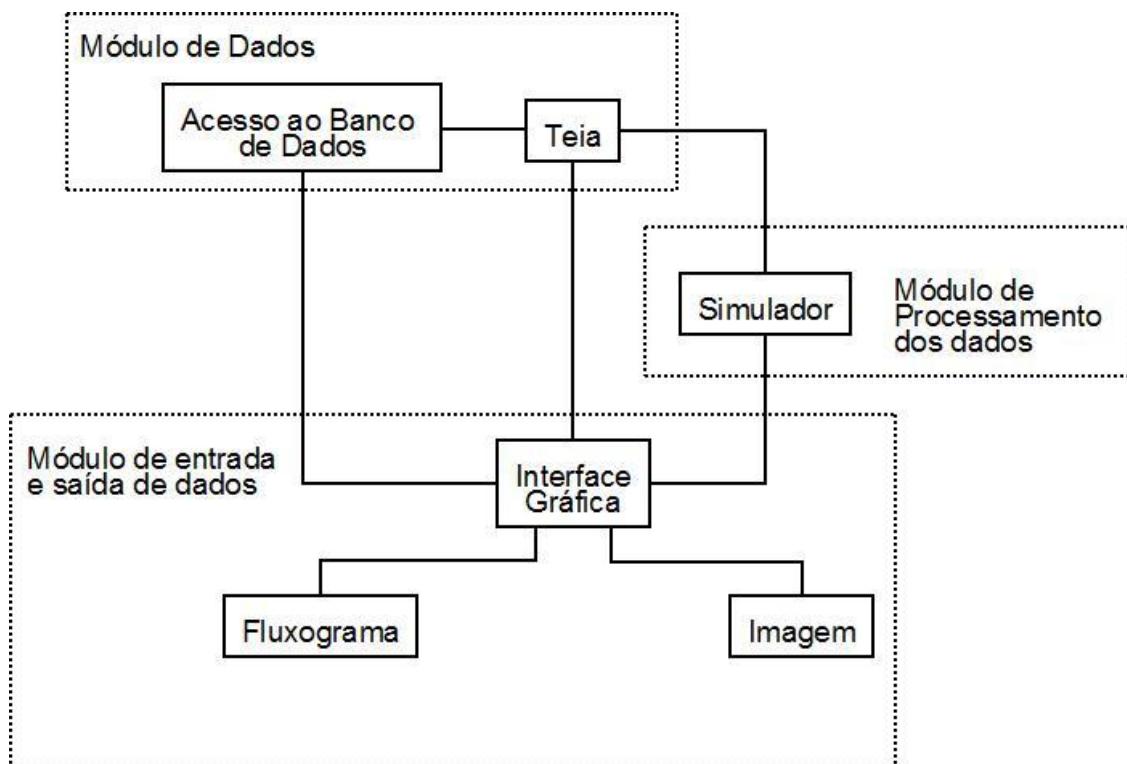


Figura 6 - Divisão de módulos

4.3.1 – Módulo de dados

No módulo de dados, há a parte que trata do acesso ao banco de dados. Essa parte possui apenas uma classe, que é responsável pela conexão com o banco de dados e possui todos os métodos de acesso para adquirir informação, gravar ou apagar (Figura 7).

Bd
+conectarBd(): bool +desconectarBd(): void +setConexaoBd(nomeFonte:String, usuario:String, senha:String) +getOrganismo(cdOrganismo:int): OrganismoTeia +setOrganismo(organismo:OrganismoTeia): bool +deleteOrganismo(cdOrganismo:int): bool +getListaOrganismos(): listaOrganismos +getOrganismoTeia(cdOrganismoTeia:int, cdTeia:int): OrganismoTeia +setOrganismoTeia(organismo:OrganismoTeia, cdTeia:int): bool +deleteOrganismoTeia(organismo:OrganismoTeia): bool +getListaOrganismoTeia(cdTeia:int): listaOrganismo +getRelacaoTeia(nmPredador:String, nmPresa:String, cdTeia:int): RelacaoTeia +setRelacao(relacao:RelacaoTeia, cdTeia:int): bool +deleteRelacao(relacao:RelacaoTeia): bool +setListaRelacao (lista:listaRelacoes): bool +deleteListaRelacao(cdTeia:int): bool +getTeia(cdTeia:int): Teia +setTeia(teia:Teia): bool +deleteTeia(cdTeia:int): bool

Figura 7 - Classe Bd

Ainda no módulo de dados há a parte onde são tratadas as informações da teia. Nessa parte existem algumas classes para essa representação. As classes de organismo, organismo da teia, relações da teia e teia compõem esse bloco. Essas classes de dados foram previstas no capítulo anterior sobre a especificação de requisitos de software, e nesse capítulo são mostradas da forma que estão implementadas (Figura 8).

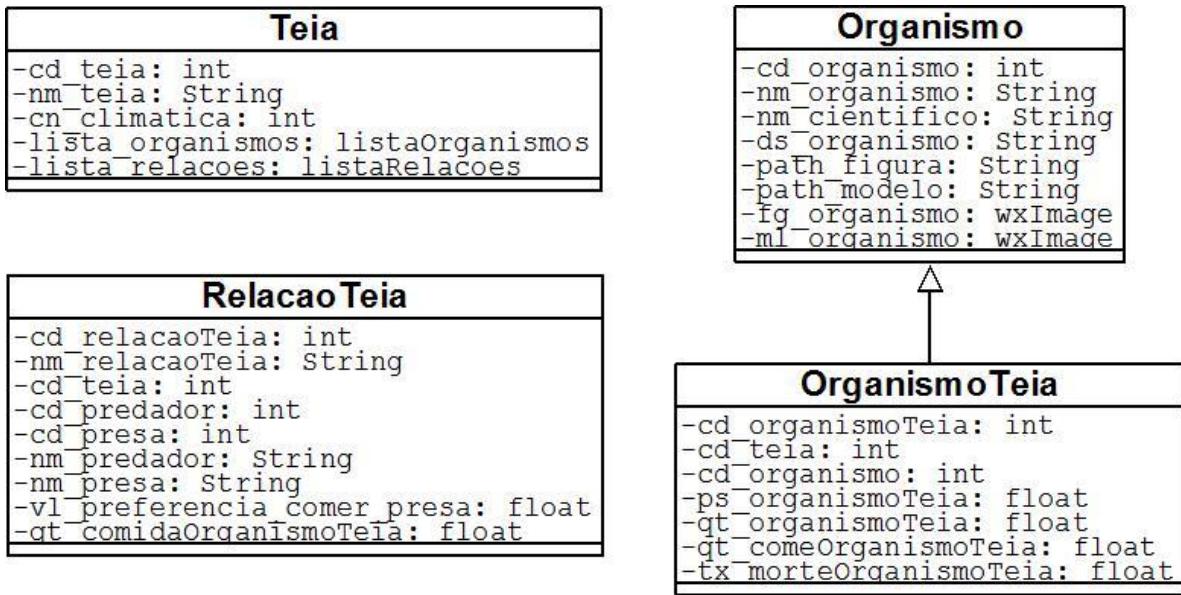


Figura 8 - Classes de dados (Reais)

4.3.2 – Módulo de processamento

No módulo processamento existe uma classe, a classe simulador. Ela é responsável pela resolução do algoritmo da teia, ou seja, ela recebe as populações de todos os organismos presentes na teia, e calcula qual será a próxima população (Figura 9).



Figura 9 - Classe Simulador

O algoritmo utilizado nessa versão foi baseado no modelo Lotka-Volterra. Há uma explicação sobre esse modelo no documento de projeto que está em apêndice. Esse modelo trata de uma relação simples entre um predador e uma presa. Porém, nesse algoritmo deveria ser considerada a presença de vários organismos com várias relações diferentes. Uma possibilidade encontrada foi a de tratar as relações individualmente utilizando o modelo Lotka-Volterra como se os seres fossem isolados. Essa consideração foi feita devido à forte relação que existe entre os organismos de uma teia

alimentar. Quando ocorre a alteração de uma população de um organismo, influencia obrigatoriamente a todas as outras populações.

4.3.3 – Módulo de entradas e saídas

No módulo de entradas e saídas, existem classes que estão em funções diferenciadas. A interface é uma interface gráfica, que é responsável pela aquisição de dados do usuário e pela demonstração das saídas. Porém, é responsável por mostrar somente uma imagem das saídas, as quais são processadas em duas outras classes: a classe Imagem e a classe Fluxograma (Figura 10). Para detalhes de funcionamento das classes, olhar no documento de projeto em apêndice.

Fluxograma	ImagenSaida
<pre>-teia atual: Teia -pictūre: wxBitmap -posList: listaPosicoes -tamList: listaTamanhos -niveis: niveisTroficos +DoFluxograma () : wxBitmap</pre>	<pre>-list: listaOrganismos -imagem: wxImage -backgroundImagem: wxImage +DoImagen () () : wxImage</pre>

Figura 10 - Classes de saída

A interface foi pensada para que o visual fosse atraente, além de ser de fácil utilização (Figura 11). Foi projetada com a estrutura em abas, separando fisicamente as entradas das saídas do software.

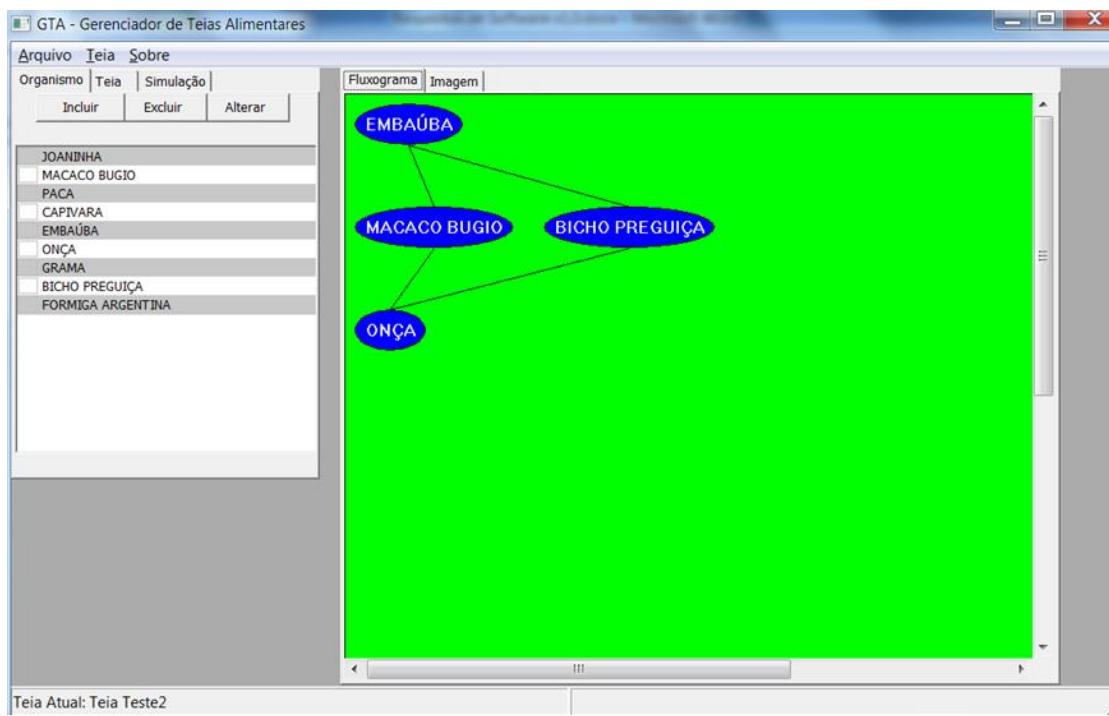


Figura 11 - Interface gráfica do software – Aba fluxograma

Nas abas da esquerda estão as entradas do sistema. Na primeira aba, podem ser feitas a inclusão, a exclusão e a alteração de organismos. Na segunda aba podem ser feitas a inclusão e a exclusão de organismos na teia, e a alteração das características variáveis do organismo para aquela determinada teia. Na terceira aba podem iniciar, pausar e alterar as condições da teia para a simulação. As abas da direita são as saídas do sistema e em cada uma delas é mostrada a imagem que lhes cabe (Figura 12). Na primeira aba, de fluxograma, como está mostrado na Figura 11, serão mostrados os animais componentes da teia e como estão relacionados. Na segunda aba de imagens são mostradas as imagens dos organismos da teia com suas respectivas figuras ilustrativas e a indicação de sua população.

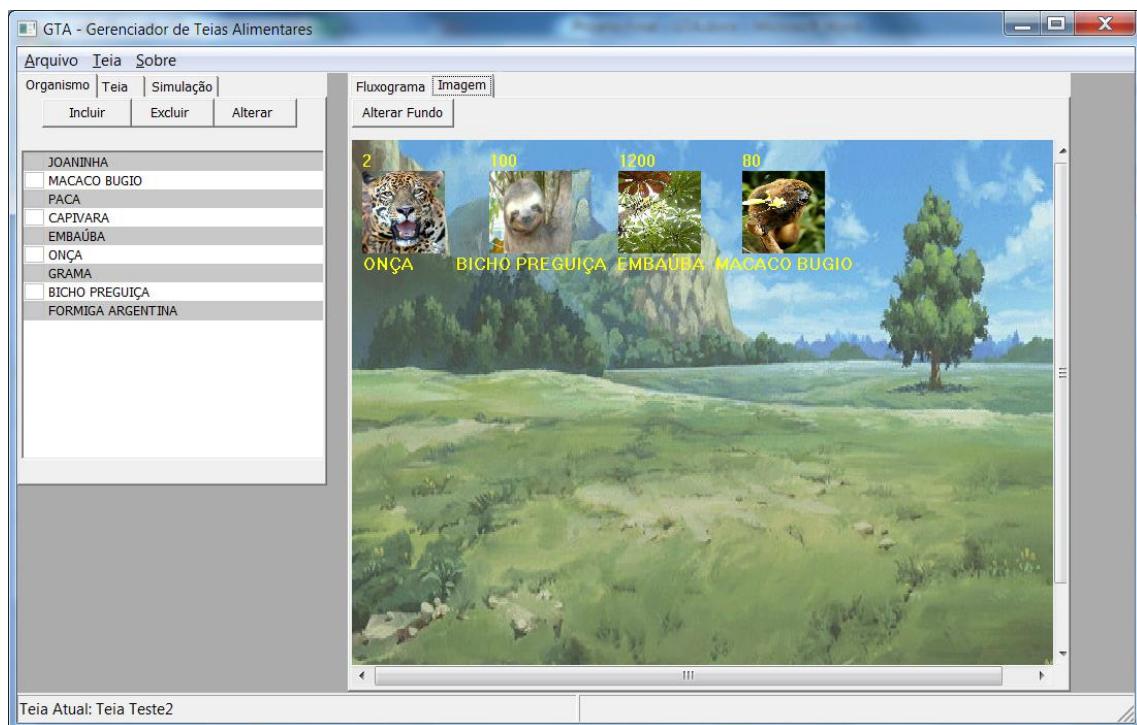


Figura 12 - Interface gráfica - Aba imagem

O diagrama de classes para as interfaces gráficas estão representados nas Figura 13 e 14.

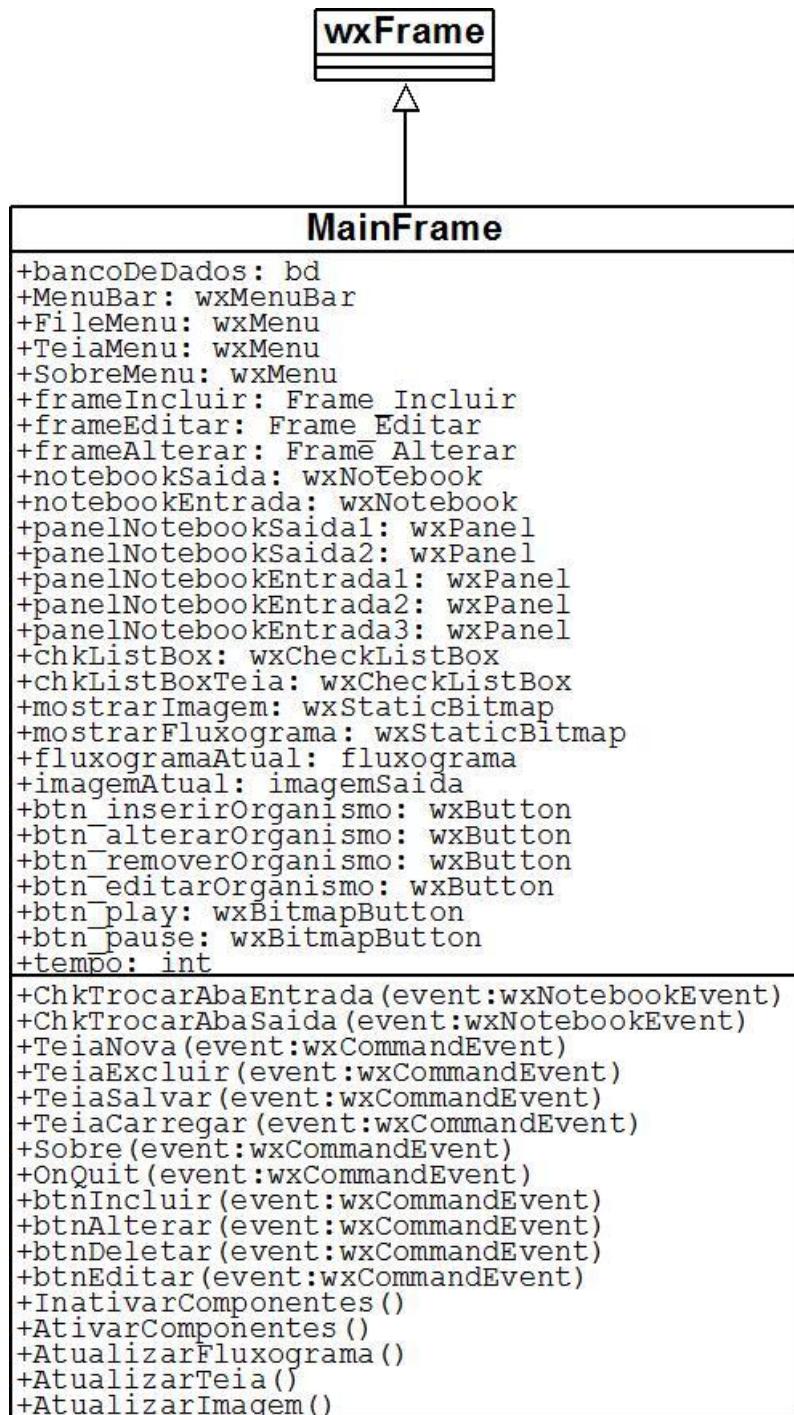


Figura 13 – Classes interface gráfica 1

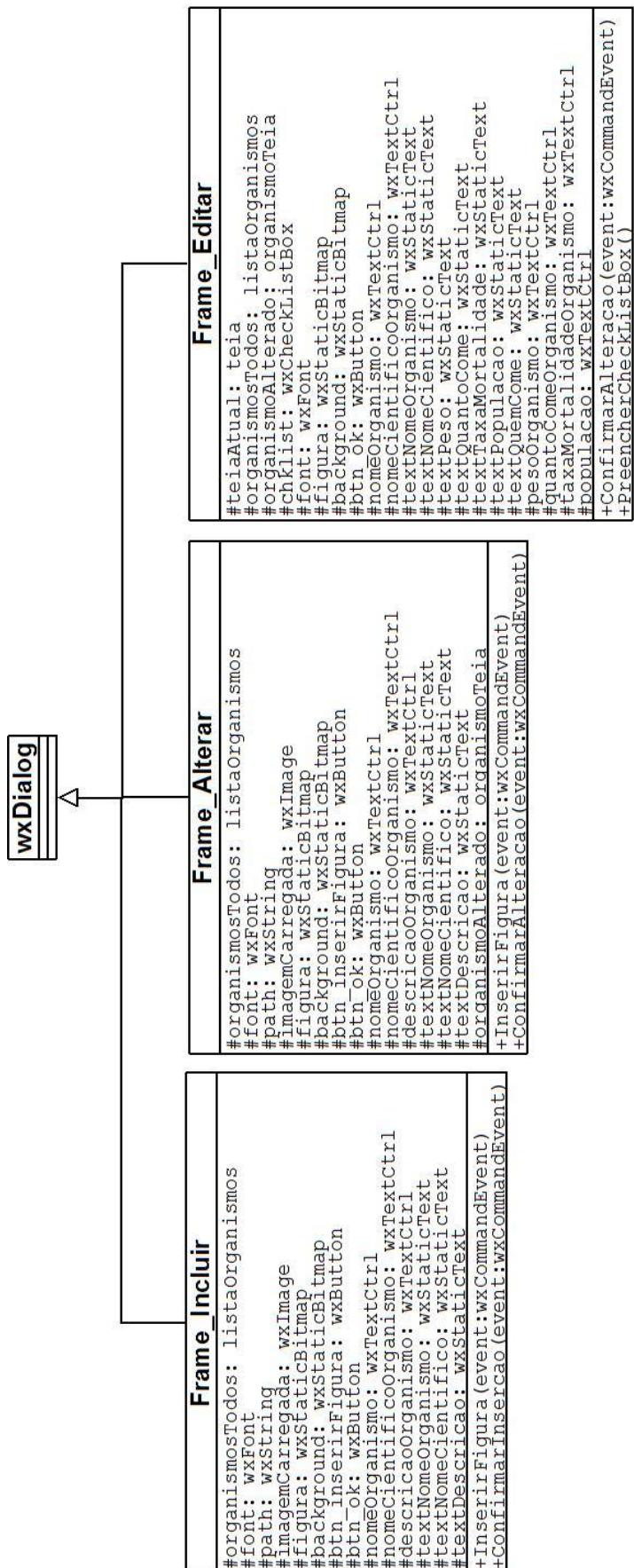


Figura 14 - Classes interface gráfica 2

Para obter mais informações sobre a interface e como ela funciona pode-se consultar os documentos de projeto e do manual que estão nos apêndices “Apêndice C – Documento de Projeto” e “Apêndice E – Manual do Usuário”.

4.4 – Problemas e soluções

Durante a implementação, e após a implementação do software foram encontrados alguns problemas. Os problemas encontrados e solucionados estão descritos no capítulo 6, juntamente com a resolução do problema.

Capítulo 5

Correção de Especificação de requisitos

5.1 – Introdução

Esse capítulo trata da alteração realizada na especificação de requisitos do projeto depois de gerada uma primeira versão do software. Essa alteração significou o desenvolvimento de uma nova funcionalidade do software, um gráfico.

5.2 – Interfaces gráficas

Uma nova interface gráfica foi colocada como requisito após a geração de uma versão do software. Essa nova funcionalidade surgiu depois de várias conversas com o orientador do projeto. Previu-se a necessidade da implementação de uma forma de visualização por gráfico, ou seja, uma alternativa de verificação da evolução das populações com o passar do tempo (Figura 15).

Essa funcionalidade permite que o professor explique conceitos de forma mais fácil utilizando esse gráfico, que torna possível visualizar a queda da população da presa ao passo que a população dos predadores sobe. Com a extinção de uma presa pode-se visualizar a população do predador caindo, e outros conceitos que podem ser elaborados pelo professor durante a execução do software, conforme a demanda.

Esse gráfico deveria apresentar um mínimo de facilidades para o usuário. Dentre muitas foram escolhidas algumas: escolher qual dos organismos fica visível; escolher um organismo para ter a sua curva mostrada em destaque; auto ajuste do gráfico; além de ser possível também, com a simulação em pausa, ampliar o gráfico em algum ponto específico.

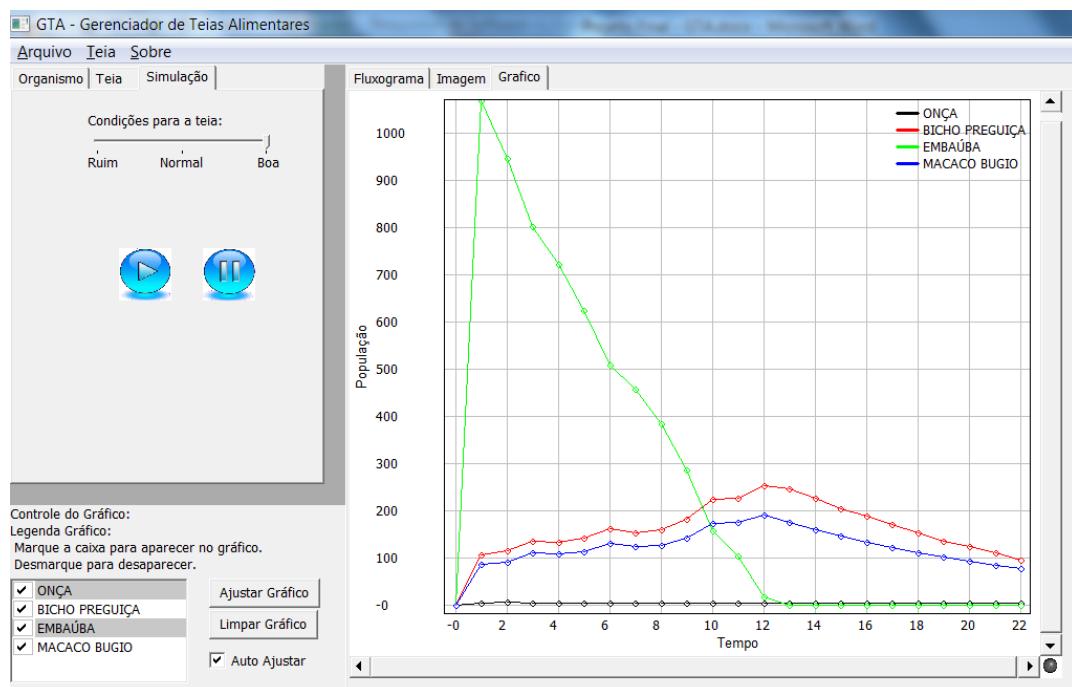


Figura 15 - Interface gráfica para o gráfico

5.3 – Conclusão

Foi determinada essa alteração no documento de especificação de requisitos, e essa deveria ser passada para a próxima fase na etapa da cascata, a implementação.

Capítulo 6

Correção de Implementação

6.1 – Introdução

Terminando a implementação foram encontrados alguns problemas que deveriam ser solucionados, e esse capítulo tem o objetivo de mostrar os problemas encontrados e suas respectivas soluções.

6.2 – Ferramentas

Como foi dito no capítulo anterior, após a implementação ficou decidido que uma nova funcionalidade seria introduzida ao software, o gráfico. Essa funcionalidade foi implementada utilizando uma biblioteca chamada “wxPlotCtrl”. Ao realizar uma tentativa de compilação do software com essa biblioteca observou-se que foi projetada para utilizar a versão 2.8 da biblioteca WxWidgets, ou seja, a versão que estava sendo utilizada deveria ser atualizada (versão 2.6 para 2.8). Foi feita uma tentativa de atualização e não estava sendo possível conseguir uma versão nova que se encaixasse no projeto (a biblioteca WxWidgets tem diretivas de compilação, e nenhuma se encaixava com a utilizada). Continuando com a procura na internet foi encontrada uma nova IDE que utilizava a biblioteca WxWidgets 2.8 nativamente, e o seu nome era wxDev-C++. Ao instalar essa IDE o código compilou sem problemas, assim como a biblioteca “wxPlotCtrl”.

6.3 – Algoritmo da teia

Com o algoritmo de resolução funcionando com o modelo Lotka-Volterra não foi possível obter o controle esperado sobre as características variáveis dos organismos. Uma pequena variação nos coeficientes do modelo Lotka-Volterra gerava gráficos que não tinham um significado correto, ou a população aumentava indefinidamente, ou caia a zero. Se um dos objetivos do projeto, o professor utilizar o software para ensinar teias

alimentares, não estivesse sendo alcançado, então o software deveria ser alterado. Essa alteração consistiu na mudança do algoritmo, o modelo Lotka-Volterra não seria mais usado, mas outro algoritmo que pudesse fornecer o controle necessário para a utilização, assim como apresentar dados coerentes. Partindo daí, elaborou-se o algoritmo que está demonstrado na Figura 16.

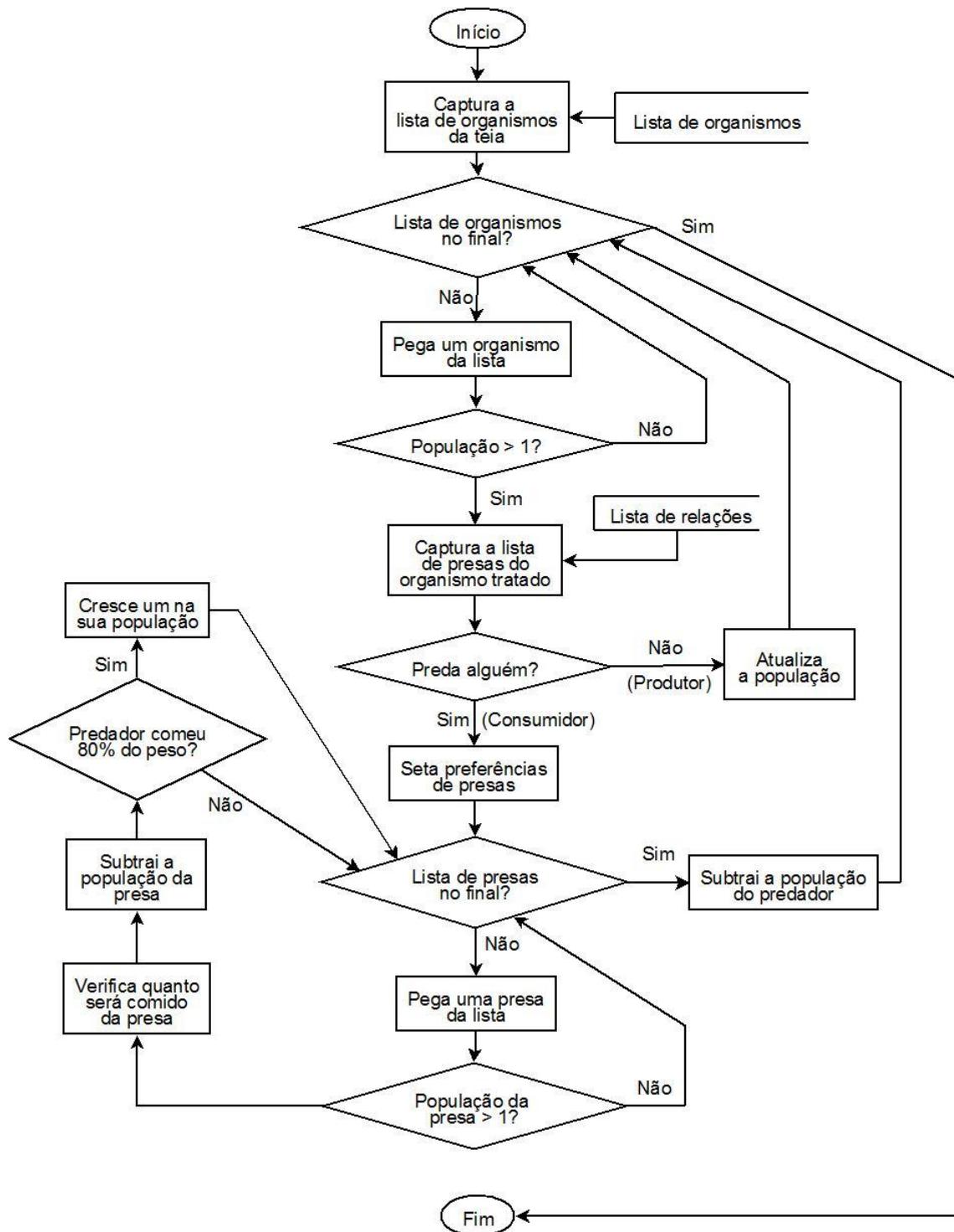


Figura 16 - Fluxo do algoritmo da teia

6.4 – Gráfico

Com a implementação da nova forma de saída em gráfico, algumas características no software foram alteradas. Na divisão dos módulos o gráfico deveria ser incluído no módulo de entrada e saída de dados (Figuras 17 e 18).

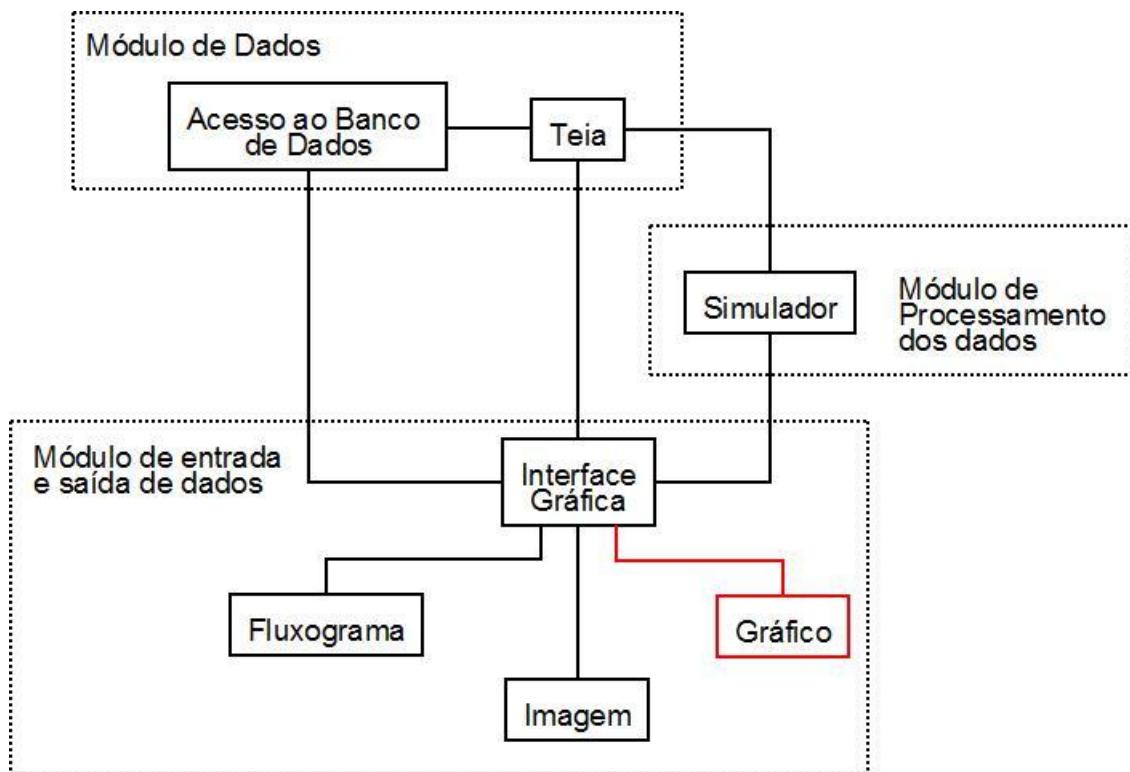


Figura 17 - Divisão dos módulos - inserção do gráfico

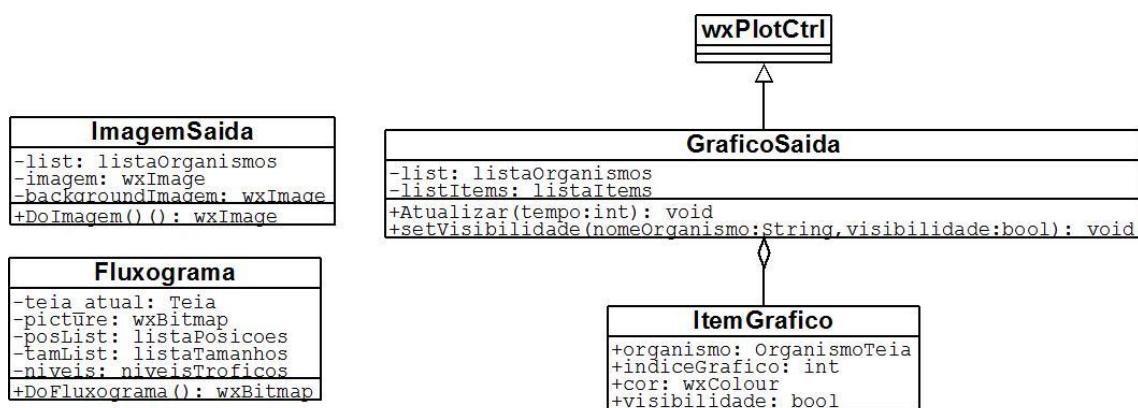


Figura 18 - Classes de saída - com gráfico

6.5 – Instalador

Para a instalação do software deveriam ser seguidos um grande número de passos: copiar a pasta do software para um local, instalar o banco de dados MySQL, executar o script para instalar as tabelas e fazer uma carga de dados inicial, instalar o driver ODBC e criar a fonte de dados ODBC para o banco de dados.

Todas essas tarefas demandariam tempo e um pouco de experiência do usuário. Pensando nisso, foi feito um instalador do software, que faria tudo de forma automatizada, poupando o tempo do usuário, evitando possíveis erros e com isso o usuário não precisaria ter nenhum tipo de experiência em nenhum desses tópicos para fazer o software funcionar.

Para isso foi necessário buscar um tipo de instalação do banco de dados MySQL que não demandasse uma instalação prévia, e também, já viesse com todas as tabelas prontas, com uma carga inicial. O mesmo pensamento seguiu para o driver ODBC. Para inserir a fonte de dados conectando ao banco de dados foi utilizado um arquivo de lote do Windows (.bat) inserindo a fonte diretamente no registro.

Ao término desse processo faltava criar o instalador que copiasse os arquivos e iniciasse os instaladores e arquivos de lote. Para tal, foi usado o software “Mep Installer v 3.0.0”. Com ele é possível gerar um script que cuida de todo o processo de instalação.

Capítulo 7

Resultados

7.1 – Introdução

O software foi testado em uma escola localizada na cidade de Petrópolis-RJ com três turmas do segundo seguimento do ensino fundamental. A professora utilizou o software para reforçar o ensino de teias alimentares, tema que havia sido discutido previamente.

As turmas que testaram o software eram do 7º, 8º e 9º anos, e a professora foi a mesma para as três turmas. Durante a aula, foi utilizado um notebook ligado a um projetor, de modo que toda a turma pudesse acompanhar, ao mesmo tempo, as explicações sobre as funções do software e aplicar os conceitos referentes às teias alimentares. Todas as explicações e dúvidas sobre o funcionamento do software foram realizadas pela professora.

Após essa explicação, foi sugerido que alunos criassem suas teias, primeiro em papel e posteriormente no computador da professora. Assim, os alunos, em duplas ou trios, montaram suas próprias teias no software. Eles puderam, também, inserir novos organismos, alterar os existentes, simular e verificar como sua teia se comportaria com o tempo.

Conforme os alunos terminavam suas simulações, uma folha de avaliação, não obrigatória, foi passada para eles preencherem. A professora também recebeu uma ficha de avaliação. Foi passada ainda uma lista de e-mails para que o software fosse enviado para os mesmos e utilizado em casa.

7.2 – Quantitativo

A Tabela 1 descreve alguns dados sobre o teste realizado com o software:

	7º ano	8º ano	9º ano	Total
Tempo de Aula	100 min	50 min	50 min	200 min
Quantidade	14	12	10	36
Erros de usuário	00	01	00	01
Falhas do software	01	02	00	03

Tabela 1 - Tabela de resultados

O tempo de aula se mostrou um fator determinante para que todos os alunos pudessem utilizar o software. Na turma do 7º ano todos os alunos conseguiram criar organismos e teias. Já nas outras turmas, como o tempo era menor, alguns alunos não puderam utilizar o programa. A turma do 7º ano teve a possibilidade de usar o software por mais tempo, e com isso aprenderam mais sobre ele. Além disso, a professora pode explorar melhor o tema, a partir do software.

A quantidade de alunos não foi um fator problemático nesse caso, pois o número de alunos por turma era pequeno, e dessa forma a professora teve um controle melhor. Além disso, a utilização de apenas um computador facilitou o monitoramento.

Os erros cometidos pelos usuários foram ínfimos em relação ao total de alunos. Apenas um erro ocorreu na turma do 8º ano, e o aluno, nesse erro, confundiu o local onde deveriam ser incluídos os organismos na teia com o local onde se exclui os organismos. Todos os outros alunos conseguiram realizar e compreender as funções do software com clareza.

Durante a utilização, o software apresentou duas falhas que ocorreram durante a aula do 8º ano. Uma delas foi uma falha de conexão com o banco de dados, que foi resolvida com o reinicio do software. A outra ocorreu quando o aluno, no momento de estabelecer as relações entre os organismos, colocou um organismo predando o outro e vice versa, com isso o software entrou em um loop infinito na montagem do fluxograma.

Em relação aos erros do software, em um determinado momento a população de um dos produtores não estava atualizando, e isso gerou um erro de interpretação do

gráfico por parte do aluno. Aparentemente o organismo tinha sido extinto, o que não era o caso.

7.3 – Qualitativo

Para indicadores qualitativos podem-se levar em consideração os requisitos não funcionais, explicitados no capítulo que trata da especificação de requisitos. Entre eles estão: usabilidade, manutenibilidade e confiabilidade. A análise qualitativa do software foi realizada levando-se em conta a usabilidade e a confiabilidade.

A usabilidade se caracteriza pela facilidade com que os usuários aprendem o software, o tempo de aprendizado e até mesmo se os usuários conseguem intuir alguma função do software (Sommerville [5]). A facilidade de aprendizado se mostrou presente em todas as turmas, pois a maioria dos alunos prestou atenção na explicação dada pela professora, e quando solicitados sobre algum tópico, responderam prontamente. Quanto à questão do tempo de aprendizado pode-se averiguar que ocorreu um aproveitamento significativo. Em 50 minutos de aula a professora foi capaz de explicar o funcionamento do software, relembrar brevemente o conceito de teias alimentares e ainda foi possível que os alunos utilizassem o software. Observou-se, também, o uso da intuição, por parte de alguns alunos, na hora de decidir onde procurar uma função do programa.

O software ficou ligado por 200 minutos e durante esse tempo apresentou três falhas. Não foram executados testes exaustivos no software, e isso levou a ocorrência de pelo menos uma das falhas. A falha que leva ao loop infinito pode ser consertada com uma alteração no algoritmo de montagem do fluxograma. As outras falhas ocorreram com o decorrer do funcionamento do software. A confiabilidade não pode ser avaliada nessa versão, uma vez que não há um padrão de tempo de falha a ser referido. Em uma versão posterior será possível avaliar a confiabilidade do sistema, utilizando as marcas obtidas com os resultados desse teste.

7.4 – Avaliação

Uma ficha de avaliação do software foi distribuída para os alunos e para a professora que utilizou o software. Nesta ficha foram inseridas questões relevantes para cada perfil. A ficha dos alunos continha, em sua grande maioria, perguntas sobre a aparência e a usabilidade do sistema, e ao final um espaço para opiniões e sugestões. Já a ficha da professora continha mais questões de cunho pedagógico. Essas questões abordaram a eficácia do software em atingir o seu objetivo como ferramenta de ensino, algum possível problema em utilizar o software e as opiniões, juntamente com as sugestões.

Os alunos apresentaram, em sua grande maioria, críticas e sugestões pertinentes para o software. Cerca de 80% dos alunos gostariam que as cores do software, ou pelo menos do fluxograma, fossem alteradas. Outros 20% gostariam que a interface ficasse mais bonita. Outros 10% queriam que o software fosse mais realista, com animações, vídeos e som. Uma pequena quantidade, 10%, gostaria também que a fonte e as imagens fossem maiores. Com exceção do que já foi citado, ocorreram algumas sugestões de forma pontual.

7.5 – Conclusão

Os resultados obtidos foram acima das expectativas. A participação dos alunos, a utilização do software de forma correta pela professora e o grande número de sugestões fizeram desse teste um sucesso. Além disso, foi importante saber que o software foi bem aceito tanto pelos alunos quanto pela professora.

Capítulo 8

Conclusão

8.1 – Métodos e ferramentas utilizados

O método de desenvolvimento de software foi orientado a objetos com ciclo de vida cascata, e este ciclo de vida tem prós e contras. Para passar de etapa nesse modelo é necessário que a outra esteja pronta. Isso é positivo, pois faz com que a equipe pense mais em cada etapa, visando uma diminuição nos possíveis erros futuros. Em compensação, o modelo cascata torna o projeto mais demorado, é necessário mais tempo para executar todas as fases. Mas utilizar esse modelo foi importante para ver que para projetos em que não se domina completamente os requisitos, o modelo de prototipação pode ser mais produtivo. Primeiro foi feito um software e depois foi feito um segundo baseado na experiência adquirida no primeiro. O ciclo foi efetivamente um ciclo de vida com prototipação.

As ferramentas de IDE utilizadas foram a DevC++ e a wxDevC++. As duas possuem algumas facilidades quanto à instalação de bibliotecas e compilações, da mesma forma que todas as IDE's, porém, a wxDevC++ ainda é pouco desenvolvida e em alguns momentos apresenta falhas inesperadas. A ferramenta de *debug* de ambas não funciona de forma correta. É necessária bastante pesquisa na hora de escolher a IDE a se utilizar, pois isto pode diminuir de forma significativa o tempo de programação.

A ferramenta para gerenciamento de versões (TortoiseSVN) não foi muito utilizada, uma vez que o software não estava sendo desenvolvido por uma equipe, e sim por uma pessoa somente. Foi preciso utilizar apenas uma vez, e foi o suficiente para demonstrar a importância em usar esse tipo de ferramenta.

O repositório na internet (Dropbox) representou uma importante característica na mobilidade. Permitiu acessar os arquivos de lugares diferentes, e o mais importante de tudo, agindo de forma sincronizada, evitando assim carregamento de arquivos de um lado para outro e versões diferentes.

A biblioteca WxWidgets foi utilizada em todas as áreas do projeto, desde a conexão ao banco de dados até as interfaces gráficas. No começo da programação ela se mostrou um pouco complicada, mas com o uso ela passou a facilitar a programação.

Essa biblioteca é muito poderosa, e conta com uma enorme abrangência de áreas de atuação.

O gerador de instalação (Mep Installer) no início apresentou algumas dificuldades, pois era uma linguagem nova a se aprender. Mesmo com o instalador pronto, houve a necessidade de estudar mais sobre essa ferramenta que possui grandes possibilidades na geração de instalações.

A ferramenta usada para gerar os fluxogramas e diagramas de classes (DIA) já era conhecida. Possui diversas ferramentas para muitas áreas diferentes, desenhando desde simples fluxogramas, esquemas de ligação de redes de computadores, circuitos elétricos, enfim, diversos modelos para variadas áreas de atuação. Utilizar esse software facilitou a criação dos diagramas e fluxogramas pelo conhecimento prévio sobre ele. Como essa ferramenta é apenas um editor de gráficos, para sistemas mais complexos é mais conveniente utilizar ferramentas dedicadas à análise e ao projeto orientado a objetos, essas ferramentas permitem verificações sintáticas e semânticas dos modelos.

8.2 – Conhecimentos adquiridos

Nesse projeto foram usados conhecimentos de áreas distintas. Foi necessário o uso de engenharia de software para escolher um método de desenvolvimento, utilizar banco de dados, estudar teias alimentares e desenvolver formas de analisá-las. Além disso, a análise das informações baseada em objetos foi necessária para modelar devidamente o sistema proposto. Foi necessário também utilizar ferramentas de gerenciamento e desenvolvimento de software, uma biblioteca com inúmeras possibilidades, como a WxWidgets, e formas de acesso ao banco de dados.

Os conhecimentos adquiridos com esse projeto não ficaram somente no escopo do aprendizado da faculdade, foram necessárias pesquisas, várias dúvidas foram geradas e solucionadas, e foi possível perceber que não é trivial desenvolver um projeto sozinho. É importante conversar com outras pessoas sobre as possíveis soluções para problemas, sendo assim, o trabalho em equipe se mostra relevante nesses casos.

O fato do projeto não ser especificamente na área de engenharia, e abranger áreas diferentes como a educação e a biologia, proporcionou um aprendizado maior e não restrito. Foi possível perceber com esse projeto que é importante para a engenharia atuar em outras áreas, resolvendo outros problemas que estão fora do cotidiano.

Foi gratificante ver o software sendo utilizado com as turmas de teste. O software se mostrou simples para elas e o mais interessante foi a resposta positiva que se percebeu com seu uso. As turmas se mostraram atentas ao que a professora estava explicando, participaram da aula, e o conhecimento sobre teia alimentar foi reforçado.

Os objetivos do software foram cumpridos, o teste se mostrou muito positivo e o que foi aprendido com esse projeto tende a ser levado por todos os projetos futuros.

8.3 – Propostas futuras

Ao longo do desenvolvimento várias idéias diferentes foram surgindo, porém foi necessário fechar o escopo do projeto, senão ele não seria finalizado. As idéias não foram esquecidas, foram anotadas para versões futuras. Além das idéias, estão incluídas nessa lista também as falhas encontradas durante o teste realizado com os usuários.

Quando o teste foi feito com as turmas, uma ficha de avaliação foi passada e várias dicas surgiram dessas fichas. Muitas delas relevantes e passíveis de serem levadas em consideração.

As idéias surgidas durante a implementação se basearam formas de alterar o software para incluir novas funcionalidades. Uma possível nova implementação seria a utilização em rede, onde haveria um servidor, que se conectararia diretamente ao banco de dados, e os clientes fariam a comunicação com esse servidor fazendo as requisições. Dessa forma seria possível a professora trabalhar com a mesma teia para os alunos e verificar as diferentes combinações possíveis que cada um encontrou, mostrando dessa forma a grande variação que pode ocorrer em uma teia alimentar. Outra inclusão de funcionalidade seria a adição de eventos aleatórios. Como exemplo disso, poderiam ocorrer quedas bruscas na população de algum organismo, um aumento significativo na população de outro organismo, e tudo isso com algumas explicações biológicas, como a ocorrência de uma migração, de alguma doença que possa devastar uma população, e esses eventos poderiam ser incluídos e excluídos da mesma forma.

Uma ótima sugestão de uma das alunas da turma foi a inclusão de uma funcionalidade para imprimir os dados da teia, o fluxograma e o gráfico. Outra idéia relevante foi alterar as cores do software para cores menos vivas.

Referências Bibliográficas

- [1] RICKLEFS, R. E. , *A Economia da Natureza*. Rio de Janeiro, Editora Guanabara Koogan S.A., 1996.
- [2] LIMA, J. G. S. e SELLES, S. L. E. , *O conceito de teia/cadeia alimentar como base para a discussão do processo de transposição didática*. Anais do VIII EPEB (Encontro Perspectivas do Ensino de Biologia). (CD). São Paulo, 2002.
- [3] BRADBURY, RAY, *Os frutos dourados do sol*, 2^a Edição, conto Um Som de Trovão, Editora Francisco Alves, 1982.
- [4] LINHARES, S. e GEWANDSZNAJDER, F. *Biologia hoje*. São Paulo, Editora Ática, 2003.
- [5] SOMMERRVILLE, IAN, *Engenharia de Software*, 8^a Edição, São Paulo, Editora Pearson Addison-Wesley, 2007.

Apêndice A - Documento de Planejamento

Apêndice B – Documento de Especificação de Requisitos de Software (ERS)

Apêndice C – Documento de Projeto

Apêndice D – Documento de Projeto de Testes

Apêndice E - Manual do Usuário

Anexo A - Normas de documentação

Anexo B - Fichas de avaliação de software